9-2024

2024 Summer Proceedings Teuscher Lab

Teuscher Group

Christof Teuscher
Portland State University

Chelsea Ogbede
Portland State University

Lauren Sanday
Portland State University

Sofia Vargas
Portland State University

See next page for additional authors

Authors

Teuscher Group, Christof Teuscher, Chelsea Ogbede, Lauren Sanday, Sofia Vargas, and Artem Arefev

NSF REU, altREU, TLAB

Portland State University

teuscher Lab
teuscher-lab.com

NSF

altREyou

2024 SUMMER PROCEEDINGS

NSF REU, altREU, TLAB

SUMMER PROCEEDINGS

Edited by: Chelsea Ogbede
Lauren Sanday
Sofia Vargas
Artem Arefev

Portland State University

teuscher Lab
teuscher-lab.com

altREyou

NSF

2024 Summer Proceedings
– Contributed Books –

Table of Contents

Utilizing Technology for the Greater Good: A Foreword.. 4

Modeling Vegetation Water Stress on Green Roofs: A Study on Idaho Fescue......................... 5

Microcontroller Based Memcapacitor Emulating Platform... 20

CNN-Powered Resume Analysis for Keyword Relevance and Industry Suitability.................. 27

Capturing Active Transportation Counts at Intersections Using Ultralytics
YOLOv8 Image and Video Tagging.. 39

Prediction of Suicide Rates Across the US Using Healthcare Access Data............................... 48

Can Evolution Harness Chaos Inside Analog Circuits for Advantageous Behavior? 65

Reinforcement Learning for Assistive Robots... 82

Predictive Crime Analysis and Visualization in Chicago.. 91

Enhancing Menstrual Health in Kenya: Access to Sanitary Products and Education............... 100

Green Roof: Monitoring Soil Moisture, Plant Health, and Stormwater Management.............. 112

Development and Evaluation of a Cyberbullying Detection AI Model.................................... 128

Exploring Fractional Leaky Integrate-and-Fire Neurons in Spiking Neural Networks............. 138

Integrating Deep Q-Networks and Reinforcement Learning for Autonomous
Navigation in the Santa Fe Trail Problem.. 159

Innovations in Educational NLP: Creating a Syllabus-Specific LLM....................................... 178

SunMaps: Sustainable Urban Planning Insights for Phoenix.. 185

RunTempo: Dynamic music system based on cadence for enhanced running experience......... 195

Linear Forecasting: Predictive Stock Market Trends.. 203

Forecasting Quarterly GDP Using Multivariate LSTM.. 214

Develop an Agent-based Model for Assessing the Impact of Autonomous Vehicles on Travel and Land Use in Portland, Oregon.. 223

Multilingual Misinformation in Artificial Intelligence.. 238

CrisisComm: LLM for Post-Natural Disaster Aid Acquisition... 249

Remedying Mathematics Education via Adaptivity and Personalization.................................. 260

Orphan Drug Trials: Forecasting Phase Three Clinical Trial Success....................................... 269

Applying Deep Learning Techniques to Diagnose Pneumonia Using Lung X-ray Images....... 281

Utilizing Technology for the Greater Good: A Foreword

How will computation evolve in the coming years? What problems can be tackled using artificial intelligence, in a world increasingly driven by data? And how can that data be used to better inform our decisions as a society? In this unique collection of research projects, each chapter represents a distinct work undertaken by a single individual or a group of students as part of the altREU program led by Christof Teuscher. The projects, rooted in applications of artificial intelligence and innovative computation techniques, examine impactful solutions to numerous pressing challenges affecting communities around the world.

Special thanks go to the editors-in-chief who dedicated their time, skill, and ingenuity to this collective work: Chelsea Ogbede, Lauren Sanday, Sofia Vargas, and Artem Arefev.

Modeling Vegetation Water Stress on Green Roofs: A Study on Idaho Fescue (*Festuca idahoensis*)

Abigail Chable

Abstract This study explores the potential of Idaho fescue (*Festuca idahoensis*), a C3 perennial bunchgrass, as a viable vegetation option for green roofs in Portland, Oregon. Using the Photo3 model, key physiological parameters for Idaho fescue were identified and incorporated into the simulation. The analysis included a comparison of C3, C4, and CAM photosynthetic pathways, as well as an examination of model-projected transpiration rates, carbon assimilation, and stomatal conductance under specific environmental conditions. The findings suggest that Idaho fescue demonstrates promising characteristics for green roof applications, including effective cooling, stormwater management, and resilience to Portland's climate. These insights highlight Idaho fescue's potential for improving the sustainability and performance of green roofs.

1 Introduction

Green roofs are transforming urban landscapes by offering a multitude of environmental benefits, including effective stormwater management, enhanced energy conservation, and a boost in urban biodiversity[18]. However, the quest to identify the most suitable vegetation for these green roofs, especially in specific climates, remains an ongoing challenge. This paper delves into the potential of Idaho fescue (*Festuca idahoensis*) on green roofs in Portland, Oregon.

Imagine a plant that can endure the chill of winter and the arid spells of summer—Idaho fescue is just that. This perennial cool-season bunchgrass, with its C3 photosynthetic pathway, boasts good tolerance to cold weather and moderate drought resistance, setting it apart from other fescues[15]. These attributes make *Festuca idahoensis* a compelling candidate for green roofs in Portland, which experiences summer droughts and wet, cool seasons.

Abigail Chable
Portland State University, 1825 SW Broadway Portland OR 97201, e-mail: achable@pdx.edu

The choice of vegetation for green roofs is heavily influenced by the plant's photosynthetic pathway. Photosynthesis, the life-sustaining process where plants convert light into energy, varies across three broad types: C3, C4, and CAM. Each pathway bestows plants with unique advantages, influencing their adaptability and performance in different environments.

This study embarks on an exploration of these photosynthetic pathways and their implications for green roof success in Portland. By comparing the distinctive characteristics of C3, C4, and CAM plants, I hope to uncover which vegetation can maximize the ecological and practical benefits of green roofs in this Mediterranean climate. Join as the potential of Idaho fescue and its peers are examined, paving the way for greener, more resilient urban spaces.

1.1 Differences Between CAM, C4, and C3 Photosynthesis

C3 is the most common photosynthetic pathway and the default for most plants. The C4 and CAM pathways are adaptations from the "classic" C3 pathway [8]. In C3 photosynthesis, the Calvin cycle produces a 3-carbon compound, hence the name C3 [5]. Plants using this pathway face several limitations. One major factor is that when stomata open to take in carbon dioxide, they also lose water vapor, making them less efficient in water use. Additionally, C3 photosynthesis is constrained by photorespiration [9]. Dr. Samantha Hartzell compares C3 photosynthesis to an Otto cycle internal combustion engine: both are early, consistent technologies with inherent drawbacks [9].

C4 photosynthesis, named for its 4-carbon product, is known for efficient water use. C4 plants thrive in warmer, drier climates due to reduced photorespiration, an adaptation that allows them to better handle changing environmental conditions. This type of photosynthesis is considered a "turbocharger" for C3 carbon concentration, improving performance and water use efficiency [9] [12].

Crassulacean acid metabolism (CAM) plants uptake CO_2 at night and, as a result, metabolize malic acid during the day [16][1]. Like C4 photosynthesis, CAM is able to avoid photorespiration. This adaptation is beneficial in environments with very limited water access, like deserts, and is typically found in succulent species.

Grasses can be categorized into C3 and C4 types, each with its own seasonal growth patterns [17]. C3 cool-season grasses are prevalent in temperate climates [4] and tend to enter summer dormancy. In contrast, C4 warm-season grasses, which dominate in warmer regions [4], thrive during the hotter months but tend toward winter dormancy.

Idaho fescue, the main plant under investigation in this study, utilizes the C3 photosynthetic pathway. As a C3 plant, it may face challenges in water use efficiency, particularly in warmer climates where water conservation is crucial. However, its suitability for green roofs in Portland, Oregon, is promising due to the region's cool climate characteristics. In such environments, Idaho fescue might perform well because the lower temperatures reduce photorespiration and water loss. Additionally, Idaho fescue has demonstrated some drought tolerance, which could help it survive the summer droughts commonly experienced in Portland.

1.2 Key Concepts in Plant Physiology

In order to understand the analysis performed in this paper, there are a few key concepts to highlight.

Stomatal conductance: Stomata are small openings on the surface of plants that play a critical role in gas exchange [6]. They regulate the entry of carbon dioxide (CO_2) into the plant for photosynthesis and the exit of water vapor through transpiration [2]. Stomatal conductance is a measure of the rate at which CO_2 enters and water vapor exits the plant through these stomata, reflecting the plant's efficiency in managing gas exchange and water loss [6].

Carbon assimilation: Carbon assimilation refers to the process by which plants take in carbon dioxide (CO_2) from the atmosphere and convert it into organic compounds during photosynthesis. Carbon assimilation takes place via the three photosynthetic pathways, each having its own method of assimilation [13].

Transpiration: Transpiration is the way in which plants release water vapor into the atmosphere via the stomatal openings. This movement of water from the roots, through the plant, and out into the atmosphere plays a crucial role in transporting nutrients and cooling the plant [13].

2 Related Work and Background

There are a great number of studies done on green roofs, many of which prove their value when it comes to environmental benefits. Plenty of studies evaluate different types of vegetation on green roofs, monitoring their success in terms of stormwater runoff amounts. One in particular, Nagase et al. [14], demonstrated that the type of vegetation on a green roof significantly influences the amount of stormwater retained, with their study concluding that while *Sedum* is commonly used, it is one of the least effective species for this purpose. This highlights the importance of seeking and identifying the most effective species to be placed on green roofs.

Another study, Starry et al. [16], followed the photosynthesis and water use of two different types of *Sedum*. One of which is *Sedum album*, the same *Sedum* we are focusing on in modeling. The study goes on to conclude that different types of *Sedum* yield different water-use efficiency and conduct different amounts of CAM photosynthesis.

Hartzell et al. [9] provided valuable insights by drawing a useful comparison between cars and photosynthesis, making the complex process more accessible. Another critical contribution from Hartzell et al. [8] explains the Photo3 model, detailing its results and significance, particularly in an era when CAM photosynthetic models were sparse. Their work has been instrumental in advancing the modeling of photosynthesis.

Focusing on Idaho fescue, the central species of this study, it was determined that research on its photosynthetic rates, leaf area, root area, and other key characteristics is limited. However, Doescher et al. [3] proved particularly useful by measuring the photosynthetic rates, xylem pressure potential, and conductance of Idaho fescue. In instances where specific measurements were lacking in one study, others, like Jacobs et al. [10], provided complementary data, such as leaf area measurements under various levels of defoliation and in comparison with invasive species.

3 Data and Methodologies

To evaluate the suitability of Idaho fescue for green roofs in Portland, key parameters were identified and integrated into the Photo3 model, a sophisticated Python-based tool for simulating photosynthesis [8]. The process of finding these parameters involved extracting and analyzing Idaho fescue data from multiple studies. Although sources were limited, the chosen studies closely matched the conditions expected for fescue on green roofs. Additionally, detailed Portland weather data was also used to run the simulation.

3.1 Data Collection

Data collection for this study comprised multiple sources and methods to ensure comprehensive analysis.

3.1.1 Parameter Data

Leaf area index (LAI), which measures the leaf area per unit ground surface area, is a crucial parameter in plant modeling. To determine the LAI of Idaho fescue, data from a study by Jacobs et al. [10] was used. This study measured several characteristics of Idaho fescue, including leaf area. Two-year-old fescue plants were grown in pots alongside spotted knapweed, defoliated at different levels every 14 days, and, after 50 days, they were extracted and measured. The pot diameter for each plant was provided, allowing for the calculation of a ground surface area of 31.17 cm^2. Using the Automeris Web Plot Digitizer to accurately convert graphical data into numerical form, the leaf area for Idaho fescue clipped once at 0% defoliation was found to be 137.28 cm^2. These values yielded a leaf area index of 4.4.

To determine the stomatal parameters for Idaho fescue Doescher et al. [3] was consulted. This study found photosynthesis, conductance to H_2O, and xylem potentials for Idaho fescue under different conditions. The non-defoliated, non-grazed group of data was selected as it most closely represents the conditions expected for the Idaho fescue used on green roofs. The data from the study's graphs were once again extracted using the Automeris Web Plot Digitizer. Python was then used to generate a graph showing the relationship between xylem pressure potential and photosynthesis. Fig. 1 displays this relationship, with the linear fit applied. The values found, -0.99 MPa and -3.72 MPa, represent the estimated onset point of stomatal closure and the point of complete stomatal closure, respectively.

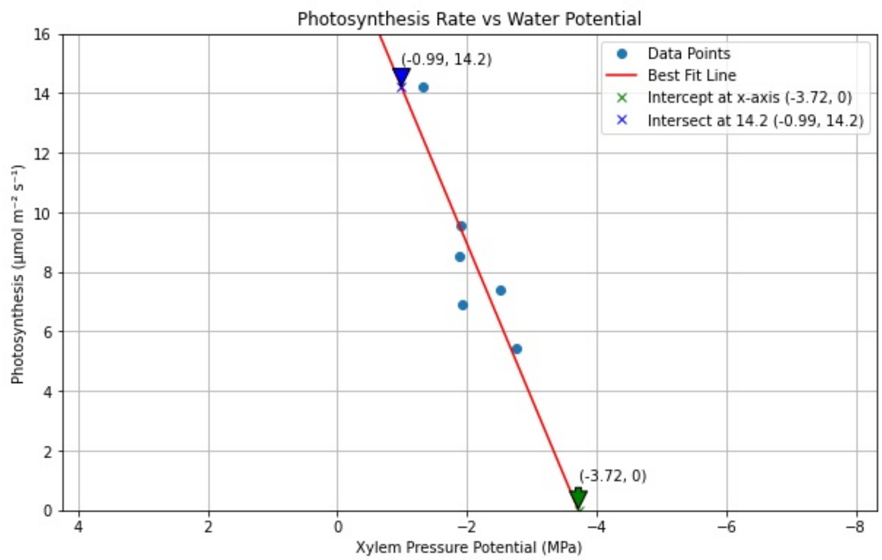


Fig. 1 Relationship between xylem pressure potential and photosynthesis for Idaho fescue. Highlighted points represent the onset point of stomatal closure (-0.99 MPa) and the complete point of stomatal closure (-3.72 MPa). Data points were extracted from Doescher et al. [3].

3.1.2 Weather Data

A spreadsheet containing a full day of Portland July weather data collected from the National Solar Radiation Database was used. Recorded in 10-minute increments, the dataset included Global Horizontal Irradiance (GHI), temperature, and relative humidity. This detailed weather report was crucial to accurately simulate photosynthesis using the Photo3 model.

3.2 Methods

To model the performance of Idaho fescue on green roofs, key physiological parameters were identified and incorporated into the Photo3 model. This involved extracting data from relevant studies, analyzing the relationship between xylem pressure potential and photosynthesis, and adjusting model parameters to match observed data. The process ensured that the model most accurately reflected the specific characteristics of Idaho fescue under various environmental conditions.

3.2.1 Determination of Stomatal Closure Points

Building on the principles established by Farquhar et al. [7], the Photo3 model employs a key equation central to its framework [8]:

$$A_d = A_{\phi,c_x,T_l}\,(\phi, c_x, T_l)\, f_{\psi_l}(\psi_l).$$

In this model, the function $f_{\psi_l}(\psi_l)$ is defined as:

$$f_{\psi_l}(\psi_l) = \begin{cases} 0, & \psi_l < \psi_{IA0} \\ \frac{(\psi_l - \psi_{IA0})}{(\psi_{IA1} - \psi_{IA0})}, & \psi_{IA0} < \psi_l \leq \psi_{IA1} \\ 1, & \psi_l > \psi_{IA1} \end{cases}$$

Using this predefined function, I was able to generate a graph that represents the relationship between photosynthetic rate and water potential. The initial graph which displayed the relationship between xylem potential and photosynthesis originally included an outlier that didn't quite fit the piecewise linear pattern of the data. To improve the accuracy of the model, this outlier was removed, and the linear fit represented the data well. Using this adjusted graph, as seen in Fig. 1, the onset of stomatal closure (PSILA1) was determined to be at -0.99 MPa, where photosynthesis begins to decline from its maximum rate of 14.2 μmol m^{-2} s^{-1}. The maximum rate of photosynthesis was also derived from this study, as 14.2 μmol m^{-2} s^{-1} was the highest value reached by the fescue. The complete point of stomatal closure (PSILA0) was determined to occur at -3.72 MPa, where photosynthesis ceases entirely according to the linear piecewise fit.

3.2.2 Model Implementation

With the newly identified parameters, the model was ready to be tested. Using the weather data mentioned earlier and a few generic parameters, the model was run to observe the initial behavior of *Festuca idahoensis*. As anticipated, the models' carbon assimilation didn't meet the 14.2 μmol m^{-2} s^{-1} maximum. Adjustments to other parameters were necessary to achieve the desired fit. The maximum carboxylation capacity was adjusted to 42.5, which allowed the model to reach the maximum carbon assimilation rate of 14.2 μmol m^{-2} s^{-1}. Additionally, the maximum electron transport rate, typically two times the maximum carboxylation capacity [11], was set to 85. With these adjustments, the model's accuracy was enhanced. See Fig. 2.

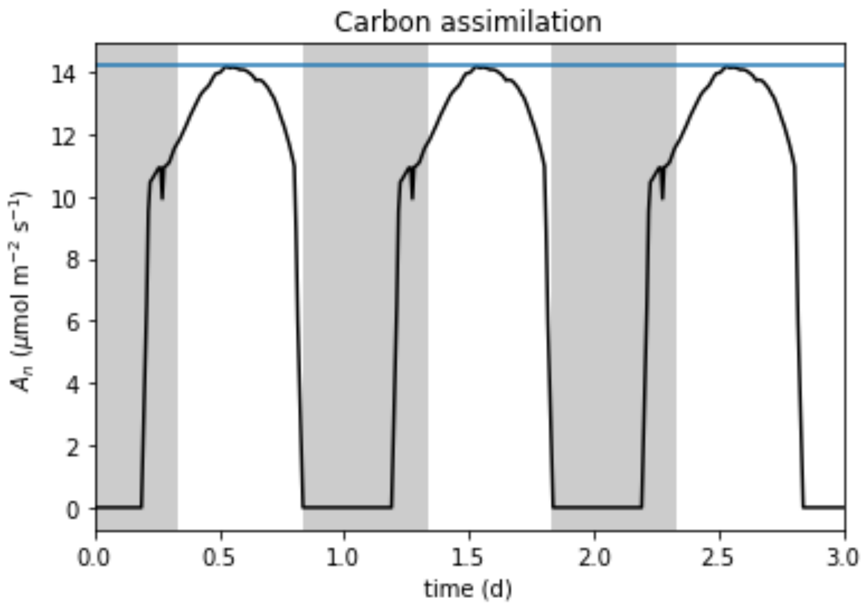


Fig. 2 Carbon assimilation for Idaho fescue over a period of 3 days. The figure shows the diurnal pattern of carbon assimilation, with peaks corresponding to daylight hours and troughs during the night. The blue line indicates the maximum carbon assimilation rate of 14.2 μmol m^{-2} s^{-1}.

For comparative analysis, the model was also run for C4 and CAM species. The Photo3 model had been previously parameterized for *Bouteloua gracilis* (blue grama grass), a C4 photosynthetic pathway species, and *Sedum album* (White stonecrop), a CAM photosynthetic species. Table 1 provides the parameters used to run the simulation.

Each simulation for the three different plants used identical weather data, was run over a period of 3 days, and at a constant soil moisture level. The data for each plant was stored in separate pandas DataFrames, which were then used to generate combined outcome graphs.

Parameter (Description)	*F. idahoensis*	*S. album*	*B. gracilis*
ZR (Rooting depth in meters)	0.1	0.1	0.1
LAI (Leaf area index)	4.4	6.89	2
GCUT (Cuticular conductance in mm/s, per unit leaf area)	0.1802	0	0.1802
GA (Atmospheric conductance in Mm/s, per unit ground area)	61	29	61
RAIW (Root area index, well watered)	5.6	5.6	5.6
GPMAX (Maximum plant hydraulic conductance in μm/(s-MPa), per unit leaf area)	0.13	0.04	0.13
VCMAX0 (Maximum carboxylation capacity in μmol/m^2/sec)	42.5	2.6	39
JMAX0 (Maximum electron transport capacity in μmol/m^2/sec)	85	5.2	180
PSILA0 (Leaf water potential at point of full stomatal closure in MPa)	-3.72	-2.5	-3.2
PSILA1 (Leaf water potential at onset of stomatal closure in MPa)	-0.99	-1	-0.8

Table 1 Comparison of parameters for *Festuca idahoensis*, *Sedum album*, and *Bouteloua gracilis* in the Photo3 model.

4 Results

This section presents the simulation results for Idaho fescue, as well as comparative analyses with *Bouteloua gracilis* (C4 species) and *Sedum album* (CAM species).

4.1 Transpiration

Highlighted in Fig. 3, Idaho fescue shows a steady increase in transpiration starting around sunrise, peaking at approximately midday, and then gradually decreasing until sunset. The peak transpiration rate for Idaho fescue reaches the highest transpiration value, indicating that this C3 plant has the highest transpiration rate among the three plants.

The transpiration rate for blue gramma grass also increases with the onset of daylight, but it peaks slightly earlier and at a lower maximum rate compared to the C3 plant. After reaching its peak, the transpiration rate decreases steadily. The earlier peak and lower maximum rate reflect the C4 photosynthetic pathway's higher efficiency in water use under high light and temperature conditions.

White stonecrop, a CAM plant exhibits a unique transpiration pattern with multiple smaller peaks throughout the day. The transpiration rate remains low during the early hours, increases with minor fluctuations throughout the day, and peaks around the early evening. This pattern can be explained by the fact that *Sedum album* can experience varying degrees of CAM photosynthesis, as noted by Starry et al. [16]. Therefore, this plant engages in more daytime transpiration than stronger CAM plants, which typically restrict their gas exchange to nighttime.

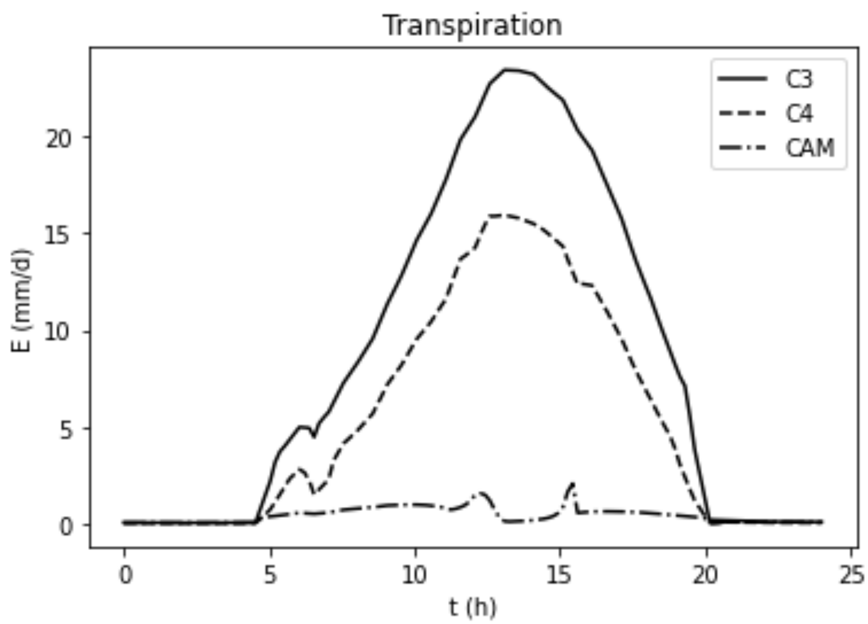


Fig. 3 Daily transpiration rates in C3, C4, and CAM plants: The graph illustrates the variation in transpiration rates (E) over a 24-hour period for plants with C3, C4, and CAM photosynthetic pathways. C3 plants show the highest transpiration rates, followed by C4 plants, while CAM plants maintain low transpiration rates throughout the day.

4.2 Carbon Assimilation

In Fig. 4, Idaho fescue shows a steady increase in carbon assimilation starting around sunrise, reaching a plateau during the late morning to early afternoon hours, and then gradually decreasing until sunset. This steady assimilation during daylight hours is typical of C3 plants, which perform best under moderate light and temperature conditions.

The carbon assimilation rate for the C4 blue gramma grass increases sharply with the onset of daylight, reaching a peak much higher than the C3 plant, and then decreasing steadily. This high peak and efficient carbon assimilation reflect the C4 photosynthetic pathway's adaptation to high light intensity and temperature, allowing these plants to fix carbon more efficiently during the hottest parts of the day.

White stonecrop, the CAM plant, exhibits an interesting carbon assimilation pattern, with very low assimilation rates during most of the day and a slight increase around 15 hours. The highest assimilation rate is significantly lower than those of the C3 and C4 plants, highlighting the CAM pathway's strategy of fixing carbon primarily at night to minimize water loss.

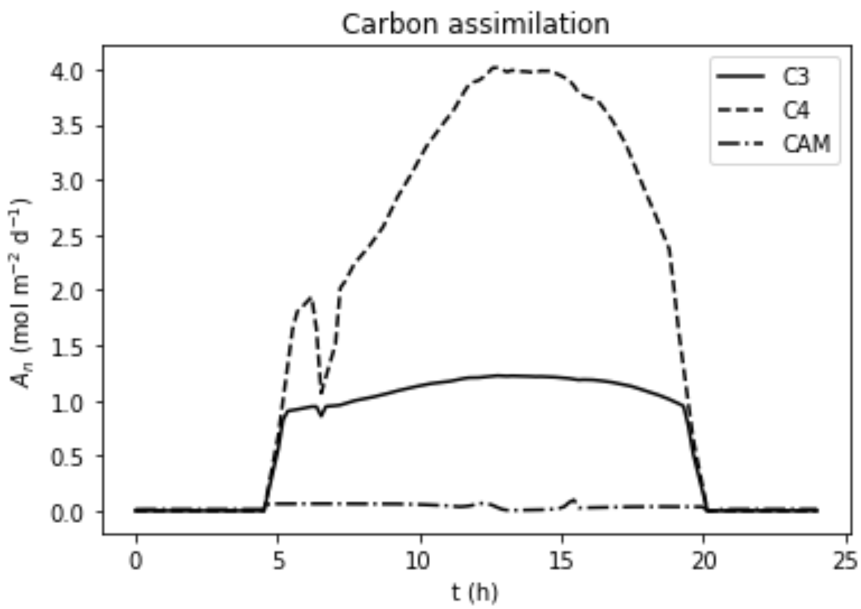


Fig. 4 Daily carbon assimilation rates in C3, C4, and CAM plants: The graph shows the variation in carbon assimilation over a 24-hour period for plants using C3, C4, and CAM photosynthetic pathways. C4 plants exhibit the highest assimilation rates, followed by C3, while CAM plants maintain low assimilation rates during the day.

4.3 Stomatal Conductance

In reference to Fig. 5, Idaho fescue shows a rapid increase in stomatal conductance starting around sunrise and then maintains a relatively high level throughout the daylight hours. The conductance rate begins to decrease around sunset, returning to near-zero levels by nighttime. This pattern reflects the C3 photosynthetic pathway's reliance on stomatal opening during daylight for CO_2 uptake, leading to higher water loss through transpiration.

Blue grama grass also shows a sharp increase in stomatal conductance at sunrise, reaching a peak slightly lower than that of Idaho fescue (C3). The conductance remains relatively steady and is greater than that of Idaho fescue, slightly declining throughout the day.

White stonecrop exhibits very low stomatal conductance during the day, with minor increases at night. The conductance remains significantly lower than that of C3 and C4 plants, highlighting its adaptation to arid environments.

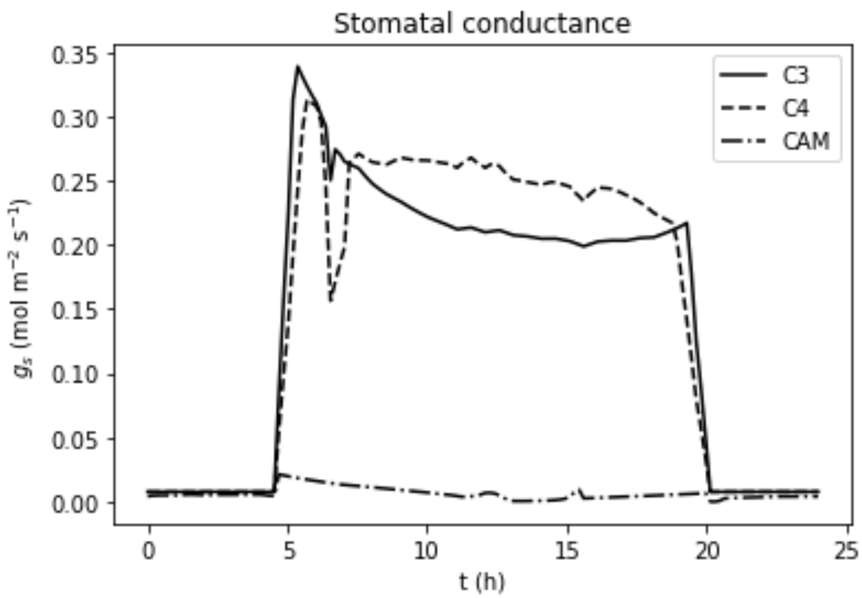


Fig. 5 Daily stomatal conductance in C3, C4, and CAM plants: The graph shows the variation in stomatal conductance (gs) over a 24-hour period for plants with C3, C4, and CAM photosynthetic pathways. C3 and C4 plants exhibit higher stomatal conductance during the day, while the CAM plant maintain low conductance throughout the day.

5 Conclusion

Taking a close look at these three plants provides valuable insight into which plant may be most efficient on Portland green roofs. Though more extensive studies should be done, Early green roof trials with grass similar to *Bouteloua gracilis* (blue grama grass) have shown that it doesn't perform as desired in cold Portland winters. In contrast, *Festuca idahoensis* (Idaho fescue) could prove to be a better-suited grass for Portland's green roofs.

5.1 Comparison of C3, C4, and CAM Plants

- **Idaho fescue (C3)**: Demonstrated higher daytime transpiration rates, which contribute to effective cooling and enhanced stormwater management. Its adaptation to colder climates makes it a strong candidate for green roofs in Portland, where winters are cool and wet.
- **Blue grama grass (C4)**: While efficient in water use and carbon assimilation under high light and temperature conditions, early trials on similar grass indicated that blue grama grass does not perform well in Portland's cold winters.
- **White stonecrop (CAM)**: Showed low transpiration rates and carbon assimilation during the day, with slight increases at night. CAM plants are well-suited for arid environments but may not provide the same level of cooling and stormwater management as C3 plants in Portland's climate.

5.2 Discussion

Idaho fescue exhibits higher transpiration rates than the other contenders during the day. This quality is significant because it indicates the plant may be able to enhance the cooling effect that green roofs provide, further mitigating the urban heat island effect. Additionally, the high transpiration rate suggests better stormwater management. Idaho fescue's higher transpiration rates mean it can effectively reduce soil moisture levels during dry periods, thereby enhancing the green roof's ability to retain stormwater during wet periods. Its resilience to cold also makes Idaho fescue a more viable option for sustaining green roof vegetation year-round. This combination of effective cooling and stormwater management makes Idaho fescue a promising candidate for green roofs in Portland.

5.3 Next Steps

There are several directions for future research to build on the findings of this review. Firstly, conducting field studies on Idaho fescue is essential. These studies should monitor the plant's performance on green roofs over an extended period, capturing data across different seasons and varying weather conditions. Such evidence will provide verifiable information on its suitability and effectiveness in real-world scenarios.

Optimization of green roof systems should also be a focus of future research. Investigations into the most effective soil substrates, watering regimes, and management practices will support the health and growth of Idaho fescue and other suitable plant species. Advanced modeling tools like the Photo3 model can be utilized to predict plant performance under various scenarios, providing valuable insights for designing optimal green roof systems.

Acknowledgements I'd like to express my sincere gratitude to my mentor, Dr. Samantha Hartzell, for her incredible guidance, support, and consistent encouragement throughout this project. Her expertise and insights have been truly significant in shaping my research and understanding of the field, and I will forever be grateful to her for introducing me to these concepts with such kindness. I also want to extend my thanks to Christof Teuscher for leading an outstanding program; his dedication to his interns is heartwarming, and the supportive environment he has helped cultivate has tremendously strengthened my learning experience. My heartfelt thanks go to Cara Swift for her invaluable collaboration—her help in learning these concepts has been instrumental, and I am deeply grateful for her support. In addition, I would like to thank all of my fellow interns for their hard work and for being such inspirational academics. Finally, I am deeply appreciative of the National Science Foundation for their funding and support, which made this research possible.

References

1. Andrea Bräutigam, Urte Schlüter, Marion Eisenhut, and Udo Gowik. On the Evolutionary Origin of CAM Photosynthesis. *Plant Physiology*, 174(2):473–477, 2017.
2. G. Damour, T. Simonneau, H. Cochard, and L. Urban. An overview of models of stomatal conductance at the leaf level. *Plant, Cell, and Environment*, 33:1419–1438, 2010.
3. P. S. Doescher, T. J. Svejcar, and R. G. Jaindl. Gas exchange of idaho fescue in response to defoliation and grazing history. *Journal of Range Management*, 50(3):285–289, 1997.
4. J. R. Ehleringer. Implications of Quantum Yield Differences on the Distributions of C3 and C4 Grasses. *Oecologia*, 31(3):255–267, 1978.
5. James R Ehleringer and Thure E Cerling. C3 and C4 Photosynthesis. In *The Earth system: biological and ecological dimensions of global environmental change*, volume 2, pages 186–190. John Wiley & Sons, Ltd, Chichester, 2002. Encyclopedia of Global Environmental Change.
6. Michele Faralli, Jack Matthews, and Tracy Lawson. Exploiting natural variation and genetic manipulation of stomatal conductance for crop improvement. *Current Opinion in Plant Biology*, 49:1–7, 2019.
7. Graham D. Farquhar, Susanne von Caemmerer, and Joseph A. Berry. A biochemical model of photosynthetic CO_2 assimilation in leaves of C3 species. *Planta*, 149(1):78–90, 1980.
8. Samantha Hartzell, Barlett Mark S., and Amilcare Porporato. Unified representation of the C3, C4, and CAM photosynthetic pathways with the Photo3 model. *Ecological Modeling*, 384:173–187, 2018.
9. Samantha Hartzell, Barlett Mark S., Jun Yin, and Amilcare Porporato. Similarities in the evolution of plants and cars. *PLoS ONE*, 13(6), 2018.
10. J. S. Jacobs and R. L. Sheley. Relationships among idaho fescue defoliation, soil water, and spotted knapweed emergence and growth. *Journal of Range Management*, 50(3):258–262, 1997.
11. J. Kattge and W. Knorr. Temperature acclimation in a biochemical model of photosynthesis: a reanalysis of data from 36 species. *Plant, Cell & Environment*, 30(9), 2007.
12. Elizabeth A Kellogg. C4 photosynthesis. *Current Biology*, 23(14):R594–R599, 2013.
13. T. Lazar, Lincoln Taiz, and Eduardo Zeiger. *Plant Physiology*, volume 91. Annals of Botany, 3rd edition, 2003.
14. A. Nagase and N. Dunnet. Amount of water runoff from different vegetation types on extensive green roofs: Effects of plant species, diversity and plant structure. *Landscape and Urban Planning*, 104(3–4):356–363, 2012.
15. Daniel G. Ogle, James Henson, Mark Stannard, Loren St. John, and Thomas A. Jones. Idaho fescue (Festuca idahoensis) Plant Guide. *U.S. Department of Agriculture, Natural Resources Conservation Service, Idaho and Washington State Office*, 2007.
16. O. Starry, J.D. Lea-Cox, J. Kim, and M.W. van Iersel. Photosynthesis and water use by two sedum species in green roof substrate. *Environmental and Experimental Botany*, 107:105–112, 2014.
17. Christopher J. Still, Joseph A. Berry, G. James Collatz, and Ruth S. DeFries. Global distribution of C3 and C4 vegetation: Carbon cycle implications. *Global Biogeochemical Cycles*, 17(1):1006, 2003.
18. Nicholaus D. VanWoert, D. Bradley Rowe, John A. Andresen, Charles L. Rugh, Robert T. Fernandez, and Lan Xiao. Green Roof Stormwater Retention: Effects of Roof Surface, Slope, and Media Depth. *Journal of Environmental Quality*, 34(3):1036–1044, 2005.

Microcontroller Based Memcapacitor Emulating Platform

Adrian Gomez-Donato

Abstract The overarching goal of this work is to create a memcapacitor emulator platform that is accessible over the internet for anybody who would like to run their own tests on the emulator circuit. A physical circuit that emulates a memcapacitors electrical behavior will be constructed and connected to a microcontroller; the microcontroller will be used to power the circuit as well as read the resulting voltage and current and then plot them on a webserver. The voltage and current values will be plotted over time and the current integrated to find the charge. The idea is to allow the user to control the input voltage frequency as well as amplitude within a reasonable range which would require signal amplification. Hence the user will be given the ability to customize the test they will run and get results as though they were testing on the physical memcapacitor emulator themselves and not through any web interface.

1 Introduction

The memristor, the proposed fourth passive electrical element by Chua in 1971, is an electrical element that links electrical charge with magnetic flux. It behaves as a non-linear resistor controlled by its voltage input as well as current, and it has memory capabilities (hence the name memory resistor) due to the fact that its behavior varies depending on its past voltage and current values [1]. Chua's proposed memristor became a reality in 2008 and was no longer purely theoretical as a device displaying memristive properties was discovered [6]. The memcapacitor and meminductor were then proposed and their behaviors were modeled [2]. A memcapacitor has non-linear capacitance with its capacitance being controlled by the input charge as well as input voltage. Similar to the memristor, memcapacitors have memory capability due to how

Adrian Gomez-Donato
Portland State University Maseeh College of Engineering and Computer Science, 1930 SW 4th Ave # 500, Portland, OR 97201, e-mail: gomezd.adrian@gmail.com

the capacitance is dependent on the previous charge and/or voltage values depending on if it is a voltage or charge controlled memcapacitor model. A true memcapacitor has never been created as only circuits that emulate its behavior exist. There are many circuits that are able to display memcapacitive and memristive behavior [5], but memristors and memcapacitors have yet to be implemented into neuromorphic systems nor have they been used for their memory capabilities to replace RAM with RRAM (Resistive Random Access Memory). Neuromorphic computing, a method of computing in which the brain's structure and way of operation are mimicked, continues advancing and becoming more efficient. This method of computing is most used for machine learning as it is more efficient as well as uses less energy than silicon-based chips. A memristor crossbar or another array with mem-elements such as memcapacitors would be able to do computations alongside store information which essentially allows them to function similar to the neurons and synapses in the brain that do computations as well as store information. Creating neuromemristive systems for machine learning is the next big leap that must be taken in hardware in order to keep up with the rapid progression and advancements happening in the field of machine learning software [4]. Performing tests and viewing the behavior of these mem components as well as their behavior in an array is crucial to advancing in the field and constructing neuromorphic systems. The call to action was the dire need to have a widely accessible mem-element emulating platform. Memchar, a cheap memristor characterization tool, has already been made, and it uses an Arduino in order to measure different parameters and log them into an SD card [3]. It is a far cheaper alternative to characterizing mem-elements than semiconductor analyzers are. Replicating it to run your own tests is not all that difficult; however, it does require the purchase of many individual components including a real memristor which are hard to come by and very expensive for a singular individual electronic element. In this work, a webserver based platform which allows the user to run tests on a real life memcapacitor emulator [5] and receive the results back is created. This will omit the need to construct an emulator which entails purchasing all of the necessary components to recreate the memcapacitor emulator as well as remove the need to purchase an oscilloscope or an arbitrary function generator as that is all taken care of on the backend of the project. This allows for seamless use of the platform without having to worry about breadbording a circuit yourself or having to wait for any components to arrive in order to construct the circuit. Considering the importance of neuromorphic systems as well as RRAM and the part that mem-elements may have in both of them, it is crucial that there be a widely accessible and easy to use platform that does not require someone order a handful of parts and equipment to test and measure the built circuit.

2 Methodologies

LTspice was used to observe the behavior of the proposed memcapacitor emulator circuit [5]. A Raspberry Pi 4 will be used to collect the voltage as well as the current data. The same microcontroller will also be used for the input signal for the emulator circuit.

2.1 LTspice Simulation

The specific circuit used for the physical memcapacitor emulator was chosen due to its simplicity and it being so very replicable [5]. The circuit was replicated in LTspice and simulated to view the relationship between current and voltage as well as charge and voltage. The vast majority of the circuit topology remained unchanged aside from one redesign that had to be made due to the unavailability of the analog voltage multiplier used in the original circuit and thus a replacement one had to be put in its place. In the original circuit there is an MLT04GP analog multiplier, but it would have been too costly to purchase one. The alternative was an AD734AM which is an analog voltage multiplier as well as a divider; this can be viewed in Figure 1. The divisor had to be set to 2.5 to go along with the emulator proposed in the paper [5]. A separate circuit had to be created with its own symbol to set the 2.5 divisor as well as have no subtraction.

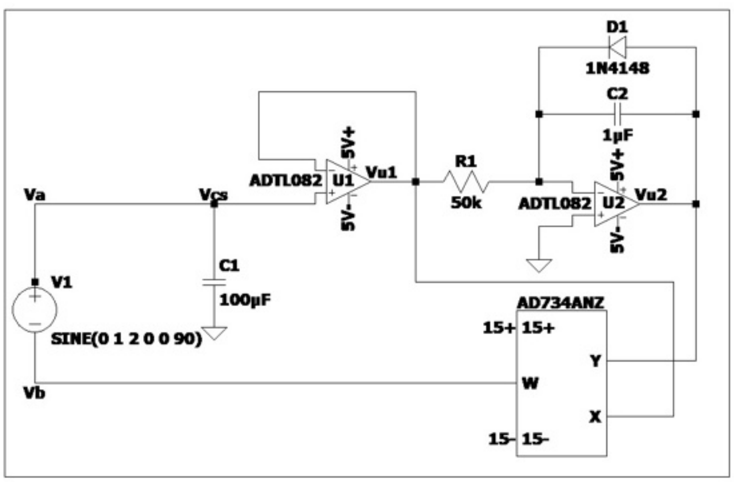


Fig. 1 Schematic used for the LTspice simulation. Identical to the memcapacitor emulator found in a paper with a substitution of analog voltage multipliers/dividers from the MLT04GP to the AD734AMZ [5].

2.2 Data Collection via Raspberry Pi

The Raspberry Pi was used for the creation for the webserver as well as the measuring of the resulting voltages in the circuit via a oscilloscope shield attachment. The oscilloscope used for the raspberry pi is the WPSH206. The VM205 Velleman oscilloscope pascal source code had to be modified in order to have it output a CSV file. Originally, the software would plot the byte data received from the shield, but the code was altered to map the byte data to voltage values. The voltage values would then be written into the CSV file. The data from the CSV will then be plotted onto the website. The Raspberry Pi was used to generate the AC signal used to power the circuit via an Adafruit digital to analog converter (DAC). There are hopes to have an operational multiplier (OPAMP) implemented into the circuit in order to achieve voltages higher than the standard 5 Volts that the Raspberry Pi can output. Another addition that can be made to the system is a method to allow the user to both control the input voltage to the circuit as well as the frequency of the incoming signal. The webserver in place is simply meant to allow the user to alter the input signal parameters such as frequency and amplitude and receive the plotted results after the circuit is allowed to run.

3 Results

Findings alongside data and graphs from the LTspice simulation are presented. The method of integration, the pinched hysteresis loop, as well as the troubleshooting of the denominator disable pin on the AD734 analog voltage multiplier/divier are presented.

3.1 Integration of Current on LTspice

There was no clearcut way to integrate one signal such as the current to observe the charge for instance. The integration occurs during post processing which essentially just means that LTspice waits for the program to end in order to integrate whichever graph you have chosen to integrate. Ideally, you would want to integrate different ranges, so a step function is necessary to increment the range of the integral. Different iterations of the simulation must be ran in order to get sufficient charge points to be able to plot a smooth graph. NGspice has a more straightforward integration method as opposed to that of LTspice. The step function running the same simulation a hundred or so times to get a good graph makes the process take a gruelingly long time.

3.2 Denominator Disable pin on AD734

After implementing the AD734 analog voltage multipler/divider into the LTspice circuit, there were issues in the current graphs. The current would drastically spike on some iterations of the simulations. The spikes occurred when the voltage was at its minimum or maximum value. The setup used for the AD734 was a modified version of one found on the datasheet for the part where the denominator could be set via altering the value of resistors in the circuit. Upon further inspection, the circuit had current going to a denominator disable pin. Upon routing the pin to ground and routing current away from the pin, the spikes occurred far less frequently. Prior to removing the connection to the denominator disable pin, about 20 out of every 100 iterations of the simulation had drastic current spikes. After removing the connection to the denominator disable pin, the current spikes would either never occur in any of the 100 iterations or they would happened in 2 to 3 iterations out of the 100.

3.3 Pinched Hysteresis Loop on LTspice

A pinched hysteresis loop is synonymous with mem element behavior. The pinched hysteresis loop is a display of memristive behavior when current is plotted against voltage as well as memcapacitive behavior when voltage is plotted against charge. In the LTspice simulation of the memcapacitor emulator circuit, a pinched hysteresis loop was achieved when the voltage and the charge were plotted against each other. Aside from integrating the current in the circuit as mentioned in Section 3.1, the measure function must also be used in order to plot the voltage value against the charge value post processing in the LTspice output log.

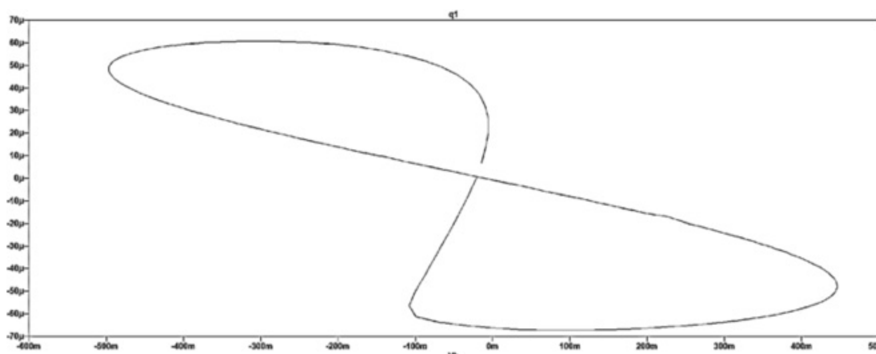


Fig. 2 A graph of the pinched hysteresis loop that occurred when the charge was plotted against the voltage in the memcapacitor emulator circuit on LTspice.

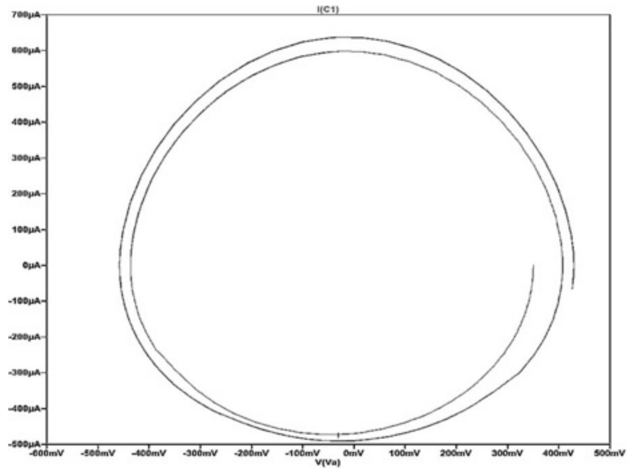


Fig. 3 Current plotted against voltage in the memcapacitor emulator circuit in LTspice [5].

4 Conclusion

This work can be much further expanded upon in the form of creating an array of memcapacitor emulators among other additions that can be made. There is still work to be done in order to complete the memcapacitor emulating platform as well as many parts that can be expanded upon or improved. The emulating platform could prove useful to those who would like to run tests on a memcapacitor emulator and yet do not have the means to do so.

4.1 Applications

The most practical application of the memcapacitor emulator would simply be to fulfil its purpose as a widely, readily available testing platform that allows the user to run their own tests on the circuit. Further testing could be done and the system could be applied to an array of memcapacitors such as a crossbar which is further explained in Section 3.3.

4.2 Concerns and Next Steps

The physical circuit has yet to be completed as there has been trouble getting it to properly work as it did in the LTspice simulations shown in Section 3. The physical circuit must be troubleshooted and pieced together. The most probable next step would be to test out pieces of the circuit individually prior to piecing together the circuit. The OPAMP as well as the analog voltage multiplier/divider are essentially the only two components that would have to be tested separately. Once both are tested and work as intended, the circuit can be put together to create the memcapacitor emulator circuit. It is crucial that the analog voltage multiplier/divider perform the correct multiplication of the two input signals as well as the correct division. As mentioned earlier in Section 2.2, a system that would allow customization of the input signal such as its amplitude as well as its frequency must also be created. This modification would be necessary as it would then allow the user true freedom in conducting their tests. This modification would require the use of an OPAMP to amplify the signal to voltages higher than 5 Volts and yet still low enough as to not damage the circuit. Another step to be taken would be to create an array of memcapacitor emulator circuits such as a crossbar in order to simulate an array of the memcapacitors instead of just a singular one. This will come much further down the line as it would require that many other memcapacitor circuits be built as well as the implementation of a different voltage measurement as the WPSH206 oscilloscope shield only has one oscilloscope input. In order to measure the different intersections on the memcapacitor crossbar, a voltage measurement system that allows more than one input would be necessary. A concern of mine relates to the crossbar configuration. A completely different method would have to be used to measure the voltages such as an analog to digital converter for the Raspberry Pi's GPIO pins.

Acknowledgements The work was supported by the Daimler Truck Portland Professorship. Without Dr. Teuscher's guidance, this work as well as this chapter would not have been possible. Special thanks to Dr. Teuscher for making this work possible and for providing I, alongside all his other interns with a unique, rich learning experience.

References

1. L. Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.
2. Massimiliano Di Ventra, Yuriy V. Pershin, and Leon O. Chua. Circuit elements with memory: Memristors, memcapacitors, and meminductors. *Proceedings of the IEEE*, 97(10):1717–1724, 2009.
3. Khaled Humood, Heba Abunahla, and Baker Mohammad. Memchar: Portable low-power and low-cost characterization tool for memristor devices. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2022.
4. Huihui Peng, Lin Gan, and Xin Guo. Memristor-based spiking neural networks: cooperative development of neural network architecture/algorithms and memristors. *Chip*, 3(2):100093, 2024.
5. M. P. Sah, C. Yang, R. K. Budhathoki, H. Kim, and H. J. Yoo. Implementation of a memcapacitor emulator with off-the-shelf devices. *Elektronika ir Elektrotechnika*, 19(8):54–58, Sep. 2013.
6. Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. The missing memristor found. *Nature*, 453:80–83, 2008.

CNN-Powered Resume Analysis for Keyword Relevance and Industry Suitability

Alena Dasha Peethala and Dr. Christof Teuscher

Abstract Most resumes don't reach hiring managers or recruiters until they pass through an automated scanning system, which can sometimes lead to qualified candidates being overlooked. This paper presents a Convolutional Neural Network (CNN) which assesses resumes for their suitability for specific jobs or industries based on keyword prevalence. The CNN model analyzes text to extract important keywords relevant to different industries, utilizing a dataset of over 2,000 resumes across 30 industries. This work demonstrates how CNNs, traditionally used for image recognition, can be effectively applied to text classification tasks. By adapting methods from Yoon Kim's research [2] on applying CNNs to text classification and models such as TextConvoNet [4], this study shows that CNNs can provide an accurate and automated approach to resume screening. The results indicate that this method aids in identifying key skills and experiences specific to various common industries, allowing job-seekers to utilize a model such as this to ensure their resumes are relevant and tailored to their target industry before submitting an application.

1 Introduction

In today's competitive job market, many resumes first encounter Applicant Tracking Systems (ATS) or other automated tools before hiring managers consider them for review. These systems scan resumes for specific keywords, skills, or other indicators of suitability. As a result, candidates must ensure their resumes include the right keywords to pass these initial screenings.

Alena Dasha Peethala
Portland State University, Maseeh College of Engineering and Computer Science, 1930 SW 4th Ave 500, Portland, OR 97201, e-mail: alenadashi@gmail.com

Dr. Christof Teuscher
Portland State University, Maseeh College of Engineering and Computer Science, 1930 SW 4th Ave 500, Portland, OR 97201,e-mail: teuscher@pdx.edu

Job seekers typically receive advice to tailor their resumes to their desired industry by researching companies, analyzing job descriptions, and reviewing company values or mission statements. However, this manual approach is time-consuming. Growing interest in automated solutions aims to simplify this process, particularly due to the rise in popularity of artificial intelligence and large language models (LLMs).

Recent advancements have introduced various methods to enhance resume screening which employ machine learning and natural language processing techniques. Notable research, such as Yoon Kim's work on text classification with Convolutional Neural Networks (CNNs), demonstrates the potential of such models for text analysis [2].

This study presents a simple CNN model that evaluates resumes for job suitability based on keyword prevalence, serving as an initial check for people wanting to understand how well their resume matches a certain industry. Trained on a dataset of over 2,000 resumes from 30 different industries, the model performs text analysis to identify and match keywords relevant to various industries and specific jobs within those industries. While the model is straightforward, it establishes a foundation for demonstrating how CNNs, traditionally used for computer vision and image-based tasks, can also apply to text-based tasks to efficiently understand relationships between words. The model's performance results (an accuracy of 86%) indicate that a simple model can offer an automated solution for resume screening and job matching, showing that a similar approach might improve other text-based tasks.

2 Related Work and Background

Convolutional Neural Networks, also called CNNs or ConvNets , are deep learning algorithms designed to assess images and assign importance to different parts of the image, with the primary goal of analyzing and identifying patterns. These patterns serve many purposes and types of data, making CNNs particularly useful for computer vision tasks [3]. Convolutional neural networks are typically associated with image-based tasks. Unlike most image-based tasks, understanding a sentence often relies on grasping its context, such as the words surrounding a certain keyword. Models like recurrent neural networks were considered more suitable for these tasks.

However, despite their traditional use with images, researchers have tested CNNs for text classification, and this approach is gaining popularity. Zhang, Zhao, and LeCun [5] conducted an empirical study on character-level convolutional networks for text classification, demonstrating their ability to achieve optimal results. More recent research includes Yoon Kim's paper [2] on using CNNs for sentence classification, which represents one of the novel approaches for applying convolutional neural networks to text-based tasks. Kim's model, utilizing GloVe pre-trained word embeddings, applies a single convolution layer to a CNN model for sentence classification, producing impressive results and accuracy. Kim's paper also influences Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore [4], who describe TextConvoNet, a CNN that, unlike Kim's model, employs two convolution layers for increased complexity and optimal results.

3 Data and Methodologies

To gather data for this project, I used a public repository from Kaggle [1] and created word clouds with term-frequency inverse document frequency (TF-IDF) to visualize the most frequent keywords in each industry. I plotted the differences in loss and accuracy on the training set versus the testing set using matplotlib. This method helped me analyze model performance and track overfitting or underfitting, allowing me to make changes based on these results.

3.1 Data Collection

The first type of data used for this project is a CSV file, containing large amounts of resume data. The dataset used for this project is a public dataset from Kaggle, titled "Resume Dataset," which was created by Snehaan Bhawal. This dataset contains resumes from various industries and fields, categorized into 30 different areas, such as Accounting, Engineering, Healthcare, Information Technology, and Sales, among others.

SR. HR CONSULTANT Executive Profile Ambitious Human Resources professional who creates strategic alliances with org	HR
HR SHARED SERVICES ANALYST Summary Versatile HR professional with a strong benefits administration background, pr	HR
VOLUNTEER HR -IVOLUNTEER Summary Sponsorship not required to work in the US A successful Human Resources Profess	HR
GLOBAL HR MANAGER Summary A Global HR Professional with 10+ years' progressive experience across industries and	HR
HR & SAFETY MANAGER Summary Human Resources Manager Certified Professional in Human Resources (PHR) Extensiv	HR
HR SPECIALIST (INFORMATION SYSTEMS) Experience 02/2013 - 12/2014 Company Name – City , State HR Sp	HR
DIRECTOR OF HR Executive Profile Ambitious Human Resources Generalist who creates strategic alliances with organiz	HR
HR SENIOR SPECIALIST Career Overview Dedicated Service Representative Professional motivated to maintain customer s	HR
HR CUSTOMER SERVICE REPRESENTATIVE Summary Excellent team player with legal background and abilities to interpret	HR
HR SERVICES REPRESENTATIVE Summary A multi-skilled professional with good all-round HR imformatory skills. Very capa	HR
HR ASSISTANT INTERN Summary New graduate seeking work as a Counselor able to facilitate both individual and group th	HR
HR BENEFITS/LEAVE COORDINATOR Summary 13 years of Human Resources experience and 27 years of administrative exp	HR
REGIONAL HR BUSINESS PARTNER Human Resources Professional Executive Profile Business-savvy, results-driven, and	HR
HR PERSONNEL ASSISTANT Summary I am a U.S. citizen who is authorized to work in the US for any employer. I have wor	HR
EXECUTIVE ASSISTANT HR Summary Skillful and dedicated Executive Assistant with extensive experience in the coordinati	HR
SENIOR HR MANAGER Professional Summary Results-driven and business-oriented professional with strong experience in	HR
HR MANAGER/BUSINESS PARTNER Summary A Human Resources Business Partner with extensive experience aligning HF	HR
DESIGNATION: HR ASSISTANT Professional Summary Human resources coordination and management professional offerin	HR
HR VOLUNTEER ASST. MANAGER Professional Summary I am dedicated to every project I have worked on with strong unde	HR
HR SPECIALIST/ HORIZONTAL ENGINEER Professional Summary Passionate HR Specialist with over 10 years' extensive ex	HR
HR CONTACT CENTER SPECIALIST Summary Forward-thinking professional with various experience in human resources, sal	HR
DESIGNER Summary Designer with more than 15 years in product design, manufacturing, exhibit design and visual merch	DESIGNER
DESIGNER Summary Established well-rounded Designer with a reputation for exquisitely designed collections, who consi	DESIGNER
DESIGNER Summary To get a strong foothold on the career ladder by doing the best I can and more, with a company that s	DESIGNER
DESIGNER / TECHNICAL DESIGNER Summary Creative fashion designer with background in the catagories of swim, intim	DESIGNER
MULTIMEDIA DESIGNER AND GRAPHIC DESIGNER Portfolio www.Artisterymedia.wix.com/creativeflow Summary A har	DESIGNER
GRAPHIC DESIGNER Summary A graphic designer, who is creative and detail-oriented; who thinks a lot, but wants to make	DESIGNER
E-LEARNING DESIGNER Career Overview Highly skilled and experienced educator with a strong	DESIGNER
GRAPHIC DESIGNER Summary Enthusiastic student majoring in Chemistry; great at performing many task in a timely ma	DESIGNER
TECHNICAL DESIGNER Summary SPECIAL QUALIFICATIONS: Textile Engineering Knitting and Garment Manufacturing Textile	DESIGNER
MECHANICAL DESIGNER Summary I am a current Mechanical Designer for I.A.S. I am extremely versatile, reliable and effic	DESIGNER
WEBSITE DESIGNER Summary Software developer well-versed in the entire workflow for developing and implementing w	DESIGNER
FLORAL DESIGNER Summary I have been involved in the Retail Industry for over 44 years. In those years I have been em	DESIGNER
INSTRUCTIONAL DESIGNER Summary Tamantha Uebele is a Home Health Clinical Analyst and Instructional Designer with	DESIGNER
INSTRUCTIONAL DESIGNER Summary Dependable and resourceful Instructor/Trainer, Technical Writer and Instructional D	DESIGNER
TECHNICAL DESIGNER Career Overview	DESIGNER
INSTRUCTIONAL DESIGNER Summary Motivating and talented Instructional Designer driven to inspire students to pursue	DESIGNER
PRODUCT DESIGNER Professional Summary 4-5 years engineering experience and 1-2 years working experience. Able t	DESIGNER
FORMS DESIGNER Professional Summary Professionally trained [job title] with experience ensuring high standards of cu	DESIGNER
INFORMATION DESIGNER Summary of Qualifications Strong leadership, project management, System Administration and te	DESIGNER
INDEPENDENT DESIGNER Professional Summary Independent and results driven Design Professional with an extensive b	DESIGNER
GRAPHIC DESIGNER Experience Graphic Designer January 2014 to January 2015 Company Name – City ,	DESIGNER

Fig. 1 CSV File Containing Raw Resume Data for Human Resources Industry

3.2 Data Visualization

To visualize the data, I created word clouds in Python. This involved removing stop words, such as pronouns and other non-industry-specific words, to highlight relevant keywords. Word clouds were generated for entire industries and specific jobs within these larger industries as well, helping to visualize the most frequent and important keywords for these roles. One of the most important aspects of creating the word clouds was term-frequency inverse document frequency, also called TF-IDF, where word vectors assign a score to various words depending on their frequency within a group of text, or, in this case, across the cleaned CSV file, after all punctuation and other stop words were removed.

3.2.1 Data Visualization Section 1

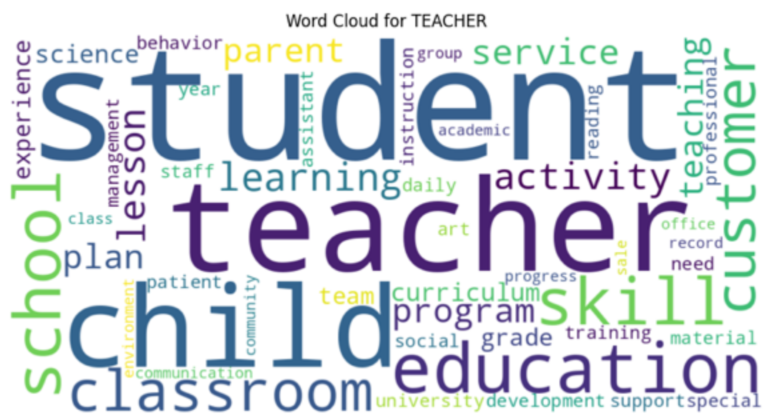


Fig. 2 Word Cloud for Teachers

3.3 Model Performance Analysis

Finally, after training the model, matplotlib was used to create graphs that visualized the differences in loss and accuracy metrics between the training and testing sets over a series of epochs. These visualizations were crucial for understanding if the model was overfitting or underfitting and for determining the optimal number of epochs for the model's best performance, especially due to the various hyperparameters tested during this process.

3.3.1 Model Performance Analysis Section 1

3.4 Methods

The project was initially developed using Google Colab, which provides a cloud-based Python environment. For local development and increased memory capacity, the project was transitioned to Visual Studio Code (VS Code). The back end of the system is implemented in Python, with TensorFlow and Keras used for constructing and training the Convolutional Neural Network (CNN) model. Scikit-learn facilitates additional machine learning tasks, while Pandas is employed for data pre processing. Numerical computations and array manipulations are handled with NumPy, and data visualization is performed using Matplotlib. The front-end interface of the project is built using Flask and HTML, allowing users to upload their resumes and receive analyses regarding industry suitability. The model is trained on a dataset containing over 2,000 resumes, categorized into 30 distinct job categories.

Upon loading the CSV file containing the resume data into a pandas DataFrame, the text data undergoes a pre-processing pipeline. Text is first converted to lowercase to ensure uniformity, as capitalization does not impact the meaning of most words. Punctuation is removed to clean the text. Lemmatization is performed using NLTK's WordNetLemmatizer to reduce words to their root forms. Subsequently, the text is tokenized into sequences of integers, which are then padded to maintain a consistent input size for the model. Categories are encoded as integers and subsequently one-hot encoded for multi-class classification.

To facilitate data visualization, word clouds are generated using Term Frequency-Inverse Document Frequency (TF-IDF) to illustrate term frequency distributions. The word cloud function is designed to account for bigrams and trigrams, capturing multi-word skills. Given the dataset's diversity within categories, such as including both UX designers and fashion designers under the broad 'designer' category, specific word clouds are also created for individual job titles to provide more granular insights.

The training dataset is resampled using Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance, as the dataset had far more information technology and business resumes than resumes in industries such as fitness or teaching. The dataset is then split into training, testing, and validation sets. The CNN model, which is a sequential model, consists of several layers: an embedding layer that converts word indices into dense vectors, a convolutional layer with a specified number of filters and kernel size, a global max pooling layer to reduce dimensionality, a dense layer with dropout for regularization to reduce over-fitting, and an output layer with softmax activation for multi-class classification purposes.

Hyper-parameter tuning for the model was extensive, conducted using Optuna to explore various combinations of optimizers, epoch sizes, batch sizes, kernel sizes, dropout rates, and activation functions. An Optuna study was run for 100 trials to determine the optimal hyper-parameters. The final selected parameters include an embedding dimension of 100, 182 filters, a kernel size of 4, 196 dense units, a dropout rate of 0.4729, and a 'tanh' activation function.

The model was compiled using the RMSprop optimizer and categorical crossentropy loss function. Performance metrics, including accuracy and loss, were visualized and plotted for both training and testing sets using matplotlib, with these plots saved as images. Finally, the trained model was saved for future use and for integration into the front-end.

The front-end of the project is developed using Flask and HTML to provide a user-friendly interface for interacting with the resume analysis model. Flask is chosen for its simplicity and flexibility, allowing for faster development and easy integration with the back end components of the project. Flask manages web requests, handles user inputs, and coordinates the interaction between the front-end and the model.

HTML is used to design the structure and layout of the web interface. It allows users to upload their resumes through a straightforward form interface. The form submission triggers a Flask route, which processes the uploaded resume file, sends it to the back end for analysis, and returns the results to the user. This ensures that users can easily interact with the model and receive feedback on their resumes' suitability for various industries.

3.4.1 Method Subsection 1

Word clouds were created by using tf-idf to extract the most common keywords from resumes from each industry. Word clouds are images were the larger a word appears, the more frequent it is within that specific industry, meaning it is an important keyword. To do this, TF-IDF is used, as this method measures the importance of the various keywords within documents or other files The word clouds are a crucial aspect of the CNN architecture in this project. Initially, word clouds were created with 1-grams only, however, as seen below, later 2 grams and 3 grams were specified in the function creating these word clouds, in order to visualize a larger variety of skills, many of which are more than 1 word. For example, the skill "six sigma" appears as so with this specificity, instead of just appearing as the word "six." The impact of specifying bi-grams and tri-grams in the function can be seen in figure 3.

3.4.2 Method Subsection 2

Hyper parameter tuning, which involves experimenting with various hyper parameters such as number of epochs, optimizer type, learning rate, and batch size, among others, was an important aspect of developing this CNN and ensuring that the model performance was to the best of its ability. The first time model building was completed, hyper parameters were adjusted manually solely within a history=model.fit() function, specifying a certain number of epochs, a certain batch size, and implement-

Fig. 3 Impact of Bi-Grams and Tri-Grams on Word Clouds

ing other strategies such as early stopping and patience, so that the model would stop training if the loss continued to increase over a specified number of epochs, or if the accuracy continued to decrease over a specified number of epochs. However, this manual process proved to be very time-consuming and inefficient, as I did not realize at first that even changing one hyper parameter and leaving the rest untouched could still did not necessarily mean that those other parameters would be unaffected. After researching methods to automate this process, such as GridSearchCV, Bayesian optimization through Optuna, which is compatible with keras, tensorflow, and sci-kit learn, all libraries used in this code, was concluded to be the most efficient option for this process.

3.4.3 Method Subsection 3

Figure 4 shows the basic architecture of this CNN, created using the plot model function from the Keras library. This figure illustrates the model's various layers and connections. "None" represents the batch size, not fixed during model definition but instead a placeholder for the actual batch size used while training the model, and the various numbers, such as "196" represent the dimensionality of the output for that particular layer.

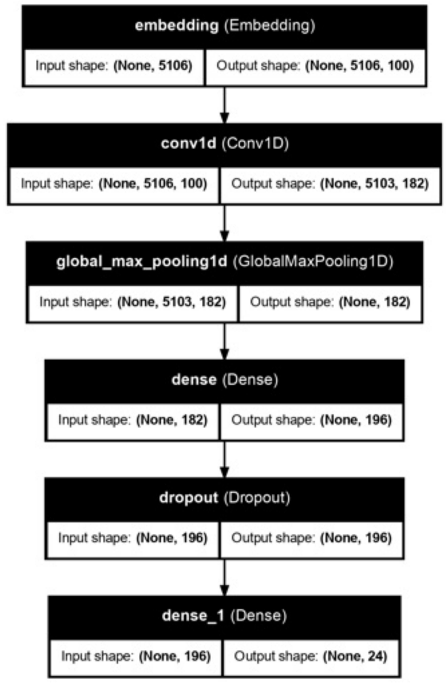


Fig. 4 CNN Architecture

3.4.4 Method Subsection 4

Though the back-end of this project is created fully with Python, the front-end was created with Flask API and HTML. Flask handles web requests and user inputs on the front-end and HTML is used for the structure and layout of the user interface, as seen in Figure 5 and Figure 6.

Fig. 5 Prototype of Front-End Where Users Can Upload Their Own Resumes, Created with Flask and HTML

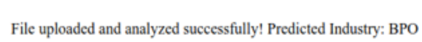


Fig. 6 Example of Uploading My Own Resume into User Interface

4 Results

In this study, model performance was visualized through graphs which depict the differences in model accuracy and model loss over the training and testing sets. This helps to understand if the model is overfitting or underfitting, as the model should ideally should perform well on data it has been trained on, but also perform just as well on data it has not seen before. As displayed in the methods section, hyper parameter tuning was performed to understand which combination of hyperparameters achieved the highest accuracy. After experimenting with various combinations of epochs, batch sizes, optimizer types, learning rates, and varying model architecture, such as incorporating batch normalization or dropout layers into the CNN architecture to prevent over fitting, the best accuracy and loss metrics over both the training and testing sets, which showed the most correlation, are shown in figures 7 and 8, and achieved an accuracy of 86%.

4.0.1 Results Subsection 1

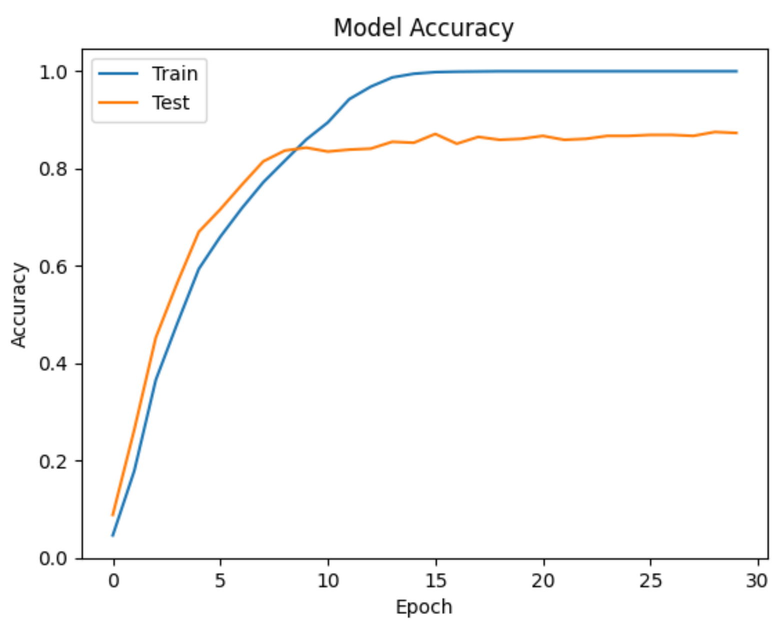


Fig. 7 Final Plot Showcasing Model Accuracy Over Training and Testing Set

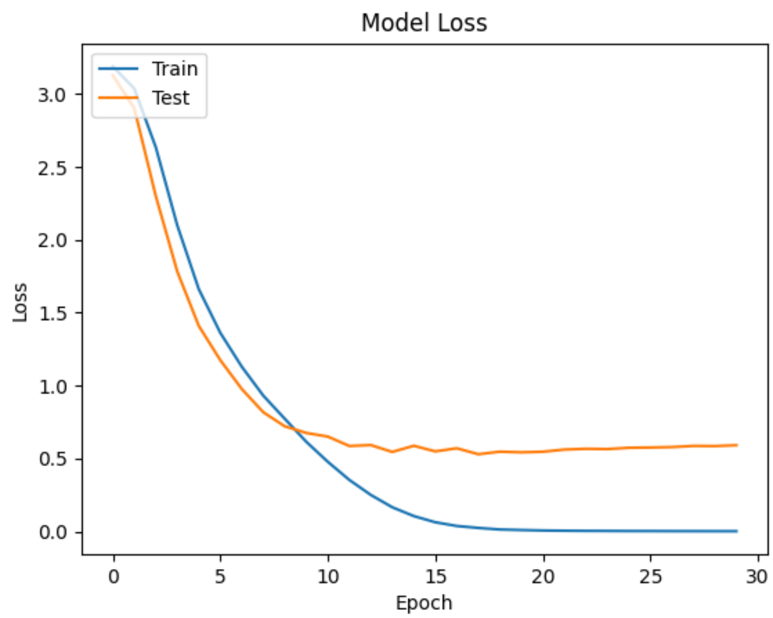


Fig. 8 Final Plot Showcasing Model Loss Over Training and Testing Set

5 Conclusion

One of the most important applications of this project is its exploration of the suitability of a convolutional neural network for text-based tasks. The project shows how CNNs can be well-adapted to text analysis as a result of their ability to recognize patterns in a unique way, and how they can be used to solve or automate processes within industries.

5.1 Applications

One significant application of this project is in automated resume screening. This tool can be adapted by and further developed by HR departments and recruitment agencies to streamline the recruitment process, significantly reducing the time spent manually reviewing each application by automatically identifying resumes that match job openings based on the presence of key skills and experiences. However, since most large companies and corporations have advanced scanners in place, perhaps the most important application of this tool would be for everyday job seekers who wish for an additional channel in which to test the suitability of their resume for their intended position. People can adapt tools like this before submitting their resumes. They can also use this tool to understand their most prevalent skills or to see potential industries they could work in based on their skills. In this way, this tool can provide a kind of personalized career guidance for people seeking jobs. Career services at universities and career counseling centers can also leverage this technology to assist students and job seekers in tailoring their resumes to specific industries or roles, increasing their chances of employment. Finally, this tool can also be used for skill gap analysis, allowing organizations to analyze their workforce's skill sets, identify gaps, and plan appropriate training programs accordingly.

5.2 Concerns and Next Steps

Although this paper presents a simple CNN model, the project will be expanded in the future. The model is limited in several ways, including the fact that the resumes this model was trained on is not fully representative of all existing jobs or industries. Additionally, there is no mechanism currently which accounts for hard and soft skills alike for certain industries, which is definitely an important aspect to consider for most industries. The next steps include training the model on further resume data, and also additional data that provides valuable insights into job history and skill sets, such as CVs or LinkedIn profiles. This will help the model adapt better to various jobs and individuals. In terms of model metrics, while accuracy and loss are generally correlated between the training and testing sets, there is a pattern where the training set performs slightly better after a certain point, while the testing set plateaus. This indicates minimal overfitting, seen in both figure 7 and 8. To achieve more accurate results and improve model accuracy correlation, the model will be trained on more data in the future, such as on new resumes uploaded through the UI. Furthermore, implementing a transformer model, such as BERT for multilingual resume analysis, would extend the model's usability to a wider audience, as it currently only scans resume that are fully in English. Another key development will be making the model more interactive by providing personalized career guidance and resume improvement suggestions. Finally, publishing the model to GitHub or another platform will enhance its accessibility and usability.

Acknowledgements Thank you to Dr. Christof Teuscher for your mentorship and guidance throughout this whole project, as well as this opportunity. Thank you to the creator of the Resume Dataset on Kaggle, Snehaan Bhawal. This work was supported by the Daimler Truck Portland Professorship.

References

1. Snehaan Bhawal. Resume dataset. www.kaggle.com/datasets/snehaanbhawal/resume-dataset, 2021.

2. Yoon Kim. Convolutional neural networks for sentence classification. *arXiv*, 2014.
3. Sumit Saha. A comprehensive guide to convolutional neural networks: The eli5 way, 2023. Accessed: 2024-07-28.
4. Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore. Textconvonet: A convolutional neural network based architecture for text classification. *arXiv*, 2022.
5. Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *arXiv*, 2016.

Capturing Active Transportation Counts at Intersections Using Ultralytics YOLOv8 Image and Video Tagging

Alicia Hopper, Tammy Lee, and Sirisha Kothuri

Abstract Current programs of collecting active transportation counts are manual and resource intensive processes. To help reduce the time and effort associated with active transportation count programs, more people are turning to automated systems. In this paper, we propose a computer vision-based solution for efficiently capturing active transportation (e.g. bicyclist and pedestrian) counts at signalized intersections. We used the EuroCity Persons Dataset to train an Ultralytics YOLOv8 model intended to generate counts of non-motorized traffic at intersections from previously obtained video. Preliminary training results show a mean Average Precision at IoU threshold 0.5 (mAP50) of 0.553 when identifying pedestrians in training data, and a mAP50 of 0.443 when identifying bicyclists, as well as a mAP50 of 0.348 for people on scooters in training data. The model is intended to be used on pre-existing video data from intersections. This approach offers a promising low-cost alternative to labor-intensive manual counts, allowing for more informed decision-making opportunities without a large drain on resources.

1 Introduction

In recent years, there has been increased interest in more pedestrian-friendly urban environments in the United States. In order to make improvements to the existing infrastructure, it is useful to know usage trends, the study of which requires large amounts of data surrounding pedestrians, bikes, and other non-motorized counts at various locations. This can include counting people crossing a crosswalk, or on a sidewalk in a given location. Collecting this data can prove time-consuming and challenging using many current strategies, such as manual counting [13].

In an attempt to reduce the cost of collecting such data, many are turning to automated systems. While there are many options on the market, such as inductive

Alicia Hopper
Carleton College, Northfield, MN, USA e-mail: hoppera2@carleton.edu

loops, piezo-electric pads, and various infrared counters and laser scanners [8] [7], in recent years there has been much progress in the field of image tagging using neural networks such as the Ultralytics YOLO (You Only Look Once) model [6]. In this project, we use the YOLOv8 model and the EuroCity Dataset [1], along with video taken at intersections in Portland, Oregon, for a previous project [8], to build and analyze a low-cost, efficient solution to extract counts of pedestrians, bicyclists, and other non-motorized traffic from existing video data. This model can be easily deployed given other video data, allowing transportation agencies an efficient way to capture non-motorized counts.

2 Related Work and Background

Estimating and modeling non-motorized traffic, such as pedestrians and bicyclists, is far from a novel concept. There are a number of different methods commonly utilized to extract these counts, each associated with its own advantages and disadvantages. Manual counts are often regarded as the most accurate method [2], especially when video is taken and manually reviewed to count each instance. The counts taken from this kind of data have many uses, such as building models to estimate the volume of traffic over a longer period of time [11] [12].

However, manual counting techniques can take up to three times the length of video taken to extract these counts [2]. Given the amount of time required to obtain sufficient data using this methodology, there has been an increase in automated devices intended to obtain the counts without the need for such labor-intensive manual counts [7] [8]. However, despite the conclusion that some devices may offer reasonable estimates of pedestrian traffic, many of these devices are costly, and few offer the reliable ability to differentiate between pedestrians, bicyclists, and people using other non-motorized devices, such as scooters.

The rise of computer vision has introduced a promising alternative to expensive devices or labor-intensive manual counts. Many studies show promising results when using neural networks or other computer vision technology to extract counts from video data [4] [5] [3] [14], indicating that taking video data and feeding it into a neural network or algorithm could be a reliable, low cost alternative to other counting strategies.

The Ultralytics YOLO model is a popular, open source neural network model which is used for image tagging, video tagging, along with other topics related to computer vision. The current model, YOLOv8, is free to use, and fairly user friendly for those without extensive knowledge of neural networks. These features make it a promising contender for use in identifying and counting non-motorized traffic, given sufficient video data. Additionally, the use of an open source model allows researchers to use CCTV systems already in place for traffic monitoring and not have to rely on adding or purchasing more hardware. Tracking multiple pedestrians using the YOLO model has been shown to be successful [10] [16]. In this paper we look to use the YOLO model to track, differentiate, and count non-motorized traffic in crosswalk regions from video data taken at intersections, to produce a tool that can eliminate the need for multiple automated counters or manual counting methods.

3 Data and Methodologies

In order to train and validate a YOLOv8 model, a large amount of video data were required. The following sections detail the types of data used, as well as the methodologies used to train and validate the model.

3.1 Data Collection

There were two types of data used for this study: data used to train the YOLO model, and intersection video data to test the model's efficacy in the intended use situation. In order to train the model, data from the EuroCity persons dataset [1] were used. In order to test the model against a possible usage scenario, video data and associated manual counts from intersections in Portland, Oregon were used. In 3.1.1, a brief overview of the EuroCity persons dataset is provided, and in 3.1.2, the data and method of collection for the intersection video data is explained.

3.1.1 EuroCity Dataset

The Eurocity dataset consists of over 47,300 images taken in urban scenes, annotated with bounding boxes which specify the difference between a pedestrian and a "rider". These annotations include a part-based annotation of the non-motorized vehicles, classified into one of seven object classes, including bicycles and scooters. The large amount of data available in this dataset, combined with the variety of cities and weather conditions captured, make it very useful for training a YOLO model.

3.1.2 Validation Data

In order to validate the model, we used video that was taken at intersections in Portland, Oregon and used for a previous study [9]. This includes an average of 48 hours of video taken at each of 49 intersections in the Portland metro area. Manual counts were extracted from this video. The process of extracting these counts included student researchers watching 2,279 hours of video, and manually noting each crossing event. The data was then validated by a trained researcher, by randomly selecting at least 5% of all crossing events at each location, and if systematic issues were identified, correcting them and checking additional entries. Finally, automated checks were run to ensure that dates, timestamps, and time differences were within the expected range, manually correcting any errors that were found by these checks [9]. In our study, the video and associated counts were used to validate and test the model's performance.

3.2 Methods

This project was conducted in Python 3.10.12, using the Ultralytics YOLOv8 model, which can be downloaded as part of the Ultralytics Python package available at https://docs.ultralytics.com/quickstart/#install-ultralytics. The YOLOv8 model was selected for its ease of use and success in past research at multi-object tracking.

3.2.1 Model Training

In order to train the model, we first used a set of images taken in the daytime and their respective annotations from the EuroCity persons dataset. A python script was developed to convert the original EuroCity persons annotation files into an acceptable format that could be used with the YOLOv8 model. Because the Eurocity Persons Dataset includes annotations and bounding boxes for items which are over 10% occluded, which were found to be difficult to train a YOLO model on, these items were omitted for initial training. . We also omitted from the training image set any items labeled "far-away", since the intended use of this model did not include categorizing objects which are far away from the video camera. These changes were made with the intention that the model could be trained on these items in later research to improve accuracy. Additionally, only "day" images were used, since the model was intended to review data taken in the daytime.

After this, 600 training epochs were run on a machine with equipped with a GPU, until sufficiently high metrics of accuracy and precision were acquired. Since the EuroCity persons data were already divided into train, test, and validated images, the train images were used for training, with the test images used after each training epoch to evaluate performance so far. Given the limitations of storage on the machine used for training, the data were split into six roughly equal batches, and training rotated through these batches, with 100 epochs of training for each batch.

3.2.2 Model Testing

After the model was trained, it was intended to be tested on the video data taken from Portland intersections. In order to do this, YOLO's built-in tracking capacities were utilized in a program written which, given user-selected regions showing crosswalks to count, produces counts over the period of video for the totals of all non-motorized traffic, as well as number of pedestrians, bicyclists, scooters, and wheelchairs.

4 Results

The model was evaluated for performance and accuracy against validation data from the Eurocity Persons Dataset, but difficulties managing graphics packages led to issues with the intended validation against intersection video data.

4.1 Initial Model Accuracy

During the training phase, the model achieved precision values of 0.655 in detection of pedestrians, 0.671 for bicyclists, and 0.72 for scooter users. A precision value measures the accuracy of the detections, by calculating the proportion of real items detected among all item detections. Thus, a higher precision value indicates that the model is avoiding false positives.

The model achieved recall values of 0.528 in detection of pedestrians, 0.401 for bicyclists, and 0.259 for scooter users. A recall value measures the proportions of real items detected among all items present in the ground truth labels from the Eurocity persons dataset. Thus, a higher precision value indicates that the model is avoiding false negatives, and detecting a higher percentage of the items.

A summary of all training metrics over the first 100 epochs of training in Figure 1 shows that the model followed the expected improvements during this time, training effectively.

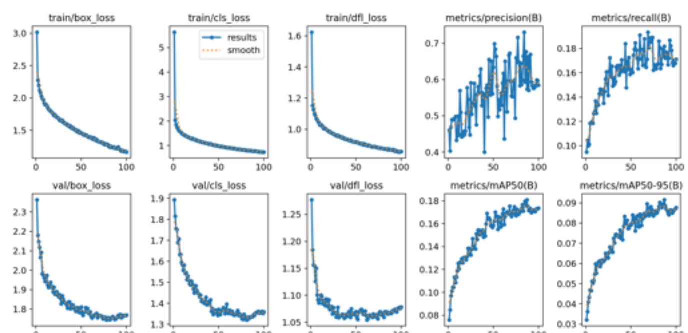


Fig. 1 Training metrics for all classes, including precision, recall, mAP50 and mAP50-95 over the first 100 epochs of training.

The mAP50 metric is calculated based on the average precision (AP), the area underneath the precision-recall curve, at a threshold of 0.5 for the Intersection over Union (IoU), the overlap between the ground truth bounding box and the predicted bounding box. According to the official Ultralytics website, "mAP extends the concept of AP by calculating the average AP values across multiple object classes. This is useful in multi-class object detection scenarios to provide a comprehensive evaluation of the model's performance" [15]. Essentially, the mAP score takes into account both precision and recall, as well as the accuracy of the bounding box localization. The mAP50 values of the model were 0.553 for pedestrians, 0.443 for bicyclists, and 0.348 for scooter users.

The precision-recall curve in Figure 2 calculated after 6 batches of training shows that the model retained a fairly high value for precision in the class of pedestrians before a recall value of 0.5. This was expected because the training images included a larger number of pedestrians than any other class.

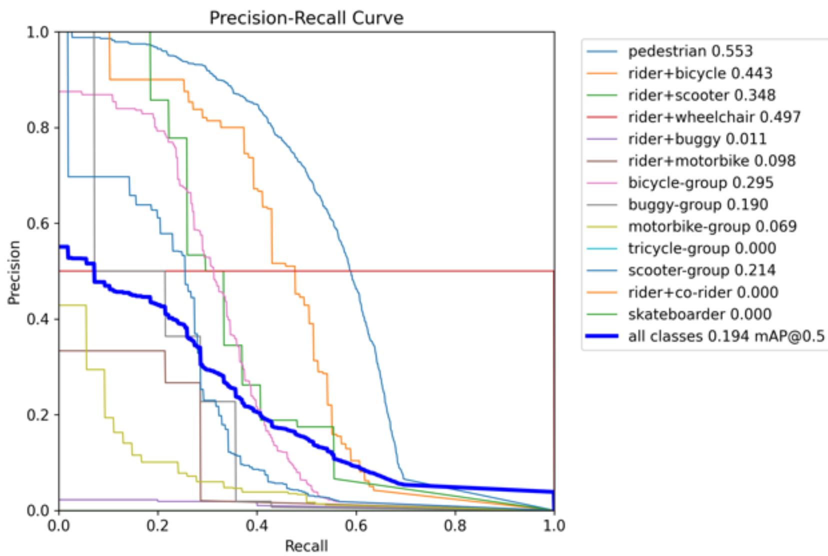


Fig. 2 Precision-Recall curve of the final model, after training. The curve was calculated using the validation data in batch 6, although similar behavior is found using validation data from the other batches.

4.2 *Model Accuracy in Video Data*

Unfortunately, due to time restrictions and issues loading python graphics packages inside of a remote server, the model was not able to undergo successful testing on the video data. While the video was being processed, it was clear that detections were being made, from the output in the terminal, as shown in Figure 3. These detections should then be added to the video file, and a running count of detections are expected to be visible on that video footage.

However, there were issues displaying the video, which caused the counting region to be incorrect, along with some other issues surrounding image scale and annotation display. Since the counter function is part of the Ultralytics python package, and the machine used for this project was unable to use the default display option, this is left to later work.

```
0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 1 pedestrian, 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.5ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 1 pedestrian, 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.4ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 1 pedestrian, 6.0ms
Speed: 2.5ms preprocess, 6.0ms inference, 1.4ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 1 pedestrian, 5.7ms
Speed: 1.0ms preprocess, 5.7ms inference, 0.5ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 (no detections), 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.2ms postprocess per image at shape (1, 3, 448, 640)

0: 448x640 1 pedestrian, 5.7ms
Speed: 0.7ms preprocess, 5.7ms inference, 0.4ms postprocess per image at shape (1, 3, 448, 640)
```

Fig. 3 Terminal output when running a program designed to extract counts from video data, using Ultralytics YOLO solutions counter and model tracker.

5 Conclusion

The model trained in this study is incomplete – typically higher values of mAP50 are expected for models with these uses. The goal was that this model could still perform well on the relatively easy detection in the video data, however, time constraints and other limiting circumstances made this difficult to verify. Still, we find that this is a promising system and process to follow, and with further training, especially on items less common in the Eurocity datset, such as scooters, could achieve higher accuracy. The following sections discuss applications, concerns and next steps for the research.

5.1 Applications

This work has many potential future applications. Although this project focused on using video data from Portland, the model was trained on data from 31 European cities, and can therefore be used in other cities to validate manual counts. Beyond the validation of manual counts in intersections, it could also be used to validate and help calibrate other types of sensors or automated counters meant to count pedestrians and bicyclists. This can help to test these other devices, and create a more robust system.

The counts extracted from this type of research are useful because counts from shorter periods of time, in conjunction with long-term models, can be used to predict the volume of traffic over larger periods of time, helping with infrastructure planning, traffic modeling, and identifying areas where safety concerns should be addressed.

5.2 Concerns and Next Steps

Due to time constraints, there were some steps left uncompleted at the end of this project. Further research is required to further test the model on the Portland video data, as well as other types of data. The model would also ideally undergo more training phases in order to reach a higher level of accuracy.

Replication of this process is also suggested, in order to verify that this can be done by other individuals and agencies interested in extracting counts from video data. The code used in this project is available for further research use on github at https://github.com/a-hopps/2024-PSU-REU. This code can be used to convert Eurocity data into the YOLO data format, and along with any dataset in the yolo format, to train a model. The repository also provides a basic program to count objects from a video file, which could be edited to work with the right systems and packages.

Acknowledgements This work was supported by the National Science Foundation under grant no. 2244551. Additionally, I would like to thank Christof Teuscher and Teuscher Labs, as well as Portland State University for the training and resources associated with the program that gave me this opportunity.

References

1. Markus Braun, Sebastian Krebs, Fabian B. Flohr, and Dariu M. Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.
2. Mara Chagas Diogenes, Ryan Greene-Roesel, Lindsay S. Arnold, and David R. Ragland. Pedestrian counting methods at intersections: A comparative study. 2002(1):26–30.
3. Alejandro Forero and Francisco Calderon. Vehicle and pedestrian video-tracking with classification based on deep convolutional neural networks. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pages 1–5. ISSN: 2329-6259.
4. Shizuka Fujisawa. Pedestrian counting in video sequences based on optical flow clustering.
5. Sanjukta Ghosh, Peter Amon, Andreas Hutter, and André Kaup. Reliable pedestrian detection using a deep neural network trained on pedestrian counts. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 685–689. ISSN: 2381-8549.
6. Jocher Glenn, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO. https://github.com/ultralytics/ultralytics.
7. Ryan Greene-Roesel, Mara Chagas Diogenes, David R. Ragland, and Luis Antonio Lindau. Effectiveness of a commercially available automated pedestrian counting device in urban environments: Comparison with manual counts.
8. Sirisha Kothuri, Krista Nordback, Andrew Schrope, Taylor Phillips, and Miguel Figliozzi. Bicycle and pedestrian counts at signalized intersections using existing infrastructure: Opportunities and challenges. 2644(1):11–18.
9. Sirisha Kothuri, Patrick Singleton, Mahir Vahedi Saheli, Elizabeth Yates, and Joseph Broach. Active transportation counts from existing on-street signal and detection infrastructure. Technical report, Portland State University and Utah State University, 2024.
10. Jingwen Li, Wei Wu, Dan Zhang, Dayong Fan, Jianwu Jiang, Yanling Lu, Ertao Gao, and Tao Yue. Multi-pedestrian tracking based on KC-YOLO detection and identity validity discrimination module. 13(22):12228. Number: 22 Publisher: Multidisciplinary Digital Publishing Institute.
11. Luis F. Miranda-Moreno and David Fernandes. Modeling of pedestrian activity at signalized intersections: Land use, urban form, weather, and spatiotemporal patterns. 2264(1):74–82.
12. Srinivas S. Pulugurtha and Sudha R. Repaka. Assessment of models to measure pedestrian activity at signalized intersections. 2073(1):39–48. Publisher: SAGE Publications Inc.
13. Robert J. Schneider, Lindsay S. Arnold, and David R. Ragland. Methodology for counting pedestrians at intersections: Use of automated counters to extrapolate weekly volumes from short manual counts. 2140(1):1–12.
14. Mohammad Shokrolah Shirazi and Brendan Morris. Vision-based pedestrian monitoring at intersections including behavior & crossing count. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1022–1027.
15. Ultralytics. Yolo performance metrics, Jul 2024. https://docs.ultralytics.com/guides/yolo-performance-metrics/class-wise-metrics.
16. Yuhan Wang and Han Yang. Multi-target pedestrian tracking based on YOLOv5 and DeepSORT. In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pages 508–514.

Prediction of Suicide Rates Across the US Using Healthcare Access Data

Aparna Kamath

Abstract The goal of this project was to explore the relationship between suicide rates in each state and factors relating to medical care access in those states. The factors analyzed were number of primary care providers, number of mental health providers, number of uninsured individuals, number of individuals with dedicated health providers and those who are avoiding care due to the cost. These numbers were collected through an official healthcare data website, America's Health Rankings, and were calculated per 100,000 population. Trends were then exposed between the factors and suicide incidence to create a Long Short Term Memory model that uses supervised learning to predict suicide rates in the US in specific years. It was found that these factors served as strong indicators, helping to produce an accurate predictive model with an RMSE of 4.459.

1 Introduction

Suicide is one of the most harmful public health crises of the current age. Affecting adults and adolescents alike, destroying families and communities and changing the lives of many every day. For years, researchers and scientists have been working to demystify this issue, and find the causes behind one of the most destructive silent killers. While no singular cure or cause has been pinpointed, it has been found that various mental health issues, such as depression and post traumatic stress disorder (PTSD), as well as substance use disorders are factors both largely connected to why people are drawn to committing suicide. The only way, at the present moment, medical professionals and communities can help those who are suffering is through preventative care.

Aparna Kamath
Portland State University, 1930 SW 4th Ave # 500, Portland, OR 97201
email: apkamathx@gmail.com

One of the biggest forms of care that can contribute to the well-being of those struggling with these various disorders is access to help. This help can come in the form of a physician or psychiatrist who can offer diagnoses to help treat the causes behind this crisis. However, not all people are connected to this level of care with an estimated total of around 83 million people living in areas with no available access to a primary care physician (”AMA president sounds,” 2023). These problems have only been exacerbated by COVID-19, resulting in a dearth of medical professionals including nurses and physicians. Furthermore this problem does not appear to be reaching a solution in the near future with the AAMC announcing that the US will face a shortage of over 80 thousand physicians by 2036. In addition, medicines and therapy, required in the care of disorders such as depression have costs that can range from hundreds to multiple thousands of dollars without insurance. This is an unbearable cost for several families, preventing them from trying to obtain the care they need.

Lack of physicians as well as the costs associated with mental health care are important factors in evaluating the overall public health access patients have. In an effort to discover the true impact of factors such as these on suicide, correlations between these factors and suicide rates were studied. If a relationship between suicide incidence and these factors can be found and used to predict suicide rates in various states across the US, it would show that these factors are strongly connected with suicide. By raising awareness about the specific effects of these shortages on public health, solutions can be found and efforts towards finding causation from these correlations can occur.

2 Related Work and Background

Works exploring ideas relating to the prediction of suicide rates using health care data have been produced. Research focusing on the prediction of suicide in adolescents used data from electronic health records (EHR). Factors focusing on personal identity of patients as well as clinical factors such as laboratory tests and specific medicines were used as the suicide “predictors.” It was found that these factors did serve as accurate indicators for patients who were at risk of committing suicide (Su et al., 2020). Additionally, research has also been done into suicide screening using EHR where certain suicide risks and indicators are screened for in health records of adolescents. The indicators chosen were factors such as demographic, medications taken as well as socioeconomic factors. These indicators were proven to be accurate in determining patients where a suicide attempt was imminent (Walsh et al., 2018). In a similar study, researchers in Korea used a health survey given to a number of patients asking questions about their demographic, education, mental health issues and stress, among other factors, to both predict likelihood of suicide and find the best indicators for prediction. This method proved to work, predicting suicide indicators with an accuracy of 0.821 (Ryu et al., 2018). However, data used in these works did not range to a wider pool of patients that included adults, who make up a huge percent of deaths by suicide. In addition, EHR show a very specific group of factors that can be used to predict and focuses on the individual. The method used in this paper uses factors concerned with a more holistic, public health approach which can allow for broader conclusions regarding correlation and causation. Data regarding specific factors that can only be found in health records can be difficult to access and gather quickly, especially in time to establish screening procedures. Additionally, the factors used to indicate often only work towards very specific results that do not have easily actionable solutions. Work has also been done to predict first-time suicide attempts using EHR data. This data was searched for any mentions of past depression diagnoses, clinical contact and other factors. It was found that this model was accurate in predicting if a patient would attempt to commit suicide (Tsui et al., 2021). While this research does work on suicide prediction, it focuses on whether a patient would commit suicide whereas the research done here focuses on what factors can influence suicide rates. A related piece of hypothesizing research was done where researchers explored the possibility of suicide prediction as a classification problem. A list of indicators would classify if a patient was either at risk for suicide or not (Ribeiro et al., 2016). Most research regarding suicide prediction views the problem using classification. This paper takes a different approach of using a long-short term memory supervised learning model, ideal for time series prediction.

3 Data and Methodologies

Data was collected from two main websites, America's Health Rankings and the Centers for Disease Control and Prevention. The data collected from both websites are organized per 100,000 of the total population. America's Health Rankings is the oldest and longest-running assessment platform of the nation's health done on a state-by-state basis. It contains data on physician and mental health provider quantity by state as well as factors on health care cost accessibility such as number of individuals uninsured or avoiding care due to the cost. Centers for Disease Control and Prevention collects less specific health care data but contains the most accurate and up-to-date suicide rates by state.

3.1 Data Collection

The following are all excerpts from a larger dataset containing all data used. Full data ranges from the years 2017 to 2021.

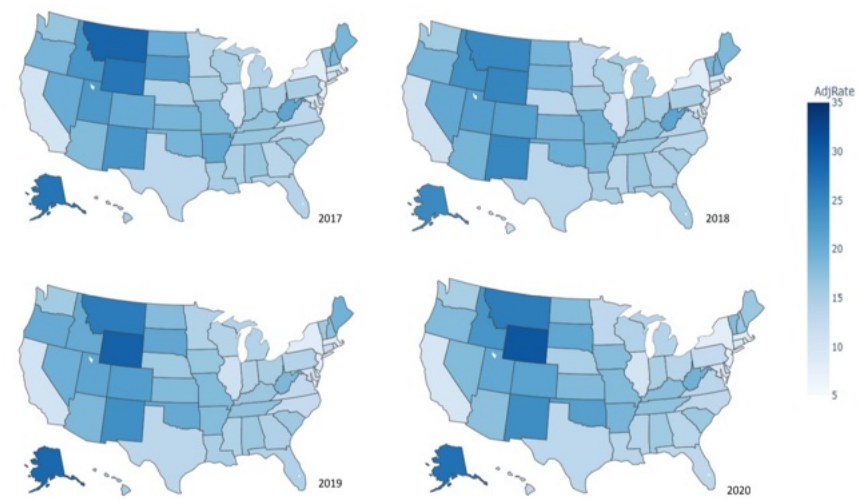


Fig. 1 Visualization of Suicide Rates in the US from 2017 to 2020

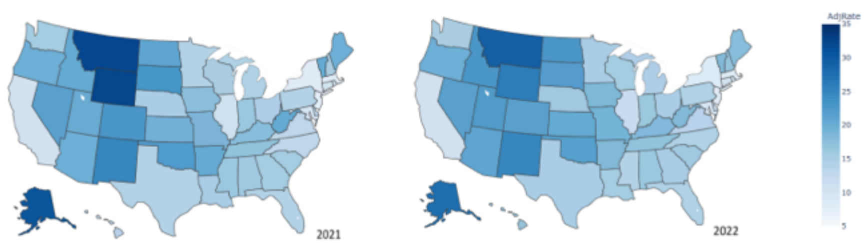


Fig. 2 Visualization of Suicide Rates in the US in 2021 and 2022

Year	State	Primary Care Providers
2021	Alabama	213.5
2021	Alaska	332.2
2021	Arizona	221.2
2021	Arkansas	216.1
2021	California	197.8
2021	Colorado	266.2
2021	Connecticut	287.5
2021	Delaware	296.2
2021	Florida	266.9
2021	Georgia	231.8
2021	Hawaii	242
2021	Idaho	220.2
2021	Illinois	252.3
2021	Indiana	252.3
2021	Iowa	255.3
2021	Kansas	265.2
2021	Kentucky	281.6
2021	Louisiana	226.8
2021	Maine	226.8

Fig. 3 Number of Primary Care Providers By State (per 100,000 population)
Includes active primary care providers (including general practice, family practice, obstetrics and gynecology, pediatrics, geriatrics, internal medicine, physician assistants and nurse practitioners).

Year	State	Mental Health Providers
2021	Alabama	120.8
2021	Alaska	625.9
2021	Arizona	154.8
2021	Arkansas	254.3
2021	California	409.7
2021	Colorado	407.2
2021	Connecticut	449.3
2021	Delaware	299
2021	Florida	185.1
2021	Georgia	159.8
2021	Hawaii	267
2021	Idaho	234.8
2021	Illinois	275
2021	Indiana	183
2021	Iowa	181.3
2021	Kansas	219.9
2021	Kentucky	284.3
2021	Louisiana	283.6
2021	Maine	523.3

Fig. 4 Number of Mental Health Providers By State (per 100,000 population)
Number of psychiatrists, psychologists, licensed clinical social workers, counselors, marriage, family therapists and advanced practice nurses specializing in mental healthcare.

Year	State	Uninsured
2021	Alabama	9.9
2021	Alaska	11.4
2021	Arizona	11.4
2021	Arkansas	9.2
2021	California	7
2021	Colorado	8
2021	Connecticut	5.2
2021	Delaware	5.7
2021	Florida	12.1
2021	Georgia	12.6
2021	Hawaii	3.9
2021	Idaho	8.8
2021	Illinois	7
2021	Indiana	7.5
2021	Iowa	4.8
2021	Kansas	9.2
2021	Kentucky	5.7
2021	Louisiana	7.6
2021	Maine	5.7

Fig. 5 Number of Uninsured Individuals By State (per 100,000 population)
Population not covered by private or public health insurance

Year	State	Avoiding Care
2021	Alabama	11.1
2021	Alaska	10.8
2021	Arizona	10.5
2021	Arkansas	11.7
2021	California	9.2
2021	Colorado	9.5
2021	Connecticut	7.4
2021	Delaware	7.4
2021	Florida	14.5
2021	Georgia	15.5
2021	Hawaii	5.3
2021	Idaho	9.2
2021	Illinois	9.8
2021	Indiana	8.6
2021	Iowa	6
2021	Kansas	9.9
2021	Kentucky	8.8
2021	Louisiana	11.6
2021	Maine	7.6

Fig. 6 Number of Individuals Avoiding Care Due to the Cost (per 100,000 population)
People who reported a time in the past year when they needed to see a doctor but were unable to due to cost.

Year	State	Dedicated Health Providers
2021	Alabama	78.5
2021	Alaska	76.2
2021	Arizona	77.6
2021	Arkansas	84.4
2021	California	82.1
2021	Colorado	81.8
2021	Connecticut	87.4
2021	Delaware	84.9
2021	Florida	74.2
2021	Georgia	81.2
2021	Hawaii	90.1
2021	Idaho	82.7
2021	Illinois	83.6
2021	Indiana	84.6
2021	Iowa	85.5
2021	Kansas	85.9
2021	Kentucky	86.3
2021	Louisiana	85.6
2021	Maine	90.7

Fig. 7 Number of Dedicated Health Providers By State (per 100,000 population)
People who reported having a personal doctor or health care provider.

3.2 Methods

The method used in this paper featured two main parts: data analysis and the machine learning model. To find what strategy was best to use to create the prediction model, basic scatterplots to visually depict the data as well as heat maps between each of the factors chosen: primary care providers, mental health providers, uninsured, avoiding care and dedicated health providers were created. Then, using the trends found in these depictions, two machine learning models were created, a linear regression model and an LSTM. The language used for linear regression was Python in the PyCharm IDE and the statsmodel package. For the LSTM, tensorflow was used to create the model.

3.2.1 Data Preprocessing

When data was collected and collated, it was found that there were some ”gaps” in the dataset. These gaps were found in the data for uninsured individuals and those avoiding care due to cost, for a few states. To fill these gaps, a k-nearest neighbors (KNN) algorithm was used to make predictions as to what these values are. A KNN imputer works by identifying a certain number (k) of values in the dataset that are similar/ related to the unknown value. These k values are found using the numbers that are the shortest euclidean distance away from the desired input. In a situation where there are missing values, the euclidean distance is calculated by ignoring the missing values and increasing the weight of the non-missing coordinates. The formulas to find the euclidean distance and weights allotted to each coordinate are shown below.

$$d_{xy} = \sqrt{weight\ *\ squared\ distance\ from\ present\ coordinates}$$

$$weight = \frac{Total\ number\ of\ coordinates}{Number\ of\ present\ coordinates}$$

Fig. 8 Euclidean Distance and Weight Formulas for KNN

The KNNImputer scikit-learn class was used with 3 nearest neighbors, the missing values marked as NaN and all neighbors weighted equally. For example, if the state of Oregon lacked data, California, Idaho and Washington data would be used as the 3 neighbors, or values. This algorithm was used to fill any gaps in the dataset and ensure the data was complete before analysis began.

3.2.2 Data Analysis

The first step after collating the data was to create simple facet scatterplots using plotly, an open-source graphic library for Python. These scatterplots helped to visualize the large group of data used and choose what model would be best. The two models being evaluated for use was a linear regression model and a long-short term memory model (LSTM). A linear regression model predicts a dependent variable based on multiple indepedent variables. The first condition to using this type of model is that the variables being used in the analysis have a direct linear relationship. These scatterplots helped to visualize the data and see if a relationship between the independent and dependent variables is present.

In the scatterplots, the states are grouped into regions as defined by the CDC, shown as the different colors on the plot. The regions are organized as follows: New England includes Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont; Middle Atlantic includes New Jersey, New York, and Pennsylvania; East North Central includes Illinois, Indiana, Michigan, Ohio, and Wisconsin; West North Central includes Iowa, Kansas, Minnesota, Missouri, Nebraska, North Dakota, and South Dakota; South Atlantic includes Delaware, District of Columbia, Florida, Georgia, Maryland, North Carolina, South Carolina, Virginia, and West Virginia; East South Central includes Alabama, Kentucky, Mississippi, and Tennessee; West South Central includes Arkansas, Louisiana, Oklahoma, and Texas; Mountain includes Arizona, Colorado, Idaho, Montana, Nevada, New Mexico, Utah, and Wyoming; Pacific includes Alaska, California, Hawaii, Oregon, and Washington. The size of the bubbles is defined by the adjusted suicide rate, and how large it is.

Fig. 9 Scatterplot of Primary Care Physicians vs. Suicide Adjusted Rate

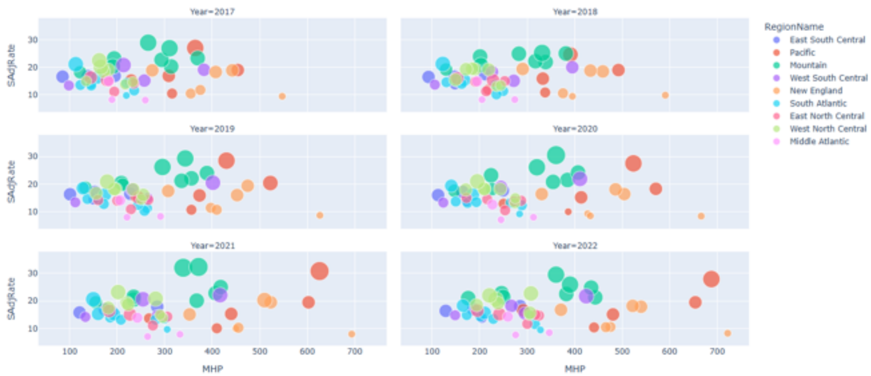


Fig. 10 Scatterplot of Mental Health Providers vs. Suicide Rate

Fig. 11 Scatterplot of Uninsured Individuals vs. Suicide Rate

Fig. 12 Scatterplot of Number of Individuals Avoiding Care Due to the Cost vs. Suicide Rate

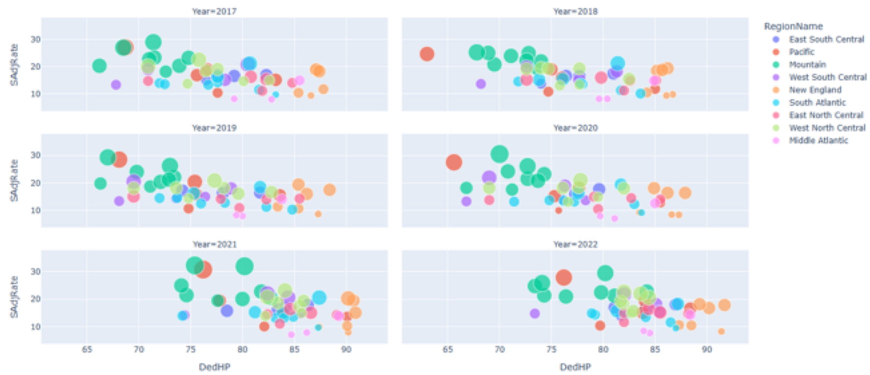


Fig. 13 Scatterplot of the Number of Individuals with Dedicated Health Providers vs. Suicide Rate

These scatterplots (Fig.10 - Fig.14) showed a relative linear relationship between the indicators being tested and the suicide rates over these five years. To explore this relationship, further analysis and statistical models needed to be created. The next step in determining whether a linear regression model could be used is to see if the data follows a normal distribution.

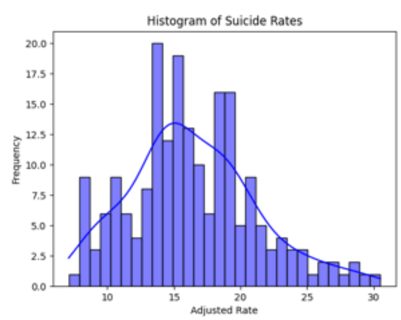


Fig. 14 Distribution of Suicide Rates From 2017 to 2022

As can be seen from the histogram, the distribution of the suicide adjusted rates is normal. Then, to take a deeper look at the correlations and ensure the directions and strength were as expected, a correlation matrix was created.

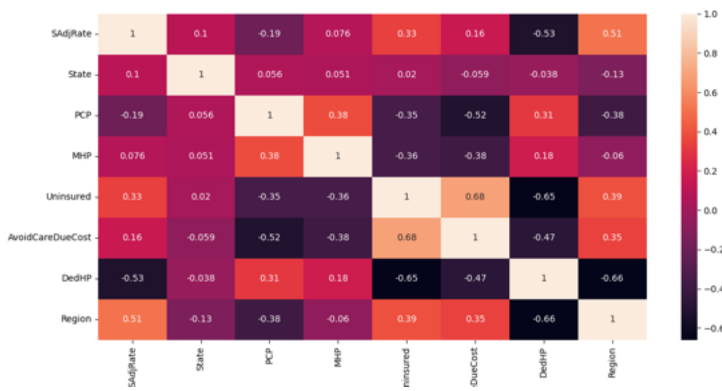


Fig. 15 Heat Map of Indicators

Since the requirements for use of a linear regression model were met, this model was created.

3.2.3 Linear Regression Model

Initial data preparation required converting the state names using integer encoding. The data was then separated into training and testing sets, the training set consisting of 2017 and 2018 data and the testing set consisting of 2019 and 2020 data. The program was then run and the statsmodel package of Python was used to evaluate the model parameters. The model was then used to make predictions for 2022.

3.2.4 Long Short Term Memory Model

A long short term memory prediction model works best for problems that require multiple input variables. It is extremely useful in time series forecasting when predicting for future events using past data and trends. First, the dataset needed to be prepared for the LSTM. The first step was to normalize the features and integer encode the state labels. The second step was to frame the data as a supervised learning problem, where the suicide rate in the next year is predicted using the conditions at the previous time step. The conditions used are the number of physicians, number of mental health providers, number uninsured, number avoiding care and number of people with a dedicated health provider. Then, to fit the LSTM to the data, the dataset was split into a train and a test set. The train set was made up of 2017 and 2018 and the test set was made up of 2019 and 2020 data. Additionally, the data was reshaped to fit the 3D shape expected by the LSTM model. The LSTM had 50 neurons in the first hidden layer and 1 neuron in the output layer. The input shape was 1 time step with 8 features. The eight features are the conditions as well as the region and the previous year's adjusted suicide rate. It was fit for 50 epochs with a batch size of 72. After fitting the model, it was evaluated on the full dataset.

4 Results

4.0.1 Linear Regression

The statsmodel package provides a statistical analysis of the linear regression model's suitability for prediction. Below is this analysis.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.390
Model:	OLS	Adj. R-squared:	0.343
Method:	Least Squares	F-statistic:	8.390
Date:	Sat, 27 Jul 2024	Prob (F-statistic):	6.70e-08
Time:	21:55:03	Log-Likelihood:	-269.56
No. Observations:	100	AIC:	555.1
Df Residuals:	92	BIC:	576.0
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	37.2503	10.650	3.498	0.001	16.099	58.402
x1	0.0361	0.027	1.349	0.181	-0.017	0.089
x2	-0.0128	0.009	-1.467	0.146	-0.030	0.005
x3	0.0075	0.004	1.847	0.068	-0.001	0.015
x4	0.2934	0.194	1.513	0.134	-0.092	0.679
x5	-0.4515	0.206	-2.193	0.031	-0.860	-0.043
x6	-0.2602	0.110	-2.356	0.021	-0.480	-0.041
x7	0.4783	0.218	2.197	0.031	0.046	0.911

Fig. 16 OLS Regression Results

An R-squared score of 1 indicates the model perfectly fits the data. The R-squared score of this example is 0.39, which is extremely low. The p-values, which indicate whether there is a relationship between predictors and the output variable, are high. They are greater than *alpha* = 0.05 which is the benchmark p-value for significance.

4.0.2 LSTM

The train and test loss from the LSTM during training is shown below.

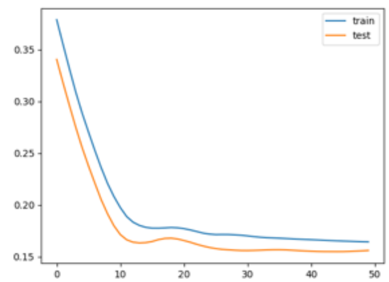


Fig. 17 Train and Test Loss During Training

The model converged. Then, predictions were made using the model. The results are shown below.

State	Suicide Rate	Predicted Rate	Region
Florida	14.4	17.6	5
Georgia	14.9	18.5	5
Hawaii	16.6	15.7	9
Idaho	22.8	17.2	8
Illinois	11.7	14.9	3
Indiana	16.5	14.7	3
Iowa	18.5	13.7	4
Kansas	20.6	16.3	4
Kentucky	18.1	16.1	6
Louisiana	15.5	17.6	7
Maine	17.9	14.5	1

Fig. 18 Results From LSTM with Real Suicide Rate and Predicted Rate - 1

State	Suicide Rate	Predicted Rate	Region
North Carolina	14.5	17.5	5
North Dakota	22.7	15.4	4
Ohio	15.0	14.9	3
Oklahoma	21.7	20.8	7
Oregon	19.5	17.9	9
Pennsylvania	14.3	16.4	2
Rhode Island	10.5	14.4	1
South Carolina	15.7	17.7	5
South Dakota	22.0	16.3	4
Tennessee	17.0	18.3	6

Fig. 19 Results From LSTM with Real Suicide Rate and Predicted Rate - 2

An RMSE of 4.459 was achieved. A graph comparing the actual with the predicted suicide rates is below.

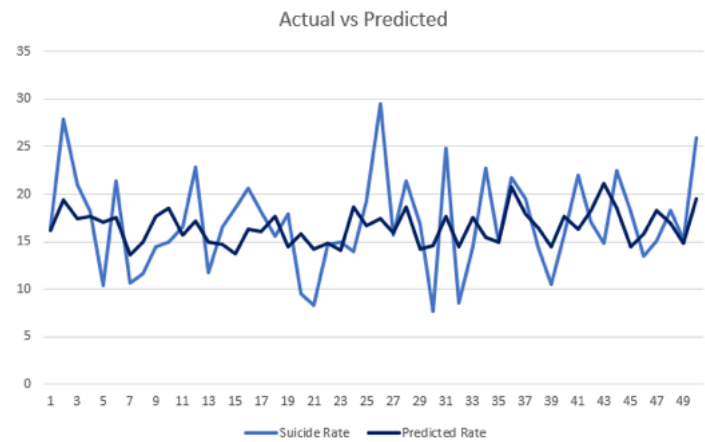


Fig. 20 Graph of Actual Suicide Rates vs. Predicted

5 Conclusion

5.0.1 Linear Regression

From the information shown in Fig. 17, it was found that the linear regression model was a poor fit to the data. The R-squared score of the model was 0.39, much lower than the perfect-fit benchmark of 1. In addition, four of the seven p-values of the indicators were higher than the significance level *alpha* = 0.05. This indicates that there lacks sufficient evidence to show a linear relationship between suicide rates and the indicators chosen. So, through this analysis, it was concluded that a linear regression model could not be used on this data. As a result, no predictions were done using this model.

5.0.2 LSTM

From the information shown in Fig. 19 and Fig. 20, it can be seen that the LSTM model is able to predict suicide rates for the year 2022 using the indicators provided with relative accuracy. While the model is not able to predict the suicide rate for every state correctly, as shown in Fig. 21, it is able to predict enough states within each region to provide accurate overall results. In addition, this overall analysis shows that the LSTM is the best model to use to not only fit this data but to provide the best prediction results for this research question.

5.1 Applications

The results found from this research can be used in public health research and outreach. By showing that indicators such as number of physicians and mental health providers have an impact on suicide rates, efforts can be started to increase the number of these health professionals in areas where there are dearths in order to reduce suicide incidence. Work can also be done to ensure more individuals in this country not just have access to cheaper healthcare but have a dedicated health professional/doctor attached to them. As the result show, this can play a huge role in the impacts of suicide in an area. They can directly help with any concerns that patients could have regarding mental health and immediately provide references to mental health providers that can provide the specific care they need. In addition, by showing real data and correlations between these factors, research can use these results to move from correlation to causation, working to explore the why of suicide. This research presents a breakthrough in this area of public health research and reinforces the power of preventative care and the need for more health professionals in the nation.

5.2 Concerns and Next Steps

One of the limitations of this project was the size of the dataset being used. The number of mental health providers in each state was not counted or considered in healthcare datasets until 2017. Since this is one of the factors deemed extremely important in this model and research, this automatically shortened the data available to use in the model. In addition, any data relating to public health or serious issues such as suicide is difficult to access, especially without any professional clearance as a high school student. Another limitation is the way healthcare data is often reported. Some sources report data only every five or ten years, others miss multiple years in a row of data collection due to external issues and not all sources report their data under the same ratio. Without this consistent data available, it becomes very difficult to compile data to use for projects such as this one. The next steps could be to add more indicators to the analysis and model and see if the results show a greater improvement. Another step could be narrowing in to more specific areas such as particular counties or states where suicide incidence is high. Changing perspective to see the project from a county level could help to focus on rural areas where shortages of health professionals impact residents more acutely.

Acknowledgements I would like to acknowledge the help of my professor Dr. Christof Teuscher for his mentorship and advice over the course of this project.

References

AMA president sounds alarm on national physician shortage — American Medical Association. (n.d.). Retrieved July 25, 2024, from https://www.ama-assn.org/press-center/press-releases/ama-president-sounds-alarm-national-physician-shortage

America's Health Rankings — AHR. (n.d.). Retrieved July 26, 2024, from https://www.americashealthrankings.org/

Geographic division or region - Health, United States. (2023, June 26). from https://www.cdc.gov/nchs/hus/sources-definitions/geographic-region.htm

Multivariate Time Series Forecasting with LSTMs in Keras - MachineLearningMastery.com. (n.d.). Retrieved July 25, 2024, from https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

New AAMC Report Shows Continuing Projected Physician Shortage. (n.d.). AAMC. Retrieved July 25, 2024, from https://www.aamc.org/news/press-releases/new-aamc-report-shows-continuing-projected-physician-shortage

Predicting suicide attempts in adolescents with longitudinal clinical data and machine learning - Walsh - 2018 - Journal of Child Psychology and Psychiatry - Wiley Online Library. (n.d.). Retrieved July 25, 2024, from https://acamh.onlinelibrary.wiley.com/doi/abs/10.1111/jcpp.12916

Ribeiro, J. D., Franklin, J. C., Fox, K. R., Bentley, K. H., Kleiman, E. M., Chang, B. P.,& Nock, M. K. (2016). Letter to the Editor: Suicide as a complex classification problem: machine learning and related techniques can advance suicide prediction - a reply to Roaldset (2016). Psychological Medicine, 46(9), 2009–2010. https://doi.org/10.1017/S0033291716000611

Ryu, S., Lee, H., Lee, D.-K., & Park, K. (2018). Use of a Machine Learning Algorithm to Predict Individuals with Suicide Ideation in the General Population. Psychiatry Investigation, 15(11), 1030–1036. https://doi.org/10.30773/pi.2018.08.27

Stats of the State - Suicide Mortality. (2023, February 15). https://www.cdc.gov/nchs/pressroom/sosmap/suicide-mortality/suicide.htm

Su, C., Aseltine, R., Doshi, R., Chen, K., Rogers, S. C., & Wang, F. (2020). Machine learning for suicide risk prediction in children and adolescents with electronic health records. Translational Psychiatry, 10(1), 1–10. https://doi.org/10.1038/s41398-020-01100-0

Suicide. (n.d.). Retrieved July 25, 2024, from https://www.who.int/news-room/fact-sheets/detail/suicide

Tsui, F. R., Shi, L., Ruiz, V., Ryan, N. D., Biernesser, C., Iyengar, S., Walsh, C. G., & Brent, D. A. (2021). Natural language processing and machine learning of electronic health records for prediction of first-time suicide attempts. JAMIA Open, 4(1), ooab011. https://doi.org/10.1093/jamiaopen/ooab011

Can Evolution Harness Chaos Inside Analog Circuits for Advantageous Behavior?

Artem Arefev

(Claude Monet, The Cliff, Étretat, Sunset 1883)

Artem Arefev
Portland State University,
1825 SW Broadway Portland, OR 97201
e-mail: artem.arefev32@gmail.com

Abstract

It is known that many biological systems utilize chaos for their advantage, and chaotic dynamics have successfully been integrated with robot controller circuits opening up to them a new repertoire of useful behavior. However, these implementations are limited, as they only realize uses for chaos that are obvious to the human designer and fail to consider the plethora of other unconventional approaches. This work aims to investigate the possibility of using genetic programming to evolve analog controller circuits that can harness chaotic dynamics. A genetic programming algorithm that can synthesize analog electric circuits to achieve some explicitly defined desired behavior is implemented in Python. An evaluation of its performance suggests that an improved version of the algorithm could indeed generate unconventional analog controller circuits that harness chaos, although more work is needed to confirm this. Similarly, two distinct chaotic analog circuits are simulated in SPICE with an attempt to recreate the chaotic dynamics present in their physical realizations. Results show that only certain circuits in which chaos happens on a large enough scale can be accurately simulated. This work is a stepping stone for further research using an improved circuit-building genetic programming algorithm and advanced evolution-assisting techniques to generate controller circuits that harness chaos for useful and possibly emergent behavior.

1. Introduction

A growing body of evidence indicates that numerous biological control systems, optimized through millions of years of evolutionary processes rely on chaotic dynamics for useful behavior. This observation suggests that incorporating chaos-harnessing capabilities into the design of man-made control systems can enhance their performance in specific contexts.

Several examples highlight this concept. Ants searching for a source of food in an unknown environment (thereby performing ”area search”) traverse the area chaotically until the food source is located [1]. This behavior has been implemented into navigation robots for effective area search, that use circuits or systems of equations behaving chaotically, as their source of chaotic dynamics [2]. Chaotic dynamics can be used to efficiently traverse any abstract search space when little information about the space is known: such as the space of all possible hexapod movement patterns using a chaotic neural circuit, in search of those that allow the hexapod robot to get its leg unstuck from a hole [3].

This work investigates the possibility of using genetic programming to evolve analog controller circuits that can harness chaos for useful behavior. In this work, two types of chaotic circuits are simulated using the SPICE circuit simulator, and their feasibility as sources of chaotic dynamics is analyzed. Additionally, a practical implementation of genetic programming to build electric circuits using Python is discussed in detail. In essence, this work lays the foundation for the automatic synthesis of chaos-harnessing controller circuits using genetic programming.

1.0.1 Genetic Programming Description

Genetic programming aims to automatically develop programs from a high-level statement of desired behavior, by evolving syntax trees composed of primitives and terminals. Syntax trees are effective at representing hierarchical instructions. For example, a syntax tree can be used to represent a mathematical expression as shown in Fig. 1.

Syntax trees are composed of interconnected nodes with a common root node, and each node represents a specific instruction or function. A node that has children and is not the last node in a branch, is called a ”primitive”. Meanwhile, a node that does not have children of its own and so is the final node of its branch, is called the ”terminal”. The primitive set defines any single instruction or function that a primitive node in a syntax tree can represent, while the terminal set defines any single instruction or function that a terminal node can represent and often is used to provide input into the syntax tree [4].

By assuming a specific set of instructions defined in the syntax tree to be an individual, one can create a population of individuals and involve them using evolutionary operators. An evolutionary operator is any operation that can alter an individual’s syntax tree or create new individuals from existing ones, thereby enabling reproduction [4].

The evolutionary process underlying genetic programming is guided by the fitness function, which defines the success of each syntax tree participating in the evolution. It does so by encoding the high-level statement of the behavior desired from the syntax trees where trees that display more of the desired behavior are assigned a higher fitness value. Trees with higher fitness values are then allowed to participate in reproduction more frequently, which propagates the more successful tree structures into future generations. This way, one can optimize any computer program to behave in a certain way, defined by the fitness function [4].

2. Data and Methodologies

In this section, this work’s methodology is discussed in detail. This work implements genetic programming uses it to generate analog circuits. The SPICE circuit simulator is used to evaluate the generated circuits and to reproduce the chaotic dynamics of two distinct analog circuits.

2.1 Data Collection

This work relies on SPICE-based circuit simulation to evaluate the analog circuit designs generated by the genetic programming algorithm during evolution, and to reproduce chaotic dynamics of the ”single-transistor circuit” [6] and the ”piecewise

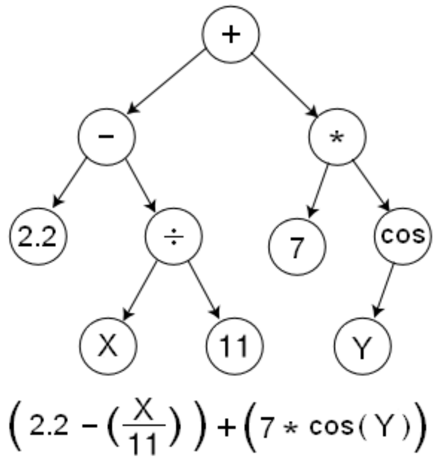


Fig. 1: Example of GP syntax tree representing a mathematical expression. Source: BAxelrod at English Wikipedia, Public domain, via Wikimedia Commons [5].

capacitor circuit" [7]. A bifurcation diagram of the piecewise capacitor circuit with descent into chaos is produced, several chaotic attractors of the single-transistor circuit are recreated, and time series data is analyzed using autocorrelation analysis and power spectrum analysis.

2.1.1 SPICE Circuit Simulations

In this work, circuits are simulated using Ngspice-43 at the default temperature of 27° Celsius. For the transient analysis of the single-transistor circuit, normal timestep values can be used as the nature of its chaotic behavior is captured well on normal scales (see 3.2). In contrast, the timestep value for the transient analysis of the piecewise capacitor circuit should be as small as computational resources permit (see 3.3).

For both circuits default SPICE components are used: resistors, capacitors, and inductors. Additionally, the single-transistor circuit utilizes the "PRF949" NPN transistor model, with parameters presented in Table 1.

Parameter:	Value:
IS	4.66×10^{-16}
BF	199.4
NF	1
VAF	53.06
IKF	0.96749
ISE	114.96×10^{-15}
NE	2
BR	27.68
NR	1
VAR	1.976
IKR	0.009943
ISC	1.42×10^{-18}
NC	1
RB	12.14
IRB	0
RBM	4.957
RE	0.5974
RC	1.988
CJE	5.681×10^{-13}
VJE	0.6
MJE	0.4122
CJC	2.058×10^{-13}
VJC	0.2962
MJC	0.118
CJS	0
VJS	0.7
MJS	0
XCJC	1
TR	0
TF	1.9×10^{-12}
XTF	30.9
VTF	3.148
ITF	0.1418
PTF	0
FC	0.9431
EG	1.11
XTI	3
AF	1
KF	4.00×10^{-16}

Table 1: Transistor Model Parameters

2.1.2 Chaotic Data Analysis

Using the output from SPICE simulations, this work reproduced a chaotic attractor, a bifurcation diagram, autocorrelation plots, and a power spectrum.

During the simulation of the piecewise circuit, the charge across its capacitor is recorded after the transient phase. This is repeated while the amplitude of the input

AC voltage being fed into the circuit is incrementally increased from 0 V to 2 V. Using this data, a bifurcation diagram of the circuit's descent into chaos is produced.

During the simulation of the single-transistor circuit, the voltage across its capacitor is recorded with respect to time, after the short transient phase. Using this data, a strange attractor is reconstructed. Additionally, both the autocorrelation function and the power spectrum are calculated for this data.

Finally, to sidestep the computational intensity associated with the direct calculation of the autocorrelation function, which has a time complexity of $O(n^2)$, the Wiener-Knichin method is employed, which utilizes NumPy's Discrete Fourier Transform to calculate the autocorrelation function.

2.2 Methods

To realize the evolution of circuit-building programs, the DEAP evolutionary computation framework [8] is used for its implementation of genetic programming. This implementation is customized to fit the specifics of circuit-building. Additionally, both the single-transistor circuit and the piecewise capacitor circuit are simulated with SPICE, to reproduce their chaotic dynamics.

2.2.1 Genetic Programming Implementation Description

Arithmetic Performing Function Subtrees

The original implementation of the circuit-building genetic programming algorithm [9] sets component values using floating-point values generated by "arithmetic performing functions". An arithmetic performing function (abbreviated as, APF) is a syntax tree that is separate from the main circuit-building syntax tree, but which is evolved simultaneously along with the circuit-building tree and can be called by the main tree in the same way that a normal terminal node would be called.

In this work, 5 APF subtrees are implemented in DEAP as "automatically defined functions" and are elements in the main circuit-building terminal set. These APF subtrees are composed from the mathematical operations of the "arithmetic-performing" primitive set. The goal of any APF subtree is to find a mathematical expression that will produce a component value in the correct ballpark, from randomly generated floating-point constants.

The return value "n" of any APF subtree is used to set a component "x" in the following manner:

- If -100 < n < -5, then $x = \frac{-5}{95} * (n + 100)$.
- If -5 ≤ n ≤ 5, then $x = n$.
- If 5 < n < 100, then $x = \frac{-5}{95} * (n - 100)$
- If -100 > n or 100 < n, then $x = 0$.

Primitive Sets

In this work, the primitive sets are mostly the same as in the original study [9] except they are shortened to simplify the creation of the genetic programming algorithm. The primitive sets consist of

$$\mathscr{F}_{cbs} = \left\{ \begin{matrix} R, \;\; C, \;\; D, \;\; FLIP, \;\; SERIES, \\ PARALLEL, \;\; NOOP, \;\; T\_GND\_0, \\ T\_GND\_1, \;\; T\_POS\_0, \\ T\_POS\_1, \;\; T\_NEG\_0, \;\; T\_NEG\_1 \end{matrix} \right\}$$

$$\mathscr{F}_{aps} = \left\{ add(+), \;\; subtract(-) \right\}$$

where the function primitive ”R” changes the selected component to a resistor, ”C” changes to a capacitor, ”D” changes to a diode, ”FLIP” swaps the component terminals, ”SERIES” adds a copy of the component in series, ”PARALLEL” adds a copy of the component in parallel, ”NOOP” does nothing, ”T_GND_0” and ”T_GND_1” establish a connection from the component to ground, ”T_POS_0” and ”T_POS_1” establish a connection from the component to a positive voltage source of 5 V, while ”T_NEG_0” and ”T_NEG_1” establish a connection from the component to a negative voltage source. For a more detailed explanation of each of the primitive functions see [9].

Only the function primitives from the circuit-building set (abbreviated as, CBS) are accessible to the syntax tree individuals. These primitives represent netlist-modifying functions that can either add components to the circuit netlist or modify it. And the function primitives from the arithmetic-performing set are only accessible to the APF subtrees. They only implement the addition or subtraction of two floating point values.

The primitives in all of the sets are strongly typed. This means that every primitive is assigned a specific data type for the input it can receive and the output it can produce. This effectively restricts how the primitives and terminals available to a given syntax tree can be combined together.

Terminal Sets

Similar to the primitive sets, the terminal sets used in this work are mostly the same as in the original study [9] except that the APF subtrees are implemented as automatically defined functions and are included in the circuit building terminal set. The terminal sets consist of

$$\mathscr{T}_{cbs} = \begin{Bmatrix} END, & SAFE\_CUT, & APF\_1, \\ APF\_2, & APF\_3, & APF\_4, APF\_5 \end{Bmatrix}$$

$$\mathscr{T}_{aps} = \{R\}$$

where R represents randomly generated floating-point constants from -1.000 to +1.000.

Fitness Function

Before an individual syntax tree can be evaluated using the fitness function, the individual must first be compiled into a circuit-building program, the program must create a netlist representing the circuit, and the circuit must be simulated using SPICE. The fitness function defined in terms of desired-behavior of the circuit can only be applied when the generated circuit's is simulated in SPICE and recorded.

In this work, genetic programming is used to evolve a capacitor in a voltage-doubling circuit, so that the voltage across the load (resistor) is double the 5 V AC voltage being input into the circuit. The capacitor is essential to the the process by which the voltage is doubled, so replacing it with a wire and relying on genetic programming to figure out what component must be substituted for the wire and what value that component should have, tests the algorithm's capabilities at solving simple circuit-building tasks. The genetic programming algorithm is implemented using DEAP and each circuit is simulated using SPICE to calculate the circuit's fitness.

Evolution Parameters

The population used in the genetic programming algorithm consists of 100 randomly generated individuals. The syntax tree of each individual is randomly generated until a depth of 2 to 3 is reached, by adding randomly selected primitive nodes to the tree until the tree is 1 node short of the maximum depth at which point a randomly selected terminal node is added to the tree. The APF subtrees are generated in a similar manner using their respective primitive sets and terminal sets.

For syntax tree replication, the tournament selection method with a tournament size of 3 is used to bias reproduction in favor of more fit individuals. While reproduction is realized using single point crossover with a probability of 50%. Finally, uniform mutation is implemented with a probability of 20%.

2.2.2 Piecewise Capacitor Chaotic Circuit Description

The piecewise capacitor circuit shown in Fig. 2 is a non-autonomous circuit composed of a piecewise capacitor, an inductor ($L_1 = 100\,\mu$H), a resistor ($R_1 = 60\ \Omega$), and an AC voltage source ($0V \le V_s \le 2V$). The capacitance value of the piecewise capacitor changes between $C = 0.1\,\mu$F when its charge polarity is positive and $C = 400$ pF. It was initially conceived as a simplification of the ”Resistor-Inductor-Diode” circuit but displays bifurcations that eventually descend into completely chaotic behavior [7].

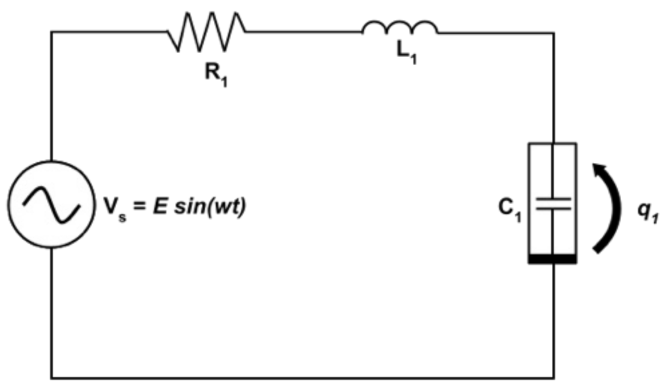


Fig. 2: Single-transistor chaotic circuit schematic.

2.2.3 Single-Transistor Chaotic Circuit Description

The single-transistor circuit shown in Fig. 3 is an autonomous circuit composed of an NPN-type bipolar-junction transistor whose emitter is connected to ground, a first inductor ($L_1 = 15\,\mu$H), a second inductor ($L_2 = 220\,\mu$H), a capacitor ($C_1 = 220$ pF), a resistor ($R_1 = 60$ kΩ), and a DC voltage source ($V_s = 5$ V). This circuit pattern was identified to produce chaotic behavior using an evolutionary algorithm, and its chaotic behavior has already been well studied [6].

3. Results

In this section, the work’s findings are presented. The data collected suggests that GP is the correct approach to generating analog circuits that accomplish their specified goal and that the single-transistor circuit could be an effective source of chaotic dynamics for controller circuits.

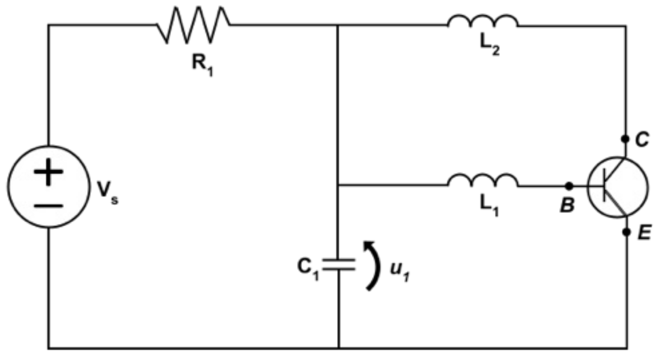


Fig. 3: Piecewise capacitor chaotic circuit schematic.

3.1 Genetic Programming Evolution

A research group has already applied genetic programming to synthesize novel analog circuits [9] and the approach is explained in section 2.2.1. In this work the algorithm is implemented using the DEAP framework in Python and can correctly add a capacitor into a voltage doubling template circuit as well as appropriately set the capacitor's value, coming within 99.4% of the target voltage output value after only 3 generations, as shown in Table 2. In contrast to the initial, generation 0 that is randomly generated and comes within 57.64% of the target value.

A successful solution to the voltage doubling circuit template problem is clearly found by the genetic algorithm in 3 generations, which cannot be done by randomly created syntax trees alone. And over the course of evolution, the population's average performance almost triples: from coming within 37.72% of the desired value to coming within 78.82% of the desired value.

The best solution found by the evolutionary algorithm is shown in Fig. 4, and its physical realization is displayed in Fig. 5. The genetic programming algorithm clearly identified the importance of adding the capacitor, while adding a positive voltage source is trivial and does not affect the functioning of the circuit. A more desirable approach would involve adding a "NOOP" primitive instead of "T_POS_1", which prevent the addition of unnecessary instructions. For this reason, although successful at doubling voltage, the circuit-building program is "bloated" because it contains useless instructions that do not contribute to the circuit's functionality and instead obscure its design. To prevent this, the use of various bloat control techniques should be investigated to ensure that every segment of the generated circuits is necessary.

Gen	Evals	Std	Min	Avg	Max
0	100	1.82474	4.23519	6.22824	13.8658
1	100	1.04131	2.65195	5.32571	10.4556
2	100	1.3953	0.785583	5.29664	13.0186
3	100	1.31911	0.0583747	4.79849	8.78793
4	100	2.18578	0.0255601	4.5049	13.9536
5	100	2.35127	0.0024785	3.72892	14.8413
6	100	2.28661	0.0008523	3.25682	11.1417
7	100	2.36928	0.0022492	3.25813	13.0381
8	99	2.07813	0.0010958	3.52144	11.4546
9	99	2.30279	0.00805088	3.46341	14.7501
10	10	2.30279	0.00805088	3.46341	14.7501
11	100	2.25038	0.000563	3.17735	9.72022
12	100	2.33675	0.0242218	3.21502	14.2865
13	98	2.30716	0.0058902	3.13816	9.0407
14	100	2.56981	0.00040031	3.24369	12.3437
15	100	2.61234	0.0060439	3.26594	10.3846
16	99	2.40622	0.0040528	3.823	11.4645
17	100	2.75711	0.000687	3.5822	11.6605
18	100	2.5289	0.0003567	3.35647	14.4768
19	100	2.22067	0.0014373	3.19717	8.32871
20	99	2.39619	0.00018008	3.21671	11.2613
21	100	2.47348	3.088e-05	2.77119	14.9577
22	100	2.46674	0.0042248	3.12078	9.86709
23	98	2.52678	0.0005183	2.99478	9.98656
24	99	2.43794	0.0001039	3.29271	9.98981
25	100	2.51464	0.0028744	3.38589	10.9608
26	100	3.1022	0.0022315	3.59531	14.983
27	100	2.40305	0.0004463	2.93931	10
28	98	2.94498	0.00014546	3.07678	14.9916
29	100	2.46974	0.0002509	2.63872	11.2967
30	100	2.28972	0.0028796	3.10848	9.99857
31	98	2.17992	0.026511	3.0457	10.1747
32	99	1.98396	0.0141262	3.29716	9.54476
33	100	2.29984	0.00191372	3.1142	10
34	98	2.16047	0.0014102	2.72856	6.66685
35	100	2.24774	0.00034629	2.59354	8.91668
36	99	2.08907	0.00038713	2.82866	6.66672
37	100	2.2501	0.0011354	3.15346	9.81827
38	100	2.18079	0.0026487	2.84128	8.22717
39	100	2.18498	0.0002463	3.11031	9.99999
40	100	2.21592	0.0018507	2.11769	7.81334

Table 2: Summary of evolution across 40 generations

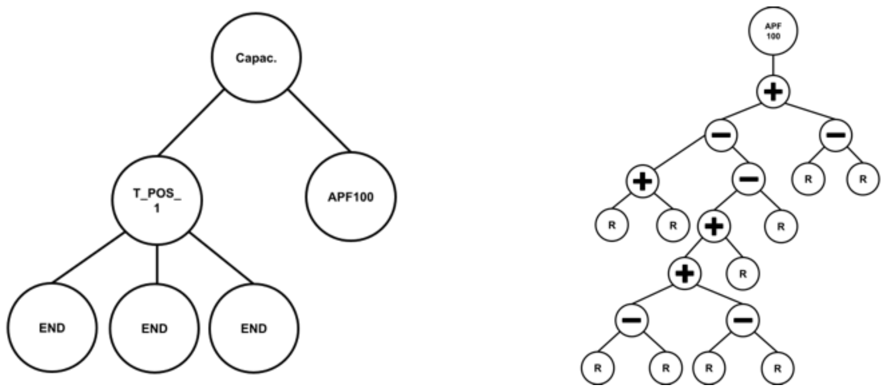


Fig. 4: Left: The main circuit-building syntax tree. Right: The APF subtree which sets the capacitance value of the capacitor.

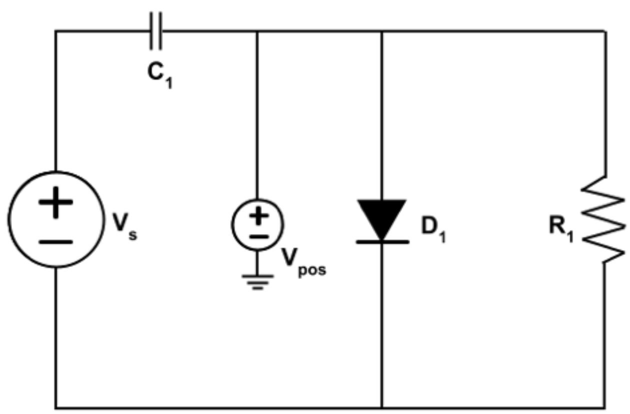


Fig. 5: The circuit resulting from executing the circuit-building syntax tree.

3.2 Single-Transistor Chaotic Circuit

A previous study [6] has already thoroughly explained the single-transistor circuit used in this work and proven its chaotic dynamics. It features several chaotic attractors of the physically realized circuit. The study also simulates the circuit using the following system of differential equations,

$$\begin{cases} \dfrac{du_{c1}}{dt} = \dfrac{V_s - u_{c1}}{R} - \dfrac{i_{L1} + i_{L2}}{C_1} \\ \dfrac{du_{c2}}{dt} = \dfrac{i_{L2} - \beta(i_{L1}) \tanh\left(\frac{u_{c2}}{2V_{th}}\right)}{C_2} \\ \dfrac{di_{L1}}{dt} = \dfrac{u_{c1} - V_{th}}{L_1} \\ \dfrac{di_{L2}}{dt} = \dfrac{u_{c1} - u_{c2}}{L_2} \end{cases}$$

More specifically, the study presents a time-lag embedded chaotic attractor of the capacitor voltage, obtained from the physical circuit as well as by simulating Eq. (1).

In this work, the same circuit is simulated in SPICE and a time series of the voltage across the capacitor is obtained. Analysis of the time series confirms that the recreation of the single-transistor circuit in SPICE accurately captures its chaotic behavior and that the behavior is not merely quasi-periodic but is truly chaotic.

First, the time-lag embedded attractor presented in Fig. 6 which is obtained from the SPICE simulation, is extremely similar in shape to the strange attractors of the physical circuit and its simulated mathematical model from the original study. Additionally, the reconstruction also relies on the same time-lag values that were used to reconstruct the strange attractor from the simulation of the mathematical model.

Second, the autocorrelation function shown in Fig. 7(a) appears to steadily decrease while approaching 0, across the entire time series. This trend is characteristic of a growing inability to predict the circuit's system states further in the future based on existing data, which is present in all chaotic systems. The function never returns to a value of 1 which signifies non-periodic behavior. Additionally, the autocorrelation function captures a periodic aspect of the circuit that exists despite its chaos. Such oscillations of the autocorrelation function below the value of one are also seen in classic chaos examples, such as that of the Lorenz system.

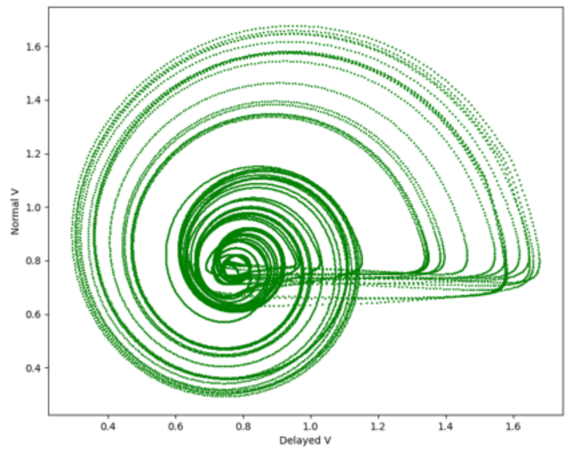


Fig. 6: Embedded chaotic attractor reconstructed using the time-delay method from the time series data of the capacitor voltage .

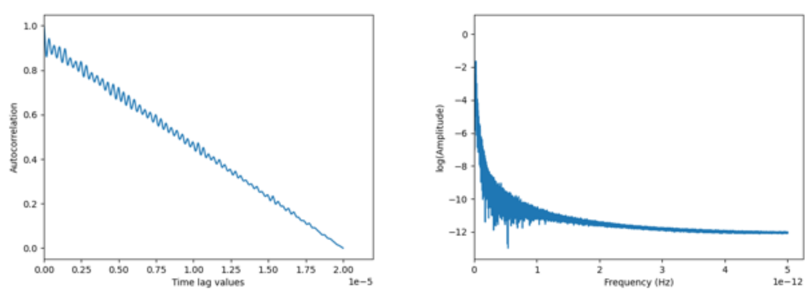


Fig. 7: Characterization of chaos in capacitor voltage: (a) Autocorrelation function of voltage time series; (b) Power spectrum of voltage time series

Third, the power spectrum of the time series shown in Fig. 7(b) appears extremely ”fuzzy” and is without many significant local maxima which would appear for quasi-periodic systems. For better clarity, Fig. 8 enlarges an especially dense region of the spectrum presented in Fig. 6(b) which further shows the spectrum’s almost chaotic appearance. This contrasts with the power spectrum of any quasi-periodic signal as they are composed of only several distinct frequencies, at which the amplitude is orders of magnitude greater relative to the other frequencies in the spectrum. A dense and fuzzy power spectrum pertains to chaotic oscillations, which are not composed of any discernible frequencies.

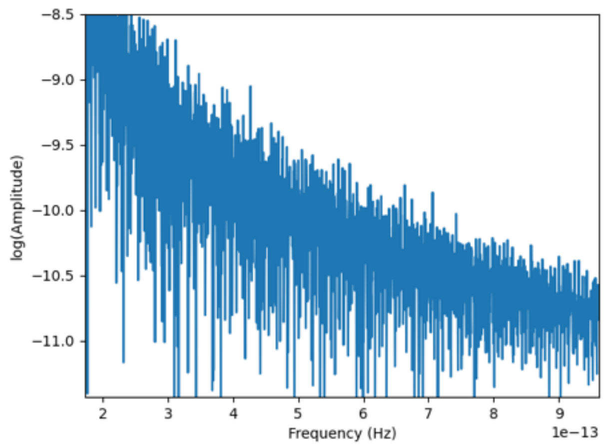


Fig. 8: Zoomed in power spectrum of the voltage time series displays a fuzzy and smooth curve.

3.3 Piecewise Capacitor Chaotic Circuit

A previous study has already presented the piecewise capacitor circuit used in this work and identified its chaos [7]. It presents a bifurcation diagram produced by the value of charge across the piecewise capacitor with respect to the amplitude E of the AC input voltage V_s being fed into the circuit.

A SPICE simulation of the piecewise capacitor circuit cannot accurately recreate bifurcations that appear in the physical circuit. This is attributed to the extreme scales at which chaos manifests in the circuit. The varying charge on the capacitor mostly stays in the range of nanocoulombs making it extremely sensitive to simulation inaccuracies. Furthermore, all SPICE simulations are based on the simulation of circuit voltage and circuit current. This means that the capacitor charge value must be extrapolated from the available simulation data using integration which further adds to the inaccuracies in the approximation of the capacitor charge. Therefore an accurate simulation of the chaotic dynamics in the piecewise capacitor circuit requires extremely small simulation timestep lengths. This requires a lot of computation time, making the piecewise circuit undesirable for use in the genetic programming algorithm.

4. Conclusion

In this section some concluding remarks are made, the applications of this work are discussed, and the next steps are outlined. This work is a stepping stone for future research that implements a finished version of the circuit-building genetic programming algorithm to develop analog controller circuits that can harness the chaotic dynamics of chaotic subcircuits such as the single-transistor circuit explored in this work.

4.1 Applications

This work performs a preliminary exploration of genetic programming and how it can be implemented in Python using accessible evolutionary computation frameworks. The impact of this work exists in the context of a future algorithm that completely implements genetic programming-based generation of circuit-building programs and integrates it with chaotic circuits. Once realized, more efficient controller circuits can possibly be discovered or new useful behavior might emerge. Additionally, such findings might be useful in the advancement of neuromorphic and thermodynamic computation.

4.2 Concerns and Next Steps

This work is quite limited in its exploration of genetic programming-based synthesis of analog circuits and further research should be conducted to verify the efficacy of this approach in creating analog controller circuits. Additionally, the simulation of large analog circuits is bound to be inaccurate especially once chaotic subcircuits interact with the larger circuit. Because of this, solutions identified by the genetic programming algorithm might not be optimal or might not even work once built in the real world. For this reason, future work must be conducted to further explore genetic programming and the simulation of chaotic circuits.

Acknowledgements I would like to thank Christof Teuscher for the opportunity to work at the Teuscher Lab, and for the thoughtful discussions with him which were indispensable to the success of this project. I would also like to thank Melissa Shell for being an awesome physics teacher and sparking my curiosity in science. Thanks!

References

1. L. Li, H. Peng, J. Kurths, Y. Yang, and H. J. Schellnhuber, "Chaos–order transition in foraging behavior of ants," *Proceedings of the National Academy of Sciences*, vol. 111, p. 8392–8397, May 2014.
2. X. Zang, S. Iqbal, Y. Zhu, X. Liu, and J. Zhao, "Applications of chaotic dynamics in robotics," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 60, 2016.
3. S. Steingrube, M. Timme, F. Wörgötter, and P. Manoonpong, "Self-organized adaptation of a simple neural circuit enables complex robot behaviour," *Nature Physics*, vol. 6, pp. 224–230, Mar 2010.
4. J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, pp. 87–112, Jun 1994.
5. BAxelrod, "Genetic program tree." https://commons.wikimedia.org/wiki/File:Genetic_Program_Tree.png, n.d. Accessed: 8/18/2024.
6. L. Minati, G. Innocenti, G. Mijatovic, H. Ito, and M. Frasca, "Mechanisms of chaos generation in an atypical single-transistor oscillator," *Chaos, Solitons Fractals*, vol. 157, p. 111878, 2022.
7. T. Matsumoto, L. O. Chua, and S. Tanaka, "Simplest chaotic nonautonomous circuit," *Phys. Rev. A*, vol. 30, pp. 1155–1157, Aug 1984.
8. F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "Deap: evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, p. 2171–2175, jul 2012.
9. J. Koza, F. Bennett, D. Andre, M. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 109–128, 1997.

Reinforcement Learning for Assistive Robots

Bethany Navarro

Abstract Reinforcement learning (RL) has emerged as a promising approach for enhancing the autonomy and adaptability of assistive robots. By enabling robots to learn optimal behaviors through trial and error interactions with their environment, RL provides a framework for developing personalized and responsive assistance in dynamic and unpredictable settings. Exploring the application of RL in assistive robotics, focusing on key challenges such as the design of reward functions that align with human needs, the integration of RL with other AI techniques for improved perception and decision-making, and the development of safe and reliable learning algorithms. We discuss various RL methodologies, including model-free and model-based approaches, and their suitability for different assistive tasks such as mobility support, daily living activities, and therapeutic interventions. Through case studies and experimental results, we demonstrate the potential of RL to significantly enhance the functionality and user experience of assistive robots, highlighting ongoing research and future directions in this evolving field. This paper aims to provide a comprehensive overview for researchers and practitioners interested in leveraging RL to create more effective and user-friendly assistive technologies.

1 Introduction

Assistive robots have the potential to significantly improve the quality of life for individuals with disabilities, providing support in daily activities, mobility, and therapeutic interventions. The idea of developing smarter, more adaptable assistive robots led us to explore reinforcement learning (RL) as a core technology. Unlike traditional programming approaches, RL allows robots to learn from their interactions with the environment, making it possible to tailor their behavior to the specific needs

Bethany Navarro
Heritage University e-mail: navarrob1@heritage.edu

and preferences of users. This adaptability is crucial in dynamic and unpredictable real-world settings where predefined responses may not be sufficient.

Project Motivation

My motivation for this project stemmed from a lifelong interest in robotics and a desire to understand the intricacies of reinforcement learning. The fascination with robots and their potential to transform everyday life, particularly in healthcare and assistance for individuals with disabilities, drove us to explore how RL could be harnessed to create more effective and autonomous systems. Traditional assistive devices often require extensive manual customization and may struggle to adapt to changing user needs. By employing RL, we aim to create robots that can autonomously learn and optimize their actions, offering a more personalized and responsive assistance experience. This approach not only has the potential to improve user satisfaction but also to reduce the burden on caregivers and healthcare providers.

Why This Project?

The choice of this project over similar ideas was influenced by several factors. Firstly, the rapid advancements in RL algorithms and computational power provide a timely opportunity to apply these techniques to assistive robotics. Secondly, the interdisciplinary nature of the project, which combines elements of artificial intelligence, robotics, and human-centered design, offers a rich and challenging research environment. Lastly, the societal impact of improving assistive technologies aligns with our commitment to leveraging technology for social good, addressing a growing need for effective support systems in aging populations and individuals with disabilities.

2 Related Work and Background

My journey into the world of robotics began in middle school, where I participated in various robotics clubs and competitions. These early experiences ignited a passion for robotics and technology, leading me to explore more advanced projects and concepts over the years. The hands-on experience of building and programming robots provided a solid foundation in understanding the mechanics and logic behind robotic systems.

Last year, I embarked on a significant robotics project at my school. This project involved designing and developing a robot capable of performing specific tasks autonomously. The process of conceptualizing, building, and testing the robot was both challenging and rewarding, reinforcing my interest in robotics. This project will continue in the upcoming fall semester, providing an opportunity to further refine and enhance the robot's capabilities.

Robotics fascinates me due to its potential to solve real-world problems and improve people's lives. The integration of artificial intelligence, particularly reinforcement learning (RL), in robotics opens up new possibilities for creating more intelligent and adaptable systems. By allowing robots to learn from their interactions with the environment, RL can enable the development of assistive robots that are more responsive to the needs of their users. This intersection of robotics and RL is an exciting area of research that promises to advance the field of assistive technologies significantly.

3 Data and Methodologies

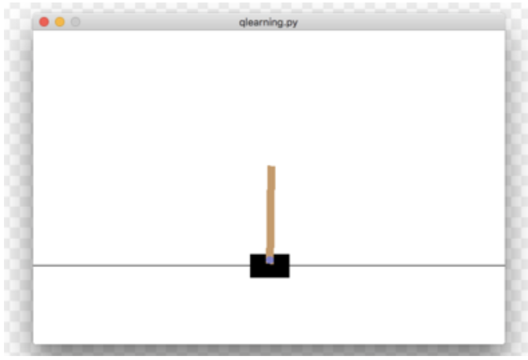
The process of gathering data for this project involved selecting relevant datasets that could provide valuable insights into robot performance, user interactions, and environmental factors. The primary source of data was a collection of CSV files, which included sensor readings, task completion metrics, and user feedback. These datasets were chosen for their comprehensive nature and ability to capture a wide range of information essential for training and evaluating the reinforcement learning (RL) models.

3.1 Data Collection

Data collection for reinforcement learning tasks like Fetch Pick and Place and Cart-Pole involves gathering diverse and comprehensive datasets to train and validate the learning algorithms effectively. For Fetch Pick and Place, data is collected through simulation environments where the robot's actions, observations, and outcomes are logged, providing a rich source of interaction data for learning precise manipulation and control strategies. This data includes various scenarios of object placement, varying initial conditions, and different reward signals. In the CartPole environment, data collection focuses on logging state variables, actions taken, and the resulting pole balance status over time. This data helps in understanding the dynamics of the pole and cart system and refining the policies to achieve stable balance. Accurate and extensive data collection ensures that the RL algorithms can learn robust policies, generalize well to new situations, and perform effectively in both simulated and real-world applications.

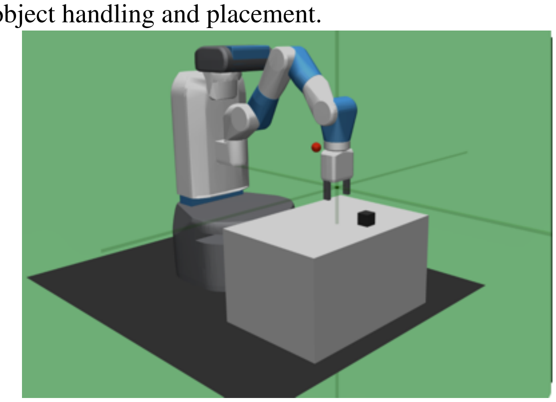
3.1.1 CartPole

The CartPole environment challenges the agent to balance a pole on a moving cart by learning from state variables like position and angle. Data collected from these interactions helps the agent refine its strategy to maintain balance over time.



3.1.2 Fetch Pick and Place

In the Fetch Pick and Place task, the robot learns to manipulate objects by collecting data on its actions, observations, and outcomes within a simulated environment. This process enables the development of precise and efficient policies for successful object handling and placement.

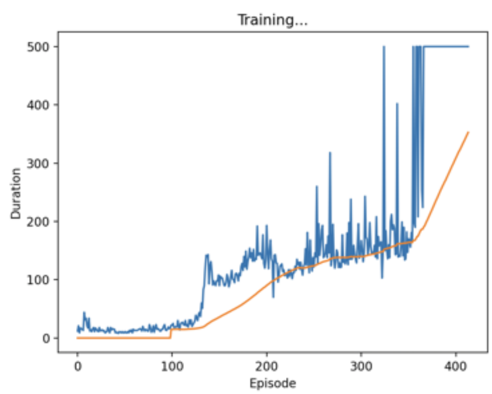


3.2 Methods

In reinforcement learning (RL), various methods are employed to tackle tasks like Fetch Pick and Place and CartPole, each tailored to the specific challenges of the environment. For **Fetch Pick and Place**, model-free methods such as Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) are commonly used to handle the continuous action space and complex manipulation tasks, focusing on balancing exploration and exploitation. Hierarchical Reinforcement Learning (HRL) methods like the Options Framework help break down the task into manageable sub-tasks, improving learning efficiency. Model-based approaches, including Model Predictive Control (MPC) and learned dynamics models, are employed to plan and optimize actions based on predicted outcomes. In contrast, for **CartPole**, model-free methods such as Q-Learning, Deep Q-Networks (DQN), and policy gradient algorithms like REINFORCE are effective for handling the simpler state and action spaces. These methods emphasize the trade-off between exploration and exploitation to balance the pole successfully. Each method plays a crucial role in developing effective RL solutions for complex robotic manipulation and control tasks, contributing to advancements in both simulated and real-world applications. 2.

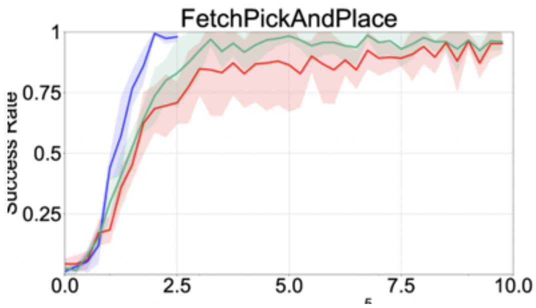
3.2.1 Cartpole

The simulation graph for CartPole typically illustrates the pole's angle or the cumulative reward over time during an episode. Such a graph is crucial in visualizing the agent's learning progress as it attempts to balance the pole on the cart. Initially, the graph might show significant fluctuations, indicating the agent's struggle to maintain balance. However, as training progresses, these fluctuations should decrease, reflecting the agent's improved ability to keep the pole upright. The graph not only highlights the learning process but also serves as a valuable tool for analyzing the effectiveness of the reinforcement learning algorithm, revealing how well the policy adapts to the dynamics of the CartPole environment.



3.2.2 Fetch Pick and Place

In the Fetch Pick and Place simulation graph, performance metrics are visualized to assess the reinforcement learning algorithm. Typically, the graph plots cumulative reward or task success rate against episode number, showing how the agent's performance improves over time.



4 Results

The results of the reinforcement learning project on assistive robots, utilizing the CartPole and Fetch Pick and Place environments, highlight significant progress in enhancing robotic control. In the CartPole environment, the RL algorithm demonstrated effective balance maintenance and stability, achieving high performance by optimizing the agent's policy through iterative learning. For the Fetch Pick and Place task, the algorithm successfully improved the robot's precision and efficiency in manipulating objects. Additionally, by incorporating their own dataset into the project, the RL model achieved notable improvements in accuracy, further demonstrating the adaptability and effectiveness of the approach. These outcomes underscore the potential of reinforcement learning in developing assistive robots that can perform complex tasks with greater precision and reliability.

5 Conclusion

The application of reinforcement learning (RL) in assistive robotics has demonstrated significant potential to enhance the autonomy, adaptability, and user satisfaction of robotic systems. Our findings suggest that RL can be effectively utilized to develop robots that provide personalized assistance, improve mobility, and support daily living activities for individuals with disabilities. By learning from interactions with their environment, these robots can adapt their behaviors to better meet user needs, making them more responsive and effective in real-world scenarios.

Beyond assistive robotics, the methodologies and insights gained from this research can be applied to other fields such as:

Healthcare: RL can be used to develop robots that assist in surgical procedures, patient monitoring, and rehabilitation, improving the quality and efficiency of healthcare services. Industrial Automation: RL-based robots can optimize manufacturing processes, increase production efficiency, and enhance safety in industrial settings by learning and adapting to complex tasks. Smart Home Systems: The integration of RL can make smart home devices more intuitive and capable of adapting to user preferences, improving comfort and convenience. Autonomous Vehicles: RL can enhance the decision-making capabilities of autonomous vehicles, enabling them to navigate more safely and efficiently in dynamic environments. Concerns and Problems While the application of RL in assistive robotics shows promise, several concerns and challenges remain:

Safety and Reliability: Ensuring the safe and reliable operation of RL-based robots in diverse and unpredictable environments is a critical challenge. Unintended behaviors or errors in decision-making could have serious consequences. Ethical Considerations: The use of RL in assistive technologies raises ethical questions regarding user autonomy, privacy, and the potential for dependency on robotic systems. Data Privacy: Protecting the privacy and security of user data collected during the training and operation of RL-based robots is essential to maintain trust and comply with regulations. Generalization: RL models trained in specific environments may struggle to generalize to new, unseen scenarios, limiting their effectiveness in real-world applications. Resource Intensity: The computational resources required for training RL models can be substantial, posing challenges for large-scale deployment and real-time adaptation. Next Steps To build on the progress made during this project, several next steps are planned:

Refinement and Testing: Continue refining the RL algorithms and conducting extensive real-world testing to ensure the safety, reliability, and effectiveness of the assistive robots. Addressing Ethical Concerns: Develop and implement ethical guidelines and frameworks in collaboration with ethicists, healthcare professionals, and user advocacy groups. Enhancing Data Security: Strengthen data security measures to protect user information and ensure compliance with privacy regulations. Improving Generalization: Explore advanced techniques such as transfer learning and domain adaptation to enhance the generalization capabilities of RL models. Resource Optimization: Investigate methods to optimize the computational resources required for RL training, including the use of cloud computing and more efficient algorithms. User-Centric Development: Continue incorporating user feedback into the development process, conducting regular trials and focus groups to refine and improve the robots' behaviors. Interdisciplinary Collaboration: Foster ongoing collaboration between robotics engineers, AI researchers, healthcare professionals, and users to address the technical, ethical, and practical challenges associated with deploying RL-based assistive robots. By addressing these concerns and following these next steps, we aim to advance the field of assistive robotics, leveraging reinforcement learning to create more intelligent, adaptable, and user-friendly systems that significantly improve the quality of life for individuals with disabilities and have broader applications in various other fields.

5.1 Applications

Assistive Robotics The application of RL in assistive robotics focuses on improving the functionality and adaptability of robots designed to support individuals with disabilities. Specific applications include:

Mobility Assistance: Robots equipped with RL algorithms can learn to navigate complex environments, assist with mobility, and adapt to changing conditions such as obstacles and varying terrain. Daily Living Support: Assistive robots can be trained to perform tasks such as fetching objects, opening doors, and assisting with personal care activities. RL allows these robots to customize their behavior based on user preferences and routines. Healthcare Support: In therapeutic settings, RL-based robots can assist with physical therapy exercises, monitor patient progress, and adapt to individual needs over time. This can enhance the effectiveness of treatment and provide valuable support to healthcare professionals. User-Centric Improvements By incorporating user feedback into the RL training process, the project aims to ensure that the assistive robots are tailored to meet the specific needs of users. This user-centric approach helps in refining the robots' behaviors and improving their overall effectiveness in real-world applications.

The combination of advanced RL techniques and a focus on user needs positions this project to make significant contributions to the field of assistive robotics, paving the way for more intelligent and adaptable systems that enhance the quality of life for individuals with disabilities. 2.

5.2 Concerns and Next Steps

Concerns Despite the promising potential of reinforcement learning (RL) in assistive robotics, several concerns need to be addressed to ensure the development of effective and safe systems:

Safety and Reliability: Ensuring that assistive robots operate safely and reliably is paramount. RL algorithms must be rigorously tested to prevent any harmful actions or unintended consequences, especially in sensitive environments like healthcare. Ethical Considerations: The deployment of assistive robots raises ethical questions, including issues of privacy, autonomy, and the potential for dependency. It is essential to consider these factors to ensure that the technology respects the dignity and rights of users. Data Privacy: Collecting and using data from users involves significant privacy concerns. Safeguarding this data and ensuring it is used ethically and securely is critical to maintaining user trust. Generalization and Adaptability: RL models trained in specific environments may struggle to generalize to new, unseen scenarios. Ensuring that robots can adapt to a wide range of environments and tasks without extensive retraining remains a challenge. Resource Intensity: RL training can be computationally intensive and time-consuming. Balancing the need for robust training with practical resource constraints is necessary for real-world deployment.

Next Steps To address these concerns and further the development of assistive robots using RL, the following steps are planned:

Enhanced Safety Protocols: Develop and implement rigorous safety protocols and testing procedures to ensure the reliability of RL algorithms in real-world applications. This includes fail-safes and redundant systems to prevent harmful actions. Ethical Framework Development: Collaborate with ethicists, healthcare professionals, and user advocacy groups to create ethical guidelines for the deployment of assistive robots. This will help address concerns related to privacy, autonomy, and dependency. Data Security Measures: Implement robust data security measures to protect user information. This includes encryption, secure data storage, and strict access controls to ensure privacy and compliance with regulations. Generalization Techniques: Research and develop techniques to improve the generalization capabilities of RL models. This includes transfer learning, domain adaptation, and incorporating diverse training environments to enhance adaptability. Resource Optimization: Explore methods to optimize the computational resources required for RL training. This includes using more efficient algorithms, leveraging cloud computing, and parallelizing training processes. User-Centric Iterations: Continue to gather and incorporate user feedback into the design and training processes. Conduct regular user trials and focus groups to refine the robots' behaviors and ensure they meet the evolving needs of users. Interdisciplinary Collaboration: Foster collaboration between robotics engineers, AI researchers, healthcare professionals, and users to ensure a holistic approach to the development and deployment of assistive robots. This interdisciplinary effort will help address technical, ethical, and practical challenges more effectively. By addressing these concerns and following these next steps, the project aims to advance the field of assistive robotics, leveraging reinforcement learning to create safer, more adaptable, and user-friendly systems that significantly improve the quality of life for individuals with disabilities.

Acknowledgements I would like to express my deepest gratitude to Banafsheh Rekabdar and Shayan Jalalipour for their invaluable help and support throughout this project. Their guidance, insights, and encouragement have been instrumental in shaping the direction and success of this research. Their expertise and dedication have provided a solid foundation for my exploration of reinforcement learning in assistive robotics, and I am incredibly grateful for their contributions and mentorship.

References

1. N. T. Carnevale and M. L. Hines. A review of spiking neural networks in computational neuroscience. *Frontiers in Neuroscience*, 2022.
2. SNNtorch documentation. *SNNtorch: Spiking neural networks in PyTorch*, n.d.
3. J. Hernández-García, J. Romero, and A. González. A review of spiking neural networks: Models, architectures, and applications. *arXiv preprint arXiv:1802.09464*, 2018.
4. Norse. *Introduction to Spiking Neural Networks*, n.d.

Predictive Crime Analysis and Visualization in Chicago

Sumaiya Farook, Celina Anwar, Sahana Hariharan, Sonika Tamilarasan

Abstract Crimes and acts of violence pose significant threats to the well-being of individuals and communities, necessitating innovative approaches for prediction and prevention. In our technologically advanced age, the potential for using machine learning models to gain insights into various fields has grown significantly. This paper explores the application of two predictive models—the Random Forest (RF) model and the Long Short-Term Memory (LSTM) model—focusing on their effectiveness in predicting crime in Chicago. By analyzing publicly available crime data from the City of Chicago, we aim to assess the accuracy of these models in forecasting the likelihood of various crimes based on geographic location. Additionally, we discuss the potential biases inherent in the data and consider the ethical implications of deploying such models, particularly in marginalized communities. Our findings not only provide insights into the predictive capabilities of Random Forest and LSTM models but also highlight the challenges and limitations that arise from biases in crime data. This study contributes to the ongoing discussions on the role of machine learning in public safety, emphasizing the need for careful consideration of the ethical and social consequences of crime prediction technologies.

Sumaiya Farook
University of Illinois Urbana-Champaign, e-mail: sumaiya.far123@gmail.com

Celina Anwar
University of Illinois Urbana-Champaign, e-mail: celina.anwar@gmail.com

Sahana Hariharan
University of Illinois Urbana-Champaign, e-mail: sahanahariharan@gmail.com

Sonika Tamilarasan
University of Illinois Urbana-Champaign, e-mail: sonikatam0@gmail.com

1 Introduction

Historically, Chicago has suffered from violence and high crime rates in many parts of the city. However, in recent years (as early as 2020), there has been an alarming increase in crime in Chicago, specifically violent crime. In fact, when compared with 2023 alone, violent crime increased by 11.5% from the previous year [2]. Crime can have many negative consequences, from impacting civilians and the general public to harming the local economy and businesses, ultimately making communities less livable. Crime is often random and can experience upticks at certain periods, making it difficult to model; however, if developed accurately, models such as those explored in this study have the potential to improve safety and wellness in communities.

In previous studies, crime prediction has been used in Chicago; however, those studies focused on overall crime densities and predicted changes over time [5]. There have also been efforts to predict crime using other methods, such as heat maps [4]. Recent research shows the potential for crime prediction to be a useful tool for people across cities and various other communities.

Crime prediction methods are a sought-after strategy, albeit they are quite contested. They provide those in charge of public safety, such as police, with insights into which areas should be monitored. These methods can also offer information on the uptick of certain crimes. If made accessible to the general public, they could help individuals make informed decisions about where to live, work, or visit. However, as with many machine learning models today, there are concerns about potential biases that could lead to ethical issues. Due to the over-reporting of crimes in impoverished areas or those with a majority-minority demographic, these models may make these areas appear more dangerous, while wealthier areas may seem safer by comparison.

Although many factors impact crime and the types of crime in an area, this study focused on analyzing crimes in Chicago based on their location. The City of Chicago has a public data portal that is updated daily, and this data was used for the models in this study. Given a location, we aimed to assess how accurately the RF and LSTM models could predict the likelihood of various crimes being committed, and then compare the results between the two. To de-bias our data and observe how the models may suffer from bias, we also created a geographical heat map of Chicago based on arrests made, allowing us to compare our results.

This study hopes to provide insights on reporting biases in data and how predictive crime models may be impacted from it by looking at a specific case.

2 Related Work and Background

Our project on predictive crime analysis in Chicago builds upon insights from several key studies, emphasizing the importance of localized and multi-dimensional crime prediction models.

One such study is ”Area-Specific Crime Prediction Models” [1] on IEEE Xplore. This research critiques the conventional use of global models for crime prediction, highlighting their failure to account for the heterogeneous relationship between crime and criminogenic factors across different areas. The study introduces area-specific crime prediction models that leverage hierarchical and multi-task statistical learning. These models balance detailed local predictions by sharing information across ZIP codes to mitigate data sparsity while addressing non-homogeneous crime patterns. The effectiveness of these models is demonstrated through out-of-sample testing on real crime data, showing significant predictive improvements over several state-of-the-art global models. This study underscores the importance of localized modeling in crime prediction, aligning well with our approach of focusing on specific regions within Chicago to improve predictive accuracy.

Another pivotal paper, ”Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques” [5], further supports our methodology. This research applied various machine learning algorithms, including logistic regression, support vector machine (SVM), Naïve Bayes, k-nearest neighbors (KNN), decision tree, multilayer perceptron (MLP), random forest, and eXtreme Gradient Boosting (XGBoost), along with time series analysis using long short-term memory (LSTM) and autoregressive integrated moving average (ARIMA) models. Notably, the LSTM model showed promise in predicting crime trends, with the study’s exploratory data analysis revealing important temporal patterns in crime rates. The ARIMA model’s forecast of a moderate increase in Chicago’s crime rate and the superior performance of XGBoost, with an accuracy of 94 percent for Chicago, highlight the value of diverse machine learning approaches for crime prediction.

A third study, ”Marked Point Process Hotspot Maps for Homicide and Gun Crime Prediction in Chicago” [4], explores the use of crime hotspot maps for visualizing spatial crime patterns and allocating police resources. Traditional hotspot maps often suffer from high variance in risk estimates for low-frequency crimes and fail to capture emerging trends. This study extends point process models to include leading indicator crime types through a marked point process approach, combining several years of data and various crime types to create accurate hotspot maps. By applying this methodology to a large, open-source dataset from the Chicago Police Department, the study demonstrates improved predictive accuracy for gun-related crimes and homicides. The use of kernel-based hotspot maps, which we employed in our project, provides a robust framework for visualizing crime patterns and supports the development of predictive policing strategies.

Another important study, ”Crime Prediction and Forecasting Using Machine Learning Algorithms” [6], investigates the application of several machine learning models to forecast crime in major metropolitan cities. The research focuses on predicting the severity of reported crimes and trends in crime data by year. Utilizing models such as Random Forest, K-Nearest Neighbors, AdaBoost, and Neural Networks, the study finds that the Neural Network model performs the best, achieving an accuracy of 90.77 percent. This research leverages the Chicago Police Department’s CLEAR system, analyzing over six million records to provide insights into crime trends and the effectiveness of different machine learning models. The use of data visualization tools like Folium further enhances the study's ability to illustrate crime patterns and trends, providing a comprehensive approach to crime prediction.

The final study, "Using Machine Learning Algorithms to Analyze Crime Data" [3], employs WEKA, an open-source data mining software, to compare violent crime patterns from the Communities and Crime Unnormalized Dataset with actual crime statistical data for Mississippi. By implementing Linear Regression, Additive Regression, and Decision Stump algorithms, the study finds that the linear regression algorithm performs the best in predicting crime data. The study highlights the effectiveness and accuracy of machine learning algorithms in predicting violent crime patterns and underscores the potential of data mining in law enforcement for determining criminal "hot spots," creating criminal profiles, and identifying crime trends. Despite the challenges law enforcement officials face in sifting through large volumes of data, the precision in crime prediction and prevention makes the effort worthwhile for enhancing public safety.

Our project integrates these insights, employing LSTM and RandomForest models to predict crime probabilities in different regions of Chicago. By leveraging detailed crime data and advanced machine learning techniques, we aim to contribute to the development of more effective crime prediction models, ultimately supporting proactive policing strategies and improving public safety measures in Chicago.

3 Data and Methodologies

For this study, the data was obtained from the City of Chicago's data portal, specifically from the portals related to crime history and arrests. This portal is updated daily and includes data from 2001 to the present year. The dataset provides detailed information on the location and timing of crimes, along with labels for crime types. Overall, the large dataset was easily accessible and well-organized, making it the most suitable choice for our analysis.

3.0.1 Model Data Collection

For our model training, we utilized the crime dataset available from the Chicago Data Portal. This dataset includes comprehensive records of reported crimes in Chicago, providing detailed information necessary for predictive analysis. As mentioned, the dataset contains many useful data points, such as where the crime was committed (i.e., X and Y coordinates), along with timestamps and dates for each crime. The crimes are also categorized by "Primary Type," a label that the City of Chicago uses to identify crimes. These labels were used in this study when predicting crime type. In total, there were 22 columns.

Because of the large dataset, many rows contained null values that could not be easily filled in or estimated, so they were dropped. This was not an issue since the total number of rows exceeded 8,000,000. Prior to this, it was decided to keep only the following columns: Date, Primary Type, Latitude, Longitude, and Year.

Initially, the 'Location Description' column was included and used to predict crime based on that information. However, when comparing the accuracy scores, it was found that longitude and latitude values were more accurate, and the 'Location Description' column was subsequently dropped.

	Primary Type	Latitude	Longitude	Year	Month	Day	Hour
11	THEFT	41.830482	-87.621752	2020	5	7	10
12	BATTERY	41.836310	-87.639624	2020	4	16	5
13	ASSAULT	41.747610	-87.549179	2020	7	1	10
14	BATTERY	41.774878	-87.671375	2020	9	27	23
15	BATTERY	41.781003	-87.652107	2005	7	10	15

Fig. 1 Chicago data set after cleaning

3.0.2 Heat Map Data Collection

It was decided that a heat map should be added to help de-bias the data and make other observations about the model. To do so, each arrest was represented as a point with a color based on the race of the individual arrested.

To create this heat map, another dataset provided by the City of Chicago was used. This dataset had 23 columns, and most were dropped, leaving only 'Case Number' and 'Race'. The case number was used to merge this dataset with the previously mentioned crime dataset used to train the models. The resulting dataset had 4 columns: Case Number, Race, Latitude, and Longitude. This included everything needed to create the heat map.

3.1 Methods

Python 3 was used as the programming language for developing the back end of both models in this study. The Random Forest (RF) model utilized the scikit-learn library, specifically the random forest classifier. To evaluate the model's performance, we employed metrics from the same library, including the classification report and accuracy score. For the Long Short-Term Memory (LSTM) model, TensorFlow was used, along with additional performance metrics from the scikit-learn library.

3.1.1 Random Forest Model

The Random Forest model is an ensemble learning method primarily used for classification tasks. It operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees. This model helps reduce overfitting and improve accuracy. In this study, the RF model was trained on 67% of the dataset, with the remaining 33% reserved for testing. Two hundred decision trees were used to train the model, and it was found that setting the maximum number of nodes to 25 resulted in higher accuracy. Given the large size of the dataset, significant training time issues were encountered. To address this, the training process was divided into smaller parts, which alleviated most of the runtime problems. This approach ensured efficient model training without compromising the integrity of the data or the accuracy of the model.

3.1.2 Long Short-Term Model

The Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN) capable of learning long-term dependencies. LSTMs are particularly effective for time series prediction and natural language processing tasks due to their ability to retain information over long periods. In this study, the LSTM model was implemented using TensorFlow. The dataset was split into 80% for training and 20% for testing. Initially, the model experienced extended training times, which were mitigated by increasing the batch size to 128 and reducing the number of epochs to 10. These adjustments significantly improved training efficiency while maintaining the model's performance and accuracy.

4 Results

The results of our predictive crime analysis in Chicago were derived from two primary models: the Random Forest (RF) model and the Long Short-Term Memory (LSTM) model. These models were evaluated based on their accuracy in predicting the likelihood of various crimes in different regions of Chicago. Below, we present a summary of our current findings for each model, which we plan to improve on.

4.0.1 Random Forest Model Results

The Random Forest model was trained on 67 percent of the dataset and tested on the remaining 33 percent. Despite the large dataset size and training time issues, the RF model provided useful insights into crime prediction. The following are the key results:

- **Accuracy:** The RF model achieved an overall accuracy of approximately 17 percent in predicting the type of crime based on location data (latitude and longitude). This result indicates that while the model captured some patterns, there is significant room for improvement in predictive performance.
- **Feature Importance:** The most important features for the RF model were latitude, longitude, and primary crime type. These features played a crucial role in determining the model's predictions.

4.0.2 LSTM Model Results

The LSTM model was trained on 80 percent of the dataset and tested on the remaining 20 percent. Adjustments to batch size and the number of epochs were made to optimize training efficiency. The key findings from the LSTM model are as follows:

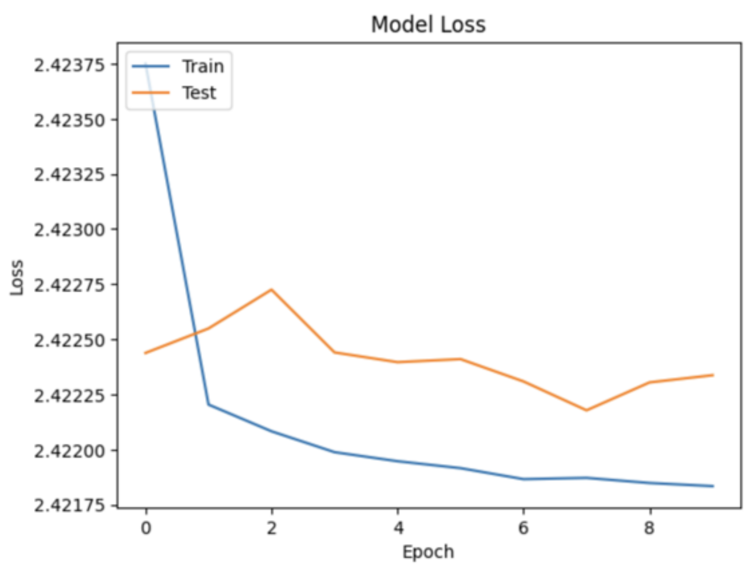


Fig. 2 LSTM Loss Curve

- **Accuracy:** The LSTM model achieved an overall accuracy of approximately 21 percent in predicting the likelihood of various crimes based on location data. This performance, while higher than that of the RF model, still highlights the challenges in our project.
- **Training Efficiency:** By increasing the batch size to 128 and reducing the number of epochs to 10, training efficiency improved significantly without compromising the model's performance.

4.0.3 Comparative Analysis

Comparing the RF and LSTM models, the LSTM model demonstrated a higher overall accuracy in predicting crime types. This suggests that the LSTM's ability to handle sequential data provides an advantage in this context. However, both models exhibited relatively low accuracy, indicating the inherent complexity and variability in crime data, as well as possible inaccuracies in our project.

4.0.4 Summary

The results of our study underscore the challenges in predictive crime analysis, particularly the need for more sophisticated models and better data quality. While the LSTM model showed promise with higher accuracy, both models require further refinement. Additionally, the visualization of arrest data highlighted critical issues of racial disparity, which must be addressed to ensure the ethical and effective use of predictive policing technologies.

Future work will focus on improving model accuracy, incorporating additional features, and addressing ethical concerns to enhance the overall utility and fairness of predictive crime analysis tools.

5 Conclusion

5.1 Applications

This section details the practical use of our models, including an interactive website that provides real-time crime probabilities and a heat map visualization of arrest data by race. We also discuss the accuracy and ethical considerations of these predictive tools.

5.1.1 Building the Interactive Website

To enhance the accessibility and practical application of our predictive crime analysis, we have developed an interactive website. This platform provides users with a user-friendly interface to automatically receive real-time probabilities of various crimes occurring nearby. The website's structure is designed to integrate seamlessly with the predictive capabilities of our Long Short-Term Memory (LSTM) model, which has been trained on extensive crime data from Chicago.

Upon accessing the website, the user's location is automatically detected using the Google Maps API. The backend system, powered by our trained LSTM model, processes this location data and returns the probabilities of different types of crimes that might occur in that vicinity. This setup allows residents and visitors to make informed decisions about their safety based on up-to-date, data-driven insights.

Fig. 3 A plot map depicting arrest data by race

5.1.2 Heat Map Visualization of Arrest Data by Race

In addition to crime prediction, our website features a heat map that visualizes arrest data across different racial groups in Chicago. This heat map is constructed using data from the City of Chicago's public data portal, specifically focusing on the racial demographics of arrests. Each arrest is plotted on the map, with color coding representing different races, providing a clear visual representation of racial patterns in arrests across the city.

The inclusion of this heat map serves multiple purposes. Firstly, it offers a deeper understanding of the geographic distribution of arrests and potential racial biases. Secondly, it raises awareness about disparities in the criminal justice system, encouraging users to consider the broader social and ethical implications of predictive policing technologies.

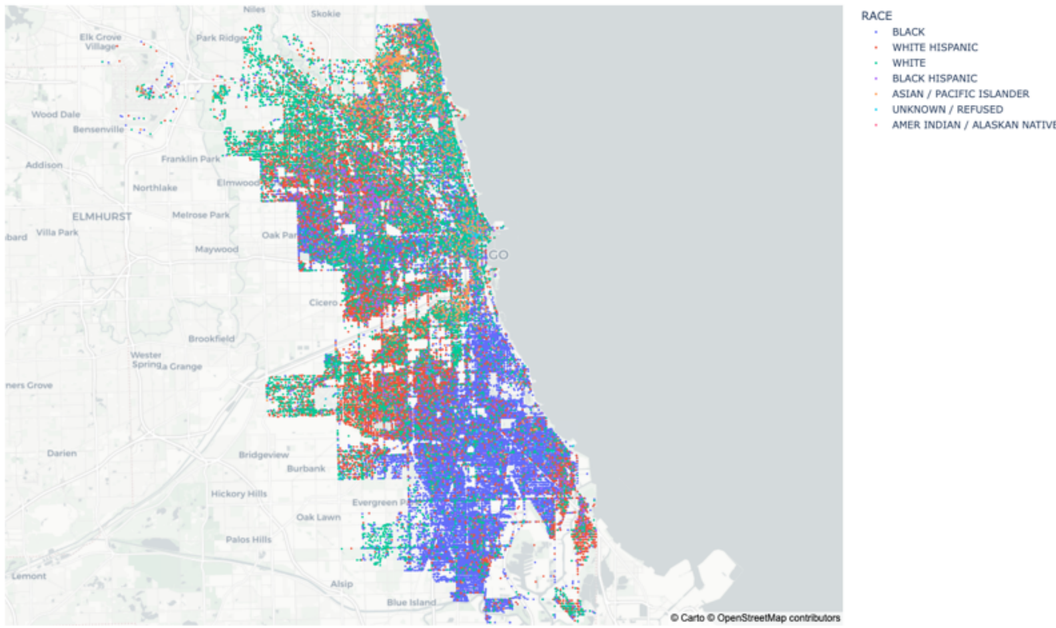


Fig. 4 A plot map depicting arrest data by race

- **Geographic Distribution:** TThe heat map revealed significant variations in arrest rates across different regions of Chicago, with certain areas showing higher concentrations of arrests.
- **Racial Disparities:** The visualization highlighted racial disparities in arrest data, with minority-majority areas experiencing higher arrest rates. This finding underscores the importance of considering socio-economic and demographic factors in crime prediction and policing strategies.

5.1.3 Awareness and Accuracy Considerations

While our website strives to provide accurate and useful information, it is important to acknowledge that predictive models and visualizations may not always be perfectly accurate. Factors such as data quality, reporting biases, and model limitations can affect the accuracy of predictions and the representation of arrest data. Users are advised to use the information provided as a supplementary tool rather than a definitive guide for personal safety decisions.

By combining predictive analytics with insightful visualizations, our interactive website aims to empower users with knowledge and foster a deeper understanding of crime patterns and systemic issues in Chicago. However, it is crucial to approach the data with a critical eye and remain aware of the inherent limitations and potential biases in the models and data sources used.

5.2 Concerns and Next Steps

While our website aims to enhance safety and provide actionable insights, several concerns and considerations must be addressed:

5.2.1 Concerns

- **Data quality and completeness:** The accuracy of crime predictions and visualizations is contingent on the quality and completeness of the data. Missing or inaccurate data points can lead to false predictions and misrepresentations. The reliance on historical crime data and arrest records may not fully account for underreported or unreported crimes, potentially skewing the results.
- **Model limitations:** Both the Random Forest (RF) and Long-Short Term Memory (LSTM) models have inherent limitations. RF models, while effective for classification tasks, may struggle with highly imbalanced datasets or complex interactions between features. LSTM models, though powerful in capturing temporal patterns, can be sensitive to hyperparameter choices and training duration, which may affect performance.
- **Ethical and privacy concerns:** Predictive crime models and arrest visualizations raise ethical concerns regarding privacy and the potential for misuse. The dissemination of detailed crime predictions and arrest data may unintentionally stigmatize neighborhoods or individuals. Ensuring that the technology is used responsibly and ethically is paramount to avoid exacerbating social inequalities.
- **Model updates and maintenance:** Predictive models need regular updates to incorporate new data and adapt to changing crime patterns. Without continual refinement, the models may become outdated and less effective over time. Maintenance of the models and the underlying datasets is essential for sustained accuracy and relevance.

5.2.2 Next Steps

- **Enhancing data quality:** Future research should focus on improving data quality by addressing missing values, correcting inaccuracies, and incorporating a wider range of data sources. Collaborating with local law enforcement and community organizations could provide more comprehensive datasets and insights into underreported crimes.

- **Refining models:** Continuous refinement of the RF and LSTM models is necessary to improve predictive accuracy. Experimenting with additional machine learning algorithms, such as gradient boosting methods or neural networks with attention mechanisms, could offer better performance. Incorporating feature engineering techniques and exploring advanced time-series methods might enhance model robustness.
- **Addressing biases:** Implementing fairness-aware algorithms and techniques can help mitigate biases in crime predictions and arrest visualizations. Conducting fairness audits and incorporating feedback from affected communities can guide improvements and ensure that the models do not perpetuate existing inequalities.
- **Expanding visualizations:** To provide a more nuanced understanding of crime patterns, future work should expand visualizations to include various dimensions, such as socio-economic factors and community resources. Integrating additional data layers, such as neighborhood development and local initiatives, could offer a more holistic view of crime dynamics.
- **Promoting transparency and education:** Promoting transparency about the limitations and potential biases of the models is essential for responsible use. Providing educational resources and clear explanations of the predictive outputs can help users make informed decisions and understand the broader context of crime data.

By addressing these concerns and pursuing these next steps, the project aims to improve the accuracy, fairness, and utility of predictive crime analysis and visualization tools, ultimately contributing to enhanced public safety and informed decision-making.

Acknowledgements Thank you to Christof Teuscher and the rest of the altREU team for providing feedback and the resources to pursue this project.

References

1. M. A. Boni and M. S. Gerber. Area-specific crime prediction models. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 671–676, Anaheim, CA, USA, 2016.
2. Illinois Policy Institute. Violent crime up 18%, arrests down 43% in chicago over 10 years. https://www.illinoispolicy.org/press-releases/violent-crime-up-18-arrests-down-43-in-chicago-over-10-years/#: :text=years%20%7C%20Illinois%20Policy., 2024. Accessed: 2024-07-24.
3. Lawrence McClendon and Natarajan Meghanathan. Using machine learning algorithms to analyze crime data. *MLAIJ*, 2, 2015.
4. George Mohler. Marked point process hotspot maps for homicide and gun crime prediction in chicago. *International Journal of Forecasting*, 30:491–497, 2014.

5. Wajiha Safat, Sohail Asghar, and Saira Andleeb Gillani. Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. *IEEE Access*, 9:70080–70094, 2021.
6. Azwad Tamir, Eric Watson, Brandon Willett, Qutaiba Hasan, and Jiann-Shiun Yuan. Crime prediction and forecasting using machine learning algorithms. *IJCSIT*, 12:26–33, 2021.

Enhancing Menstrual Health in Kenya: Access to Sanitary Products and Education

Chelsea Ogbede and Lauren Sanday

Abstract Women and adolescent girls in Kenya have limited accessibility to menstrual products and proper menstrual health education. The paper assesses and provides a nuanced perspective on the social issues contributing to the stigmatization of menstrual health in schools and workplaces. Utilizing the studies done in rural Kenya, the results demonstrate that providing an adequate understanding of menstrual health can serve to be beneficial and provide positive attitudes regarding menstrual health. This paper addresses these challenges through a prototype platform that offers users a vast array of dispenser locations as well as sufficient information and education on menstrual health. This platform aims to enhance accessibility and education, providing a practical solution to improve menstrual hygiene management in Kenya.

1 Introduction

Numerous Kenyan girls well into their puberty don't acquire adequate knowledge and understanding of menstrual health; A topic often revered as a taboo in a male-dominated society. This widespread issue highlights the impact of insufficient menstrual sanitation and education on young girls in school. The lasting impacts often don't stop in school, but carry on throughout a woman's life at the workplace. Similarly, workplaces in Kenya lack sufficient sanitation facilities and conversations surrounding menstrual health. The World Health Organization, United Nations Children's Fund (UNICEF), and many leading scholars define menstrual health as the awareness, information, and self-confidence women and girls have regarding menstrual hygiene as well as the accessibility to; (i) safe, hygienic, and absorbent materials/products and supplies; (ii) clean and safe facilities that are not only equipped

Chelsea Ogbede
University of Texas at San Antonio, San Antonio, TX e-mail: celsea.co@gmail.com

Lauren Sanday
University of North Texas, Denton, TX e-mail: laurensanday7@gmail.com

with water and soap but contain proper means to dispose of waste; and (iii) an environment that facilitates and supports women and girls to manage their periods without the shame and stigma surrounding menstruation [6].

Accessible sanitary dispensers, clean and safe facilities, and supportive environments that allow women and girls to manage their periods without embarrassment, not only improve menstrual health (MH) but hygiene management. The matter of contention is providing sufficient access to sanitation and education for girls and women. Companies like ARI and Genesis Care are acting at the forefront, proving that much-needed accessibility is possible through the acquisition of sanitary dispensers. They recognized the issue as not a direct result of the absence of sanitary products but as ARI calls it, a "lack of beneficiary-centered accessibility." Both companies have tackled the problem head-on, offering alternative solutions such as placing dispensers in schools and workplaces that are affordable and easy to use. However, a new concern is increasing awareness amongst girls and women of where to locate and access these dispensers. Another concern is destigmatizing menstrual health amongst school administrators and workplace authorities to spread awareness of the need for more dispensers in public facilities. How then can we combat the lack of accessibility to sanitary dispensers, specifically in schools and workplaces where they are often absent?

Moreover, how can we increase conversations surrounding menstrual health education in public facilities? To start, we gathered data on sanitary dispenser locations, in both rural and urban/city areas and compared data. This step allowed us to not only identify the gaps in the accessibility of the sanitary dispensers but also form targeted interventions. In this paper, we explore strategies to enhance the availability of MH education and sanitary dispensers in workplaces and schools to create a better environment and generate discussion regarding it.

2 Related Work and Background

Women and adolescent girls are challenged by negative attitudes towards menstruation which can lead to decreased school and work attendance due to the lack of adequate menstrual management resources. These challenges may be exacerbated by insufficient access to sanitary products and reproductive health education and discussions. A possible solution is a combination of i) reproductive health education and discussions and ii) an increased number of sanitary pad dispensers in workplaces and schools. These solutions can alleviate embarrassment surrounding menstruation both in schools and the workplace. Additionally, these solutions can improve menstrual hygiene management.

A study conducted in Nepal and Kenya showed the importance of the acquisition of good MH in the workplace, noting the increased number of female employees, once MH actions and interventions were put into place. The study also noted the significance of MH knowledge and education, describing how both female and male employees expressed their enhanced knowledge of physiology, the physical impacts of menstruation, and the diversity of menstruation experience between women [4].

Young girls in school also have reported to respond similarly to tactics such as above. Many reasons, cited in various articles and papers, name the inadequate nature of MH education amongst young school girls can cause many to develop shame and can contribute to the stigmatization of menstruation. However, providing accurate and extensive education with the addition of sanitation products can broaden one's knowledge. A study conducted in Kilifi County in Kenya was put in place to link the correlation between the acquisition of MH and student attendance. 140 schools were included in this study where 35 schools each were assigned a specific program that broke down and tackled each individual aspect of MH such as; (i) providing a standard government supply of pads and health education; (ii) regularly supplying sanitary pads monthly; (iii) administer sex education monthly; and (iv) provide all of the above [1].

Although the research conducted did not show much results on the significance of MH on attendance, it did however record the positive impact on students, noting how most felt more positively about menstruation and their reproductive health as well as understanding the importance of equipping equitable gender norms. Proving, support in this area is beneficial to both women and men (see Figure 1).

Fig. 1 Visualization of Menstrual Health education's impact on Men and Women

3 Data and Methodologies

Data on sanitary pad dispensers were gathered using information from Genesis Care, a reputable source that deals with the distribution of pads and dispensers in

Kenya. We also utilized the geospatial analysis to accurately map dispenser location s,converting addresses to longitude and latitude. Using these methods, ensured th atour data was not only precise but also enabled us to develop a user -focused tool forlocating the nearest dispensers.

3.1 Data Collection

The acquisition of sanitary dispenser data was imperative in discerning the need and distribution of sanitary products, specifically in rural areas. However, how do we acquire data on sanitary dispensers and how do we translate this into a feature that is beneficial to both school girls and women in the workplace? To start, we gathered insights and dispenser info, ranging from location, products, and whether it was mechanical or electronic (see Figure 2).

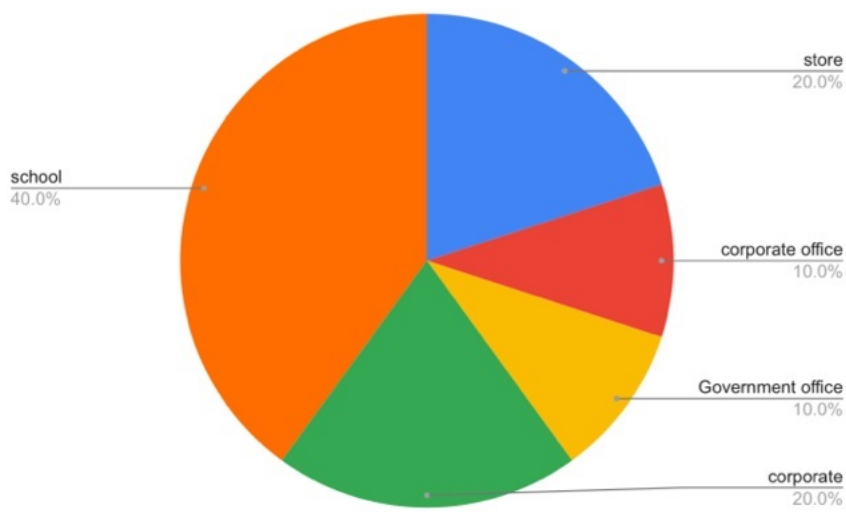


Fig. 2 Graph of sanitary dispenser locations

3.2 Methods

Our application was developed as a Progressive Web Application (PWA). This was done to ensure that it would be accessible across multiple devices, including desktops and smartphones without the need for separate app installation. The development of the application was through Visual Studio code, which was specifically chosen for its comprehensive support for technologies and Python. Our initial choice of language was Python due to its extensive library, simplicity, and reliability as well. The application was handled using Flask, which is a lightweight WSGI (Web Server Gateway Interface) web application framework that allows for quick development and easy integration with front-end technologies. The web app was built using HTML, CSS, and JavaScript for the front-end portion of the application, as HTML was used to build the structure of the web app, CSS for styling, and JavaScript for interactivity. These languages were integral in shaping and creating this application as they ensured a responsive and user-friendly interface. Moreover, Flask was used to build the back end of the application which was simply chosen for its simplicity as well as flexibility, allowing us for easy routing and handling and database integration.

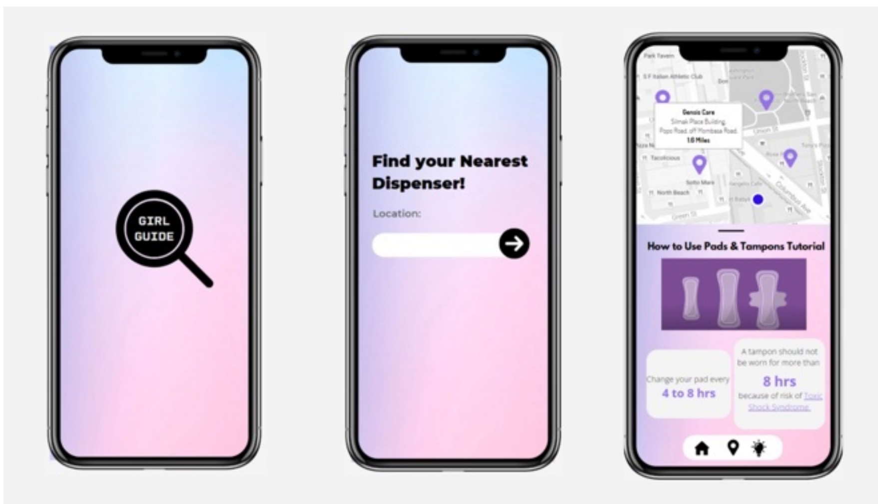


Fig. 3 Visualization of Web App Interface

3.3 Map Feature

We generated the first feature which tackled the main issue of providing sufficient access to sanitary dispensers. This was by integrating a "location" feature that would allow users to input their current location into the app, which then displays a map highlighting the closest pad dispensers available. This map also dynamically loads and refreshes; after it is opened in any web browser, users can zoom the display — to see urban streets or rural landscapes at different levels of granularity (and can experience bonuses from a light-hearted design). We recognized that not all dispensers would be captured due to the limitation of dispenser information and location in Kenya. For this reason, we included a user contribution function so users can add new locations if they find a dispenser that isn't listed. That way, the map is always updating and accurate because of a larger community (or crowd) providing data (see Figure 3).

3.4 Education Sub-Feature

The second most important feature was the educational aspect of this web application which we tackled by providing MH education-related content from globally recognized organizations like the World Health Organization (WHO) and UNICEF. This ensured that all information was true, reliable, and credible. In addition, we integrated sub-sub-features in the app, allowing users to receive advice on staying healthy during menstruation — tips include hygiene habits, how to handle menstrual discomfort, and understanding their periods. This also aids in raising awareness towards MH and wellness by addressing the psychological aspect which is so important in promoting a healthy and comfortable lifestyle. It promotes the perception of periods as a normal, healthy bodily function which helps in minimizing stigmatization and embarrassment. The educational content includes quizzes, articles, and infographics to help users remember key points (see Figure 3).

4 Results

As of right now, our application has successfully integrated both the "map" and "education" features. We will begin testing it using clients from Genesis Care, an organization with whom we have maintained communication since the project's inception. Although there are many more features we would like to incorporate into the application, our current focus is on thoroughly refining and optimizing the existing functionalities as well as creating a healthy balance between user experience and application performance before expanding the feature set.

5 Conclusion

The menstrual health crisis in Kenya has been at the forefront of this project. Through the integration of our PWA, we aim to break barriers and implement these beneficiary resources to young girls and women, both in the workplace and in school. Features such as the "map" and "education" components are crucial and will serve as a gateway to promoting MH education across Kenya. The upcoming testing phase with Genesis Care will provide valuable insights and feedback which will be instrumental in fine-tuning the app's performance and user experience. Ultimately this project is a step toward bridging the gap in MH resources contributing to a healthier more informed, and supported community.

5.1 Concerns and Next Steps

Although our application does not rely on or utilize any AI, there's a possibility in the future that this functionality could be integrated into our application to aid in optimizing, dispenser locations, recommendations, or analyzing usage patterns as well; however, no AI models were implemented in this current version. Our application will continue to undergo extensive testing across multiple devices and browsers to control compatibility and responsiveness. Although we are currently still working on the application we have been able to integrate many features as well as work on the existing features to make sure they're the best that it can be.

Acknowledgements Dr. Teuscher; Genesis team: Katherin Wanjoya & Mercy Musyoka

References

1. Austrian, Karen, et al. "Effects of Sanitary Pad Distribution and Reproductive Health Education on Upper Primary School Attendance and Reproductive Health Knowledge and Attitudes in Kenya: A Cluster Randomized Controlled Trial - Reproductive Health." *SpringerLink*, BioMed Central, 31 Aug. 2021, link.springer.com/article/10.1186/s12978-021-01223-7.

2. Phillips-Howard, Penelope A, et al. "Menstrual Cups and Sanitary Pads to Reduce School Attrition, and Sexually Transmitted and Reproductive Tract Infections: A Cluster Randomised Controlled Feasibility Study in Rural Western Kenya." *BMJ Open*, British Medical Journal Publishing Group, 1 Nov. 2016, bmjopen.bmj.com/content/6/11/e013229.short.

3. Alexander, Kelly T., et al. "Water, Sanitation and Hygiene Conditions in Kenyan Rural Schools: Are Schools Meeting the Needs of Menstruating Girls?" *MDPI*, Multidisciplinary Digital Publishing Institute, 22 May 2014, www.mdpi.com/2073-4441/6/5/1453.

4. Krishna, Aditi, et al. "Workplace Menstrual Health in the Private Sector: Results from a Pilot Study in Kenya and Nepal." *IWA Publishing*, 1 July 2024, iwaponline.com/washdev.

5. Kjellén, Marianne, et al. "THE INTERACTIONS BETWEEN SANITATION SYSTEMS AND MENSTRUAL MANAGEMENT." *Stockholm Environment Institute*, 1 Jan. 2011, http://www.jstor.com/stable/resrep00484.5.

6. Progress on Drinking Water, Sanitation and Hygiene in Schools 2015–2023: Special focus on menstrual health. New York: United Nations Children's Fund (UNICEF) and World Health Organization (WHO), 2024.

Green Roof: Monitoring Soil Moisture, Plant Health, and Stormwater Management

Ejay Aguirre and Cloie Seo

Abstract This study investigated soil quality to identify the most suitable area within the Commonwealth of the Northern Marianas Islands (CNMI) for agricultural purposes and to seek the potential to integrate the findings into a green roof initiative. The primary objective is to assess the growth medium samples' soil quality index collected from four residential villages. Recognizing the benefits of pH, moisture retention, detention rate, nutritious soil, composition, and high evapotranspiration (ET) levels, these factors were measured to assess whether the soil exhibited optimal characteristics. The pH levels were measured for each location sample to determine the acidity or alkalinity of the soil. Moisture retention, the ability to retain water in the soil, was determined by the gravimetric method, which was also used to measure ET. Detention rates—the rate at which water drains from the soil—were measured. The growth rates of mung beans (*Vigna radiata*), which are incredibly environment-adaptable, were used to gauge the nutritional status of the soil. For ET, two models were designed to distinguish between evapotranspiration and evaporation; the Penman-Monteith equation was also used to estimate ET based on meteorological data. The results showcase three different samples with similar qualities for further investigation of green roof initiatives.

Key words: soil quality index, green roof, evapotranspiration, detention, meteorological

---

Ejay Aguirre
Northern Marianas College, Saipan e-mail: ejay.aguirre03@gmail.com

Cloie Seo
Northern Marianas College, Saipan e-mail: cloieseo0343@gmail.com

1 Introduction

The authors of this paper are residents of Saipan, in the Commonwealth of the Marianas Islands. We wanted to engage in a project that would be beneficial for our community and would be able to solve environmental issues specifically. We believed that it would open the potential for cultural connection through horticulture, boost our tourism industry with its green scenery, cool down residential houses from the global heat island effect, and manage storm water more safely. However, with many issues to tackle, we decided to focus on soil quality and storm management as we determined it would be the first step to implementing green roof initiatives by conducting experiments with local soil. Therefore, our project design focuses on different soil samples' quality indices and their evapotranspiration levels for storm management.

2 Related Work and Background

The implementation of a green roof can be considered an interconnection between human construction and nature. Its emergence began during the 20th century, when sand and gravel became a replacement for simple drainage systems, leading to Germany becoming the first country to implement the principle of green roofs [6]. Many other countries followed the pursuit of integrating construction and nature, which have become a big part of attraction sites such as the roofs at Villa Mairea and the green roofs at the Monastery of La Tourette [1].

Apart from being used as an attraction for visitors, green roofs provide an engineered roofing system for residents where they cover the rooftop with growth medium, vegetation, and many other layers, providing environmental solutions depending on climate and construction circumstances [1]. he layers of the system vary depending on the environment, but they usually include a growth medium layer, vegetation layer, filter layer, water storage and drainage layer, anti-root barrier, waterproofing membrane, and protection layer [1]. It can be distinguished into three systems: extensive, semi-intensive, and intensive. The difference resides in the types of vegetation, whether it will only be grass, tall grass, or even trees correspondingly [1]. The choice of system would depend on budget affairs and environmental conditions; nonetheless, green roofs produce benefits in many areas, such as sustainability, energy conservation, acid rain, air and noise pollution, and urban heat [3].In addition, it addresses social and environmental concerns by enhancing the ornamental scenery to improve quality of life. For instance, Harris & Trauth [5] conducted a report stating that horticulture provides significant mental health benefits by reducing stress, anxiety, and depression while enhancing self-identity. Moreover, horticultural activities foster community engagement, hence offering a great therapeutic method for promoting environmental awareness and sustainability. Lastly, ornamental gardens, and green spaces—the potential that green roofs can offer—provide a great space for these implementations while bringing an appealing quality of environment that contributes positively to air quality and can be used for recreation [5].

The article was based on Saipan, an island in the Commonwealth of the Northern Marianas Islands (CNMI), home to the Chamoru and the Refaluwasch. Dating back to pre-colonial times, residents were known for agricultural techniques that included the cultivation of crops that were used for consumption and even medicine [9]. These practices would later be upscaled under the Spanish, German, and Japanese colonial periods, where they were engaged in cultivation. Despite the different agricultural techniques, it prevailed to be an important part of their culture in both pre-colonial and colonial periods [9]. In modern times, the CNMI faces conflicts over utilizing land for commercial purposes rather than the conservation of agricultural and residential usage [14] . As a consequence of global heat increasing, forest fires initiate the potential loss of wildlife and biodiversity (e.g., local plants used for medicine), ultimately leading to the degradation of soil fertility and erosion [14].Public responsibility is given to residents to maintain the ornamentation of stable agriculture in the CNMI; therefore, the innovation of green roofs proves an essential and effective method that brings benefits to residents and the land. Benefits that will be focused on will be storm-water management, quality index, and thermal reduction. The term evapotranspiration (ET) refers to the process by which water is transferred from the land to the atmosphere through evaporation from the soil and other surfaces and transpiration from plants [3]. This process is rather critical, as most of the benefits of green roofs are related to this phenomenon [3]. For instance, for the benefit of temperature regulation, green roofs cool the surrounding environment through ET, which can reduce the consumption of air conditioning in buildings [6]. Moreover, ET significantly reduces stormwater runoff from rooftops, which decreases the burden on urban drainage systems and minimizes the risk of flooding [6].

That being said, soil quality is a fundamental aspect of the ecosystem, supporting plant and animal productivity and maintaining water and air quality. Maintaining soil quality is key to ensuring environmental stability and the overall maintenance of the biosphere [17]. In addition to its agricultural significance, soil quality is essential for environmental sustainability because of its vital role in ET. Soil functions as a natural filter by purifying water and is crucial in the global carbon cycle through its role in sequestering organic carbon. Grassland ecosystems, which cover about 40% of the Earth's surface, store over 34% of the global terrestrial ecosystem carbon within both the soil (up to 1-meter depth) and vegetation [16]. This makes grasslands the second-largest carbon storage system, following forest ecosystems. However, recent human activity such as intensive farming, deforestation, urbanization, and pollution has greatly impacted soil quality [12, 13]. Extensive stressing of soil has "destroyed the soil's physical and chemical properties and relationships between soil and plants, causing ecological and environmental problems" [15].

The soil quality index (SQI) is a measure that allows for a quantitative and accurate assessment of soil quality, which can be assessed for both agroecosystems and natural ecosystems. The SQI has no exact method for measuring, it relies on evaluations of the physical, chemical, and biological properties of soil [4, 7, 8]. For example, one study reported that topographic features and land use control soil properties on different scales [2, 15]. Turgut and Güler also found that SQI values varied significantly according to classes of the normalized difference vegetation index (NDVI), which offers a better explanation for the relationship between vegetation density and soil quality (2023).

Given the existing literature on soil quality and importance, the current study utilized various approaches to examine which area within the Commonwealth of the Northern Marianas Islands (CNMI) would be most fit for our agricultural needs. The purpose of the current study was to determine the most promising area for agricultural practices in hopes of identifying the potential to integrate it into the green roof initiative. Mindful of the benefits previously listed, it was predicted that the soil that showcases the features of moisture retention, lightweight, and high levels of ET would be the optimal soil for a green roof in Saipan. Therefore, by measuring the soil's daily ET levels, retention and detention, SQI, and composition of the collected growth medium samples, the authors of the paper focused on the significance of choosing growth mediums to optimize the engineering roof system depending on the circumstances of the building and climate.

3 Methodologies and Data

As mentioned previously, the experiment took place in Saipan on the CNMI. Growth medium samples were collected in four different residential villages in the north, south, east, and west. Given the size of the area, the collection of soil was conducted using a random sampling method. A grid was placed onto an image of the chosen area and was numbered (see *Fig. 1*). Then, a random number was generated to match a square, which would be the area where the soil would be collected. The chosen square was based on the first selection in a given green area; if the area was unlikely to have soil, a new randomly generated number would proceed again until a green space was chosen. Once in the area, soil patches were the main priority for collecting soil.

3.1 Methods

Once the samples were collected, experiments were conducted to analyze the soil's quality, composition, ET, and detention. The following experiments can be categorized into two categories: soil quality and green roof.

Note. All the data collected was done by the authors prior to and during the experiment.

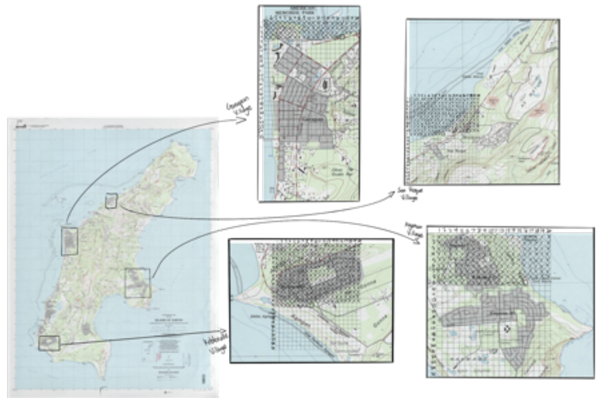


Fig. 1 A visual image of the random sampling method for collecting growth medium samples

3.1.1 Soil Quality Data Collection

To assess soil quality, four key parameters will be measured: pH, growth rate, and moisture content, and detention rate.

pH Level. For pH measurement, 10 grams of soil sample will be placed in a glass beaker, ensuring the removal of any sticks, stones, or other debris. The beaker will be filled with 20 mL of distilled water and the mixture will be vigorously agitated. After allowing the mixture to rest for 30 minutes, the sample will be drained using a coffee filter. A pH strip will be used to determine the pH level of the sample, and this procedure will be repeated for three trials with each soil sample.

Growth Rate. To measure the growth rate, seedling pots will be labeled as soil samples I, II, and III, respectively. Each pot will be filled with the corresponding soil sample, and seeds will be planted at a depth of 1 ½ inches. The pots will be thoroughly watered, and plant growth measurements will be collected daily. Weather and temperature changes, if any, will also be recorded. The vegetation of choice was the mung bean (*Vigna radiata*). This is due to several reasons, largely due to their adaptability. According to Pataczek et al., [10], mung beans are significantly adaptable to various environmental stressors, most especially drought and salinity. Considering the tropical climate in the experiment setting, mung beans make a viable option for studying plant-water interactions.

Moisture Retention. To determine moisture content, the Gravimetric Method was used. Soil samples were brought to near saturation and 50 grams of the saturated soil were weighed in a clean container, then dried using a microwave in short intervals of 1 to 2 minutes. The weight of the dry soil was collected, and the moisture content will be calculated using the appropriate formula.

$$MC(\%) = \frac{(wet\_weight - dry\_weight)}{dry\_weight} \times 100$$

Soil Composition. Three soil separation tubes were placed in a rack and labeled A, B, and C. Tube A was filled with the soil sample up to the line marked 15. The bottom of the tube was gently tapped on a firm surface to compact the soil and eliminate air pockets. Using a plastic pipette, 1 mL of the Texture Dispersing Reagent was added to Tube A, then diluted with water up to the line marked 45. The tube was capped and gently inverted for two minutes to ensure thorough mixing. Tube A was then placed back in the rack and allowed to settle for 30 seconds. The supernatant was carefully poured into Tube B, and Tube A was returned to the rack. Tube B was left to stand undisturbed for 30 minutes, after which the solution was carefully decanted into Tube C, and Tube B was returned to the rack. 1 mL of the Soil Flocculation Reagent was added to Tube C, which was then capped and gently inverted for one minute. Tube C was placed back in the rack and allowed to stand until the end of the lab period. The levels in each tube were measured, and the percentages of sand, silt, and clay were calculated using the formulas:

$$\%_{sand} = \frac{tube\_A\_level}{15mL} \times 100$$
$$\%_{slit} = \frac{tube\_B\_level}{15mL} \times 100$$
$$\%_{clay} = \frac{tube\_C\_level}{15mL} \times 100$$

Finally, the soil type was determined using the Soil Textural Triangle (see *Fig.* 2).

Detention. The purpose of the water detention experiment is to measure the amount of water retained by different soil samples and the rate at which water drains from them. This involves using the exiting containers with different soil samples and ensuring each container has drainage holes. In our case, we punctured 15 holes evenly to allow excess water to flow out. Then, to ensure unsaturated growth medium is used, a known volume of water, 500 mL, will be poured onto each soil sample to simulate rainfall events. Next, a container will collect the water that drains out of each container, and the measurement of the water will be recorded at regular intervals for every 5 minutes for a total of 30 minutes. Lastly, we are able to determine the runoff coefficient with the following formula:

$$C = \frac{V_r}{V_i}$$

where C is the runoff coefficient, V_r (mL) is the volume of runoff collected, and V_i (mL) is the volume of water initially added to simulate rainfall.

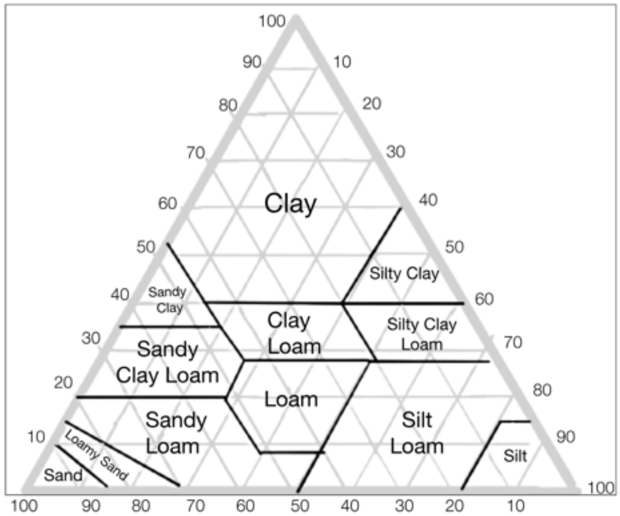


Fig. 2 Sandy soil provides few nutrients and lower retention rates, providing a dry condition for plants. Silty soil, having a size between sand and clay, provides more nutrients and retention rates than sand. Lastly, clay soil, holds the nutrients and water providing a wet condition for plants.

3.1.2 Green Roof Data Collection

Two methods will be used to measure evapotranspiration. Prior to the start of the experiment, green roofs were built to replicate a house building. 5 containers were used to replicate the house's walls, and the container lid was fixed to be a roof (see *Fig.* 3). The building height measures 19 cm, length 26 cm, and width 11.5 cm. The roof measures 16 cm by 22 cm for each of the 5 containers where it will contain planted mung beans. Another separate model was created to the size of 20 cm by 32 cm with just the soil (see *Fig.* 4). Two separate models were created to ensure a controlled environment to distinguish between evapotranspiration and evaporation.

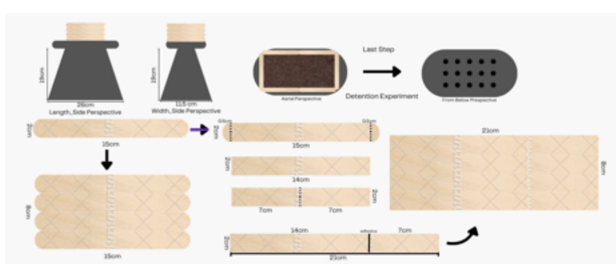


Fig. 3 A visual framework for how the green roof model was made using the materials.

Fig. 4 A photo of the models for measuring ET (left) and EV (right).

Gravimetric Method. The first measuring method includes the use of the gravimetric method. This method is a direct measurement of the change in weight of a soil-plant system before and after a certain period.

In the case of the paper, 14 days was the designated time frame for data collection. At the start of each day, the initial weight of the container lid (roof) is recorded. As for the model without vegetation, a cup that holds (fixed volume) scoops up the soil for measuring. Then, 500 mL was added to the containers each day . Lastly, at the end of the day, the container roof and a cup of soil were measured to determine the final weight. Once these variables are collected, the ET and E can be calculated with the following formula:

$$ET/E = \frac{(initial\_weight + water\_added) - final\_weight}{Area}$$

The difference in weight represents the amount of water lost due to evapotranspiration [11]. Furthermore, the final result from the calculation was multiplied by 10 to convert the units to mm/day to ensure uniform units in both methods.

Penman-Monteith Equation. The next measuring method is using the Penman-Monteith equation. This method is used to estimate ET based on meteorological data, where it combines factors such as solar radiation, temperature, humidity, and wind speed to calculate ET [11]. It will only be done with the models with vegetation, as it calculates ET. The Penman-Monteith equation can be seen as follows:

$$ET_{PM} = \frac{0.408 \times \Delta \times (R_n - G) + \gamma \times \frac{900}{T+273} \times u_2 \times (e_s - e_a)}{\Delta + \gamma \times (1 + 0.34 \times u_2)}$$

where Δ is the saturation vapor pressure $(kPa/°C)$ curve at air temperature, R_n is the net solar radiation $(MJ/m^2/day)$ at the crop surface a fixed value ($R_n = 12MJ/m^2/day$)was used due to absence of solar radiation data, G is the soil heat flux density $(MJ/m^2/day)$, γ $(kPa/°C)$ is psychometric constant, T $(°C)$ is the mean air temperature, u_2 (m/s) is wind speed, a fixed value ($u_2 = 11.5m/s$) was used again due to the absence of wind speed data, e_s (kPa) is saturation vapor pressure, and e_a (kPA) is actual vapor pressure [18].

3.2 Data Collection

Our initial phase for collecting data involved organizing the collected experimental data in Microsoft Excel. The authors meticulously recorded parameters such as soil type (5 soil type samples), initial weight, final weight, water added, and surface area (ET Roof Area = $32cm^2$, E Roof Area= $640cm^2$) ensuring consistency in units across all entries. Later, our tables were converted into comma separated values (csv) files.

3.2.1 Data Import, Cleaning, and Processing with Python

To perform advanced data analysis, the authors utilized Python's data manipulation library, 'pandas' and numerical computation library, 'numpy' to gain descriptive insight on the imported csv file. Moreover, Python was used to calculate formula equations involving gravimetric method, Penman-Monteith Equation, and the runoff coefficient.

3.2.2 Data Visualization

Visual representations of the data were created using 'matplotlib', which enables the authors to facilitate the identifications of trends and patterns. This comprehensive approach allowed us to derive meaningful insights from our experimental data, facilitating a deeper understanding of evapotranspiration and water retention processes in green roof systems.

4 Results

At the end of the data collection, a total of 28 samples were collected to measure plant growth during the span of 12 days for each of the growth medium samples. Moreover, 12 samples were collected to measuring ET for each of the growth medium samples; in addition 12 weather samples were collected during the of 12 days. Each of the samples showcases a pH level of 7 with the exception of Pot Mix. Similarly, the soil samples collected around the community have almost the same percentage levels with the exception of the Pot Mix having the highest moisture retention percentage at 2.70%. Followed after the highest is South Roque at 0.49%, Koblerville at 0.45%, Kagman at 0.41%, and the lowest being 0.35% from Garapan. In terms to the growth medium samples' runoff coefficient (C), San Roque had the highest C, followed by Koblerville, Kagman, Garapan, and lastly Pot Mix having no moisture drainage (see *Table* 1). It is interesting to note a similar percentage of sand from the community collected growth medium samples (see *Table* 2. Moreover, the gaps of percentages for both sand and clit is not are not too far from each other.

	pH	Moisture Retention (%)	Mean Growth Rate	Runoff Coefficient
San Roque (SE)	7	0.49	0.53	0.24
Kagman (KN)	7	0.41	0.63	0.04
Kobler-ville (KE)	7	0.45	0.73	0.14
Garapan (GN)	7	0.35	0.53	0.1
Pot Mix (Control)	6	2.70	0.63	0

Table 1 Descriptive Statistics for Soil Quality Index per sample type.

	Sand (%)	Silt (%)	Clay (%)	Excess Debris (%)
San Roque (SE)	90	6.7	1.7	1.6
Kagman (KN)	93.3	3.3	3.3	0.1
Kobler-ville (KE)	93.3	5	1.7	0
Garapan (GN)	93.3	4.3	2.3	0.1
Pot Mix (Control)	30	5	0.98	64.02

Table 2 Soil Composition assessment results.

Plant Growth Correlation. Correlation tests were conducted to determine if there was a statistically significant relationship between pH and growth rates, pH and moisture retention, and moisture retention and growth rates—the analysis aimed to determine a statistically significant relationship between the examined variables. Using the correlation test, it was found that there was a statistically significant relationship between pH and growth rates. As seen in *Figure* 5, findings revealed a gradually increasing trend where lower pH levels had greater growth rates, $r(9) = -0.012, p < 0.01$.

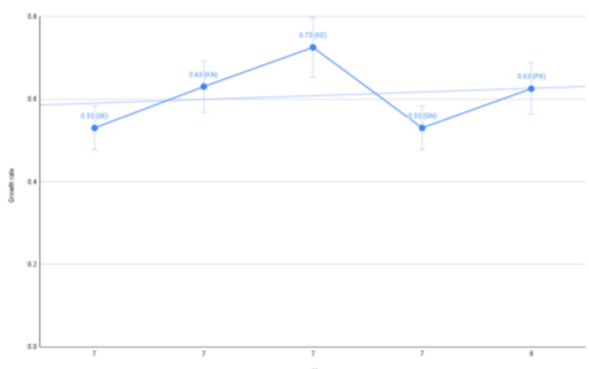


Fig. 5 Scatter plot illustrating the relationship between Moisture Retention and Growth Rate. Data points represent observed growth rates at different levels of moisture retention. The trendline indicates a general positive relationship, though variability is noted.

Between pH and moisture retention percentages, a statistically significant relationship was determined. As shown below in *Figure* 6, findings revealed an increasing trend where lower pH had higher moisture retention percentages, $r(9) = -1.00, p <$

0.01. In addition, the correlation test determined that there was a statistically signif-

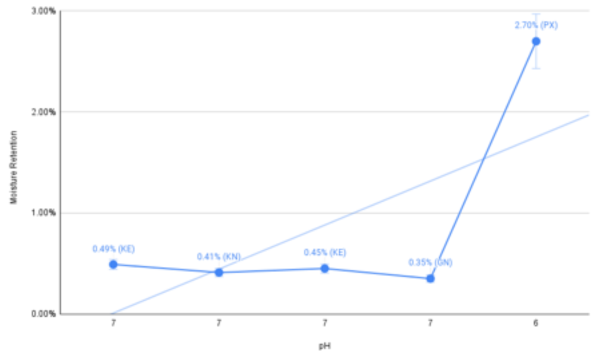


Fig. 6 Scatter plot showing the relationship between pH levels and Growth Rate. Each data point represents the growth rate observed at a specific pH level. The trendline suggests a potential correlation, with slight variations in growth rate across different pH levels.

icant relationship between growth rates and moisture retention. As shown in *Figure* 7, findings revealed a marginal trend where higher moisture retention percentages had higher growth rates, $r(9) = 0.13, p < 0.01$.

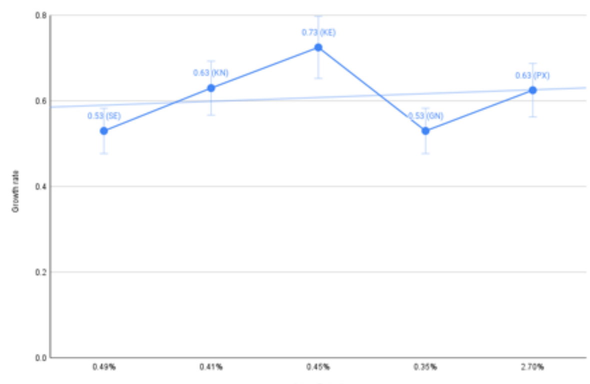


Fig. 7 Scatter plot displaying the relationship between pH levels and Moisture Retention. The data points highlight the moisture retention at various pH levels, with the trendline indicating a strong positive relationship, especially at lower pH levels.

Evapotranspiration, Detention, and Evaporation. Table 2 showcases descriptive statistics, which contain the means and standard deviation of each of the samples used in the experiments using the gravimetric method. Notice to the E data, each sample presents similar numbers, hence, analysis will be focused on ET evaluation.

Growth Medium Sample	Mean ET	Standard Deviation ET	Mean E	Standard Deviation E
Garapan (GN)	8.131667	4.945286	4.833333	0.389249
Koblervile (KE)	9.028333	5.125815	5.000000	0.000000
Kagman (KN)	8.283333	3.299139	4.833333	0.389249
San Roque (SE)	7.497500	6.800141	4.916667	0.288675
Store Pot Mix (PX)	7.645000	3.276910	4.916667	0.288675

Table 3 Samples was weighed, measured, and calculated for there ET and E (n= 12 per sample).

There is no significant linear relationship between the Runoff Coefficient and the Gravimetric ET average ($r(58) = -0.058, p > 0.05$). The weak correlation and high p-value indicate that these two variables do not strongly influence each other in a linear fashion (see *Fig.* 8).

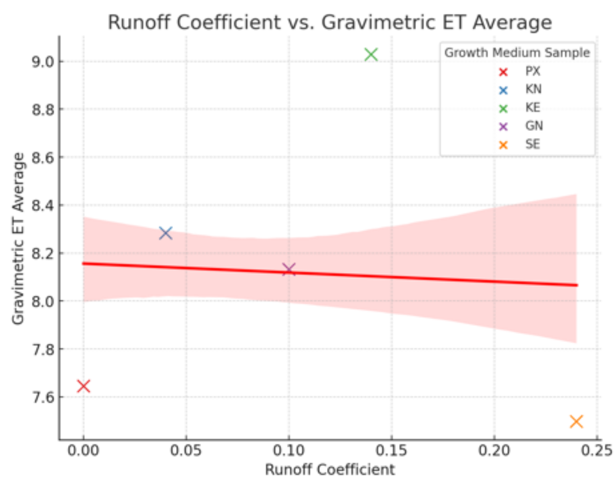


Fig. 8 Scatter plot showing the relationship between Runoff Coefficient and Gravimetric ET Levels Average across different Growth Medium Samples.

With the following methods of evaluating ET levels for soil, a t-test was conducted to measure the ET levels with the gravemtric and Penman-Monteith models per different growth medium samples. KN, $t(23) = -7.676, p > 0.05$, and PX, $t(23) = -8.848, p > 0.05$, did not showcase any significance in ET levels between the two models. On the contrary, GN, ($t(23) = -5.683, p < 0.001$; KE, $t(23) = -4.655, p < 0.001$; and SE, $t(23) = -4.417, p > 0.001$) showcase a very significant difference between the gravimetric and Penman-Monteith ET values (see *Fig.* 9).

Furthermore, an analyzation was conducted to see which of the community growth medium sample was best fit to replicate the store bought pot mix. In result, none of

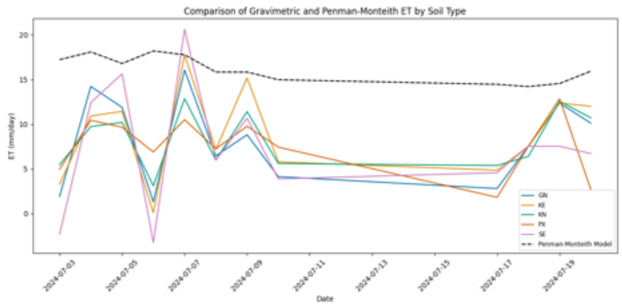


Fig. 9 Comparison of Gravimetric ET (colored lines) and Penman-Monteith ET (black dashed line) across different soil types over a range of dates. Each line represents the evapotranspiration (ET) values for a specific soil type, showing daily variations and the performance of each soil type relative to the Penman-Monteith model.

the growth medium samples (KN, $t(23) = 0.475, p > 0.05$; KE, ($t(23) = 0.788, p > 0.05$; and SE ($t(23) = -0.068, p > 0.05$)) show statistically significant differences in ET compared to the control soil (see *Fig.* 10). 56

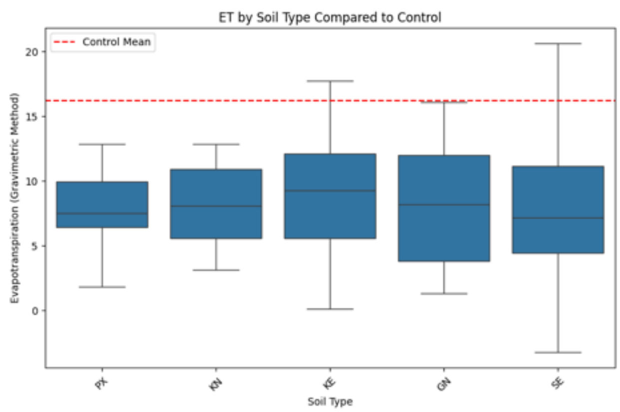


Fig. 10 Box plot showing the distribution of Gravimetric ET values across different soil types compared to the control mean (Pot Mix). The box plots illustrate the median, interquartile range, and the variability of ET values for each soil type, highlighting differences in evapotranspiration efficiency.

Moreover, the results show that there are interesting differences of ET levels between the two methods of measuring ET with three of the samples showing significance differences. In addition, there was no significant difference in ET levels between the community growth medium samples and store bought pot mix.

5 Conclusion

The goal of the study was to create a start to implementing green roof systems for Saipan located in the CNMI. The authors investigated the growth medium's composition and quality index for each sample. The authors also further investigated their ET level per sample, as it plays a crucial role in a green roof system. Proceeding to the samples' soil composition, it is keen to note that soil texture influences water management and plant health. Moreover, the soil quality index (SQI) has no exact method for measuring, as it relies on evaluations of the properties of the soil; therefore, the authors had a control sample. However, the control sample had insignificant correlations with each of the collected samples.

Soil Composition & Quality Index. All growth medium samples collected around the community had a high percentage of sand reaching 90%. In terms of their silt and clay percentage all collected soil was rather low; hence providing a dry condition for plants and having a few nutrients. They also each had similar growth rates with only approximately a 0.10 difference. Furthermore, each of the growth medium samples had a low runoff coefficient with the highest being San Roque, 0.24, and the lowest being Garapan, 0.1. The control sample was the store pot mix because it is a soil substitute for plants in pots. Although the pot mixes are pre-fertilized, they are made to have good structure and texture with high-quality air porosity. Pot mixes are meant for plot plants whereas soil is only for large volumes of plant roots.

Evapotranspiration. A weak correlation suggests that the Runoff Coefficient and Gravimetric ET average are largely independent of each other. This implies that the factors influencing the runoff in soil are not directly linked to the factors that determine the ET as measured by the gravimetric method. Overall findings suggest that Garapan, Koblerville, and San Roque all have a very high statistical significance. Garapan has the largest difference relative to the variability, with approximately 1.0 difference from Koblerville and approximately 1.2 difference from San Roque in terms of the values from both the Gravemetric ET and Penman-Monteith ET methods. The consistency of negative t-statistics implies that the Gravemetric method yields lower ET values across all the growth medium samples compared to the Penman-Monteith equation. Furthermore, none of the collected samples was able to replicate the pot mix's ET data due to its air porosity and combination of inorganic material.

Understanding the baseline for the soil health in various areas (specific to Saipan) is an overlooked subject as it is essential for monitoring changes over time. Hence, showcasing the importance of making informed decisions for agricultural practices can be done by ensuring a baseline of soil quality. Moreover, farmers and gardeners could optimize their practice to improve crop sustainability, as it offers a guide to a correct selection of appropriate crops to be enhanced based on the use of the soil's quality. Nonetheless, these findings could influence the design and implementation of green roof infrastructure projects, by providing data on which soils best support vegetation while offering a managed storm-water system. This also includes the methods and models used for ET estimation, in particular, assessing empirical methods compared to complex models. Since accurate ET estimation is critical for effective water resource management, a feature Saipan currently lacks, it implies the importance of refining ET estimates. This research helps in the precise scheduling of irrigation to ensure crops receive the optimal amount of water, reducing waste, and conserving resources.

5.1 Applications

Future research could lead to the development of innovative soil amendments tailored to specific soil quality index outcomes. This could lead to the enhancement of soil performance for a particular crop or to be adaptable to environmental conditions. Moreover, understanding the differences between the empirical and complex models can help refine methods used for ET estimation in future research. By particularly assessing the accuracy by the use of remote sensing technology and Geographic Information Systems (GIS) it can be used to integrate data collected to create a large-scale and accurate soil quality map in regional management. All in all, further research can refine the soil quality index framework to include additional parameters, such as organic carbon content, inorganic material, and microbial activity, to provide a comprehensive understanding of soil health. In addition, these insights can guide adjustments in green roof designs and irrigation strategies to ensure that water management practices account for the discrepancies between different ET estimations.

5.2 Concerns and Next Steps

All in all, a great deal of concern lies in the fact that the experiment was short-term, only consisting of 12 days in total. This heavily affects the presented data since ET levels and plant growth can vary when experiencing different types of weather conditions. Since Saipan has two seasons, wet and dry, the authors suggest designing a long-term experiment that could cover two to three months for each season. At the same time, it could be designed to assess accurate data with the use of remote sensing technology. In addition, a variety of vegetation species should be used for the soil samples used to ensure optimal soil performance and ET level. Not only will it enhance the soil performance, soil, and different materials should be assessed to create an optimal retention and detention system to create a storm-water management system. This is so due to the flat roofs and lack of system Saipan has to offer and to tackle climate change issues such as global warming, it could lead to a safety protocol to ensure less floods and temperature absorption. With that, the authors conclude that Koblerville, Garapan, and San Roque should be implemented in future studies.

Acknowledgements The work was supported by the altREU program at Portland State University. Northern Marianas College's Science, Math, Health, and Athletic department for providing chemical and physical materials for conducting soil compositions. Creating Opportunities for the Marianas: Pathway for Access and Student Success (COMPASS) for overlooking our project design.

References

1. F. Abass, L. H. Ismail, I. A. Wahab, and A. A. Elgadi. A review of green roof: Definition, history, evolution and functions. *IOP Conference Series: Materials Science and Engineering*, 713(1):012048, 2019.
2. E. Alidoust, M. Afyuni, M. A. Hajabbasi, and M. R. Mosaddeghi. Soil carbon sequestration potential as affected by soil physical and climatic factors under different land uses in a semiarid region. *CATENA*, 171:62–71, 2018.
3. S. Cascone. Green roof design: State of the art on technology and materials. *Sustainability*, 11(11):3020, 2019.
4. G. Gozukara, M. Acar, E. Ozlu, O. Dengiz, A. E. Hartemink, and Y. Zhang. A soil quality index using vis-nir and pxrf spectra of a soil profile. *CATENA*, 211:105954, 2022.
5. K. Harris and J. Trauth. International journal of current science and multidisciplinary research. *International Journal of Current Science and Multidisciplinary Research*, 3(4):61–65, 2020.
6. W. C. Li and K. K. A. Yeung. International journal of current science and multidisciplinary research. *International Journal of Sustainable Built Environment*, 3(1):127–134, 2014.
7. L. F. Marion, R. Schneider, M. R. Cherubin, G. S. Colares, P. G. Wiesel, A. B. da Costa, and E. A. Lobo. Development of a soil quality index to evaluate agricultural cropping systems in southern brazil. *Soil and Tillage Research*, 218:105293, 2022.
8. M. Mazzon, L. Cavani, C. Ciavatta, G. Campanelli, G. Burgio, and C. Marzadori. Conventional versus organic management: Application of simple and complex indexes to assess soil quality. *Agriculture, Ecosystems & Environment*, 322:107673, 2021.
9. J. Mushynsky, J. McKinnon, G. Cabrera, and H. Tudela. The foundation of island identity: Chamorro maritime practices and resistance in the commonwealth of the northern mariana islands. *AlterNative: An International Journal of Indigenous Peoples*, 18(2):299–312, 2022.
10. L. Pataczek, Z. A. Zahir, M. Ahmad, S. Rani, R. Nair, R. Schafleitner, G. Cadisch, and T. Hilger. Beans with benefits—the role of mungbean (vigna radiata) in a changing environment. *American Journal of Plant Sciences*, 9(7):1577–1600, 2018.
11. N. Romano. Soil moisture at local scale: Measurements and simulations. *Journal of Hydrology*, 516:6–20, 2014.
12. R. Sanji, Y. Kooch, and A. Rey. Impact of forest degradation and reforestation with alnus and quercus species on soil quality and function in northern iran. *Ecological Indicators*, 112:106132, 2020.
13. K. Taghipour, M. Heydari, Y. Kooch, H. Fathizad, B. Heung, and R. Taghizadeh-Mehrjardi. Assessing changes in soil quality between protected and degraded forests using digital soil mapping for semiarid oak forests, iran. *CATENA*, 213:106204, 2022.
14. A.P. Tudela. Agroforestry programs and issues in the northern marianas islands. In *Proceedings of the Workshop on Research Methodologies and Applications for Pacific Island Agroforestry*, 1993.
15. B. Turgut and S. Güler. Assessment of soil quality index for different ndvi ranges in a watershed. *Turkish Journal of Agriculture and Forestry*, 47(6):872–893, 2023.
16. R. P. White, S. Murray, M. Rohweder, and S. D. Prince. *Pilot analysis of global ecosystems: Grassland Ecosystems*. World Resources Institute, 2000.
17. X. Zhao, M. Tong, Y. He, X. Han, and L. Wang. A comprehensive, locally adapted soil quality indexing under different land uses in a typical watershed of the eastern qinghai-tibet plateau. *Ecological Indicators*, 125:107445, 2021.
18. Lincoln Zotarelli, Michael D. Dukes, Consuelo C. Romero, Kati W. Migliaccio, and Kelly T. Morgan. Step by step calculation of the penman-monteith evapotranspiration (fao-56 method) 1. 2015.

Development and Evaluation of a Cyberbullying Detection AI Model

Eleanor Isles

Abstract With the advancement of technology and the increasing time society spends online, cyberbullying has become a worldwide phenomenon that brings an abundance of negativity and needs to be addressed. This paper presents research that aims to develop and evaluate a cyberbullying detection model. Utilizing a specialized binary dataset and logistic regression, this model was built with a goal to perform sentiment analysis, and determine whether a given statement is cyberbullying. Ultimately, the proposed model was able to predict cyberbullying within the test dataset with an 86 percent accuracy. This chapter goes over the process to make, properties, and capabilities of the constructed model, as well as comparisons to similar ones within the fields.

1 Introduction

The proliferation of digital communication platforms has altered the world in many ways. Instantaneous and far-reaching exchanges are enabled, and the horizons of human's perceptions have been expanded. However, this transformation has also given rise to significant challenges, including increasing concerns of cyberbullying.

Bullying has been around for a long time. There are six main categories of bullying: physical, verbal, social (also referred to as relational), sexual, or prejudicial, and cyber[2] [5]. Cyberbullying is new and different from other forms of bullying because perpetrators do not see the faces of their targets, and subsequently may not fully grasp the consequences of their actions. This decreases the feelings of personal accountability when it comes to their actions. In literature, this is commonly

Eleanor Isles
Portland State University, 1825 SW Brodway, Portland, OR 97201 e-mail: nora.isles@gmail.com

Christof Teuscher
Portland State University, 1825 SW Brodway, Portland, OR 97201 e-mail: teuscher@pdx.edu

referred to as the "disinhibition effect"[8]. The disinhibition effect does not occur in other forms of bullying to the same degree, because the other forms are not online. Although cyberbullying is a new form of bullying, it still needs to be taken just as seriously.

According to the Cyberbullying Research Center, around 30 percent of teens in the U.S. report having ever been cyberbullied, and 13 percent of teens have in the 30 days before the survey[7]. Furthermore, 15 percent of those surveyed have confessed to cyberbullying others[7]. Getting cyberbullied is also common among those younger. When it came to tweens (9-12 years old), the studies found that about 15 percent had been cyberbullied[7]. Although cyberbullying peaks in those ages 11-14, anyone can be a victim of it[4].

The prevalence and impact of cyberbullying continues to grow, necessitating effective measures for its detection and remediation. Cyberbullying's defining characteristic is that it takes place online. Therefore, there are tools we can use to identify and address it. Various approaches to detecting cyberbullying, such as manual moderation and keyword filtering, have been implemented [11]. However, due to the pervasive spread of cyberbullying, advanced automated methods have proven to be more effective at identifying instances[11]. Supervised machine learning algorithms can be trained to recognise patterns and features indicative of cyberbullying. The specific training model used in this paper was chosen for its accuracy compared to other similar options, as well as its simplicity and effectiveness when it comes to binary classification tasks. This model utilizes over 90,000 strings of data from texts to classify data as cyberbullying and not cyberbullying[3]. In this paper, the approach will be compared to others. The following sections of this chapter will dive into the related work and background, methodology, and results, providing a thorough examination of the model's development and evaluation. The approach presented in this paper aims to contribute to ongoing efforts on combating cyberbullying and fostering safe online environments.

2 Related Work and Background

There is a lot of research for detecting cyberbullying, and the field is rapidly growing. It is driven by the increasing prevalence of online harassment and the need for effective intervention methods. This section reviews some of the key related works.

The application of machine learning to the field of cyberbullying has been brought up in other papers and explored. Various techniques have emerged, each offering unique strengths and limitations. Logistic regression has been seen as a popular choice to write about and compare in these papers.

A significant contribution to this area of study is the research by Seif-Alamir Yousef and Ludvig Svenson (2024)[14]. The project provides a comprehensive overview of different supervised machine learning methods for classifying bullying. Their work highlights logistic regression as a viable option, bringing up its utility in tasks where clear, binary outcomes are needed. Their analysis shows that logistic regression, while basic compared to more complex models, offers a balance between interpretability and performance, making it suitable for real-world applications where transparency is a key requirement.

Logistic regression's prevalence in cyberbullying detection is further reinforced by a review conducted by Daniyar Sultan et al. (2024), which evaluates various machine learning techniques in the context of cyberbullying[13]. The paper notes that logistic regression remains a common benchmark model against which other algorithms are compared. Its consistent inclusion in these reviews underscores its importance in the toolkit of cyberbullying detection methods, particularly when the goal is to achieve reliable performance with straightforward implementation.

Additionally, a study by Andrea Perera and Pumudu Fernando (2021) delves into the complexities of cyberbullying detection on social media platforms[10]. Their research emphasizes the challenges posed by the nuanced and context-dependent nature of cyberbullying, which can vary significantly across different platforms and user interactions. They discuss how logistic regression and other models must be carefully tuned to account for these complexities, ensuring that the detection algorithms do not oversimplify the issue and can effectively identify subtle instances of cyberbullying.

3 Data and Methodologies

A large dataset meant for detecting cyberbullying was selected to train this model due to its comprehensive nature and the uniformity of its data format, as well as how the data was pulled from text messages - making it ideal for the purpose of the project. The data was preprocessed and vectorized before being trained through the Logistic Regression model to predict whether a given statement is cyberbullying. The validation, and test data was analyzed through the process to determine the accuracy of this model.

3.1 Data Collection

The dataset utilized in this research is titled:

A Comprehensive Dataset for Cyberbullyinh Detection

It was authored by Naveed Ejaz, Salimur Choudhury, and Fakhra Razi[3]. This CSV file dataset was sourced from Mendeley Data, a reputable repository for research datasets. The structure and text-specific nature makes this data an ideal choice for developing and testing a logistic regression model aimed at detecting cyberbullying. The binary labels allow for clear and concise classification, which is essential for achieving accurate results in the model's predictions[3].

3.1.1 Data Content

The primary focus of the dataset is on text data, specifically designed for the analysis of cyberbullying. This is particularly advantageous compared to alternative datasets that include a mix of formats, such as Yelp reviews and tweets, as it ensures consistency and relevance for text-based cyberbullying detection. The dataset is provided in CSV (Comma-Separated Values) files, which facilitate straightforward data manipulation and analysis. The data is not pre-processed.

3.1.2 Data Features

The dataset contains text strings of user interactions, which form the basis for sentiment and feature analysis. Each text entry is annotated with a binary label, where '1' indicates the presence of cyberbullying and '0' indicates its absence. These labels are vital for training the logistic regression model and evaluating its performance because it tells the model whether a given statement is cyberbullying or not. More data was provided but was not used - such as information on who the statement came from, when it was stated, and what type of cyberbullying it is.

3.2 Methods

The process used to create this model involves data extraction, preprocessing, feature extraction, model selection, model training, and evaluation. The following subsections detail the steps and methodologies used in this research, and the way this model came to be.

3.2.1 Data Extraction

The dataset that was used for this model was provided in a ZIP file format, which was extracted to a designated directory. The CSV file containing the relevant data was then read into a pandas DataFrame for further processing. The information taken from the dataset was the review (the statement) and the status (the binary label) for each row.

3.2.2 Data Preprocessing

There were multiple steps used when it came to preprocessing the data. Firstly, all of the blank rows (the dataset came with some) were marked as rows that should be ignored. Then, all of the characters were made lowercase to ensure the algorithm didn't have information going into it that it did not need - e.g. whether a word is lowercase. Next, all of the special characters were removed. After that, the stop words were removed as well using the NLTK library. Stopwords are words that do not carry much information and orientation[6]. They were unnecessary and did not need to be put into this model. The words were also lemmatized so they could be reduced to their base forms. Lemmatizing is the process where words are converted to their base forms by removing inflectional endings - also using the NLTK library[1]. The steps of data preprocessing were vital to this project and ensure that the text data was standardized and ready for feature extraction.

3.2.3 Feature Extraction

The preprocessed text data was converted into numerical features using the CountVectorizer. This transformed the text into a binary representation through the process of word vectorization. Word vectorization is a methodology in NLP (natural language processing) where words or phrases are converted into corresponding vectors - referred to as word embeddings via count vectorizer[12]. The text is turned into a binary vector representation which allows the input of the text data into the logistic regression model.

3.2.4 Model Selection

It was important to choose the supervised machine learning algorithm that had the most accuracy. Multiple models and their accuracys were tested. This includes the libraries titled: GradientBoostingClassifier, GaussianNB, Enable Hist Gradient Boosting, HistGradientBoostingClassifier, RandomForestClassifier, LogisticRegression, XGBClassifier, LGBMClassifier, and CatBoostClassifier. In the end, the accuracy of the Logistic Regression supervised machine learning algorithm was the highest, so that is what the model uses.

3.2.5 Model Training and Evaluation

The dataset was split into training, validation, and test sets. A logistic regression model from the sklearn library was trained on the training set, and its performance was validated on the validation set. Finally, the model's performance was evaluated on the test set using accuracy, confusion matrix, and classification report metrics.

3.2.6 Statement Input

In Visual Studio Code, an input field was developed where users are able to enter a statement for analysis. The model determines whether the input qualifies as cyberbullying. The same preprocessing steps are applied to the user's input as were used during model training: converting text to lowercase, removing special characters, eliminating stopwords, lemmatizing the words, and vectorizing the data. The processed text is fed into the model, which generates a binary prediction indicating whether the statement is cyberbullying. This binary output is then translated into a user-friendly response. Although a full front-end interface is not yet implemented, users can still receive immediate feedback on whether their input is considered cyberbullying.

4 Results

The primary objective of this research was to develop and evaluate a logistic regression-based AI model that is capable of detecting cyberbullying - specifically within online communications. The results of the study demonstrate that the model achieved a high level of accuracy, making it a promising tool for cyberbullying detection.

4.1 Model Performance

The logistic regression model exhibited an overall F1 accuracy of 86 percent on the test dataset. This indicates a strong ability to correctly classify statements as either cyberbullying or non-cyberbullying. The precision was 87 percent on the statements that are not cyberbullying and 84 for the statements that were. The recall for statements that are not cyberbullying is 93 percent and 72 percent for cyberbullying. The model presents itself as a strong tool for preventing cyberbullying.

4.2 Confusion Matrices

The model predicts whether each input row contained cyberbullying content and provided corresponding prediction labels. This data enabled the creation of a graph that visualizes confusion matrices. The matrices show the distribution of true negatives, true positives, false negatives, and false positives. In the graph, the x-axis represents the model's predictions, with '0' indicating that the model classified the statement as not cyberbullying and '1' indicating it classified the statement as cyberbullying. The y-axis represents the actual classification of the statements, with '0' meaning the statement is not cyberbullying and '1' meaning it is. Two confusion matrices are presented: one for the validation dataset and one for the testing dataset.

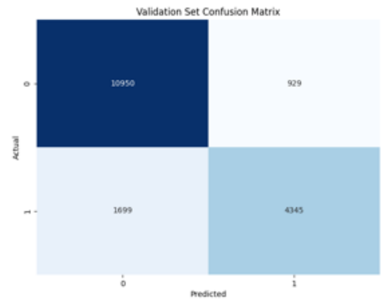


Fig. 1 This is the confusion matrix for the validation data.

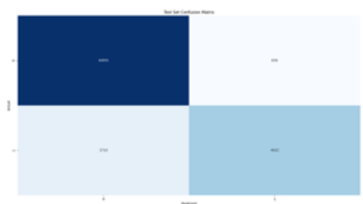


Fig. 2 This is the confusion matrix for the testing data.

4.3 Predicted Probability Distribution

The model also provided the likelihood that a given statement is cyberbullying, which allowed for the creation of predicted probability graphs for both the testing and validation datasets. These graphs display the frequency of statements (y-axis) against their predicted probability of being cyberbullying (x-axis), with color coding to indicate whether the statements were classified as cyberbullying or not. This visualization offers an additional perspective on the model's predictions. In some different research, the model seems to be more confident classifying non-bullying, but that is not the case with this model[9]. This essentially shows how off the model was, and with what data.

4.4 Accuracy Based on Training Data Quantity

When looking at the relationship between the amount of training data and the accuracy of the test data one could see that when the amount of training data increased the accuracy did as well. For example, with one thousand rows of the review and the status, the model had an F1 accuracy of 79 percent. When there were ten thousand rows, the accuracy increased to 83 percent. This continued to happen

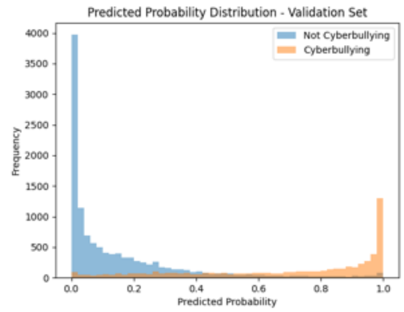


Fig. 3 This is the predicted probability distribution graph for the validation data.

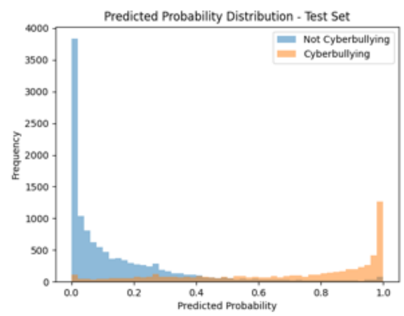


Fig. 4 This is the predicted probability distribution graph for the test data

until there were a little over ninety-thousand rows - the total amount that the dataset has. With the total amount of rows, the accuracy was 86 percent. One could notice when looking at the data that the more rows there were, the less of an increase in the accuracy. Visualized, this would be a logarithmic relationship between amount of rows (x) and accuracy (y).

5 Conclusion

This research was able to successfully develop and evaluate a logistic-regression based AI model designed for detecting cyberbullying in online text communications. It was able to achieve an accuracy of 86 percent and demonstrated potential in distinguishing cyberbullying from non-cyberbullying. It appears to be a promising tool that could be used in the fight against the negativity that cyberbullying brings. While the results are encouraging, the study also identified areas for further work.

5.1 Applications

This model is currently just the back-end but it has potential to be applied to help better the world. The developed model has potential in a variety of digital platforms such as social media, online forums, and messaging services. By integrating this model into these platforms, there could be a feature where there is real-time monitoring and detection of cyberbullying, providing users with instant feedback on harmful content. It could also be used by parents and educators to monitor a given online environment to see if cyberbullying is detected. This would lead to creating a safer and more inclusive online environment.

5.2 Concerns and Next Steps

The model's performance is promising, yet there are still some concerns that need to be addressed in future work. One of the primary goals is increasing the accuracy - specifically reducing the false negatives. 86 percent is good, but there is room for improvemnt. Additionally, expanding the dataset to include a broader range of communications would be useful so the model can be applied across multiple platforms - specifically incorporating the use of emoticons. developing a user-friendly interface or application is another important next step, ensuring that the model can be easily accessed and utilized by end-users. Finally, it may be interesting to explore implementing a continuous learning mechanism to keep the model up to date. By addressing these concerns and taking the next steps, this model can better contribute to a safer online experience for users.

Acknowledgements I want to express my gratitude to Professor Teuscher for his guidance and mentorship throughout this project.

References

1. Vimala Balakrishnan and Ethel Lloyd-Yemoh. Stemming and lemmatization: A comparison of retrieval performances, Apr 2014.
2. PhD Cynthia Vinney. What are the different types of bullying?, Aug 2024.
3. Naveed Ejaz. A comprehensive dataset for automated cyberbullying detection, Jan 2024.
4. Mike Eslea and Josette Rees, Nov 2001.
5. Caitlin Geng. Types of bullying: Examples and next steps, May 2024.
6. Kranti Vithal Ghag and Ketan Shah. Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 International Conference on Computer, Communication and Control (IC4)*, pages 1–6, 2015.
7. Esosa Isokpan, Savannah Fabio, Kaylee Elizabeth, Emma colon, Bree, Katrina, Anonymous, That One Unknown Person, An Anonymous Person, Lauryn, and et al. Cyberbullying facts, 2023.

8. Cheng-Yu Lai, Chia-Hua Tsai, Chung Yuan Christian University Cheng-Yu LaiDept. of Business Administration, and Chung Yuan Christian University Chia-Hua TsaiDept. of Business Administration. Cyberbullying in the social networking sites: Proceedings of the the 3rd multidisciplinary international social networks conference on socialinformatics 2016, data science 2016, Aug 2016.
9. Olle Pellby and Adam Larsson. Solceller på den svenska marknaden, Jun 2015.
10. Andrea Perera and Pumudu Fernando. Accurate cyberbullying detection and prevention on social media, Feb 2021.
11. Adamu Gaston Philipo, Doreen Sebastian Sarwatt, Jianguo Ding, Mahmoud Daneshmand, and Huansheng Ning. Cyberbullying detection: Exploring datasets, technologies, and approaches on social media platforms, May 2024.
12. Prabhu Raghav. Understanding nlp word embeddings-text vectorization, Nov 2019.
13. Daniyar Sultan, Batyrkhan Omarov, Zhazira Kozhamkulova, Gulnur Kazbekova, Laura Alimzhanova, Aigul Dautbayeva, Yernar Zholdassov, and Rustam Abdrakhmanov. A review of machine learning techniques in cyberbullying detection. *Computers, Materials Continua*, 74:5625–5640, 12 2022.
14. Seif-Alamir Yousef and Ludvig Svensson. Detection of bullying with machinelearning: Using supervised machine learning and llms to classify bullying in text, Jun 2024.

Exploring Fractional Leaky Integrate-and-Fire Neurons in Spiking Neural Networks

Holden T. Lee and Daniel M. Monahan

Abstract With an increasing interest in memelement-based neural networks in the field of neuromorphic computing there is a need to explore the characteristics found in these novel devices. In this paper we explore a fractional differential form of the Leaky Integrate-and-Fire neuron, first proposed by Teka et. al. [16], and compare it's performance against networks composed of more traditional determinisic LIF models. We compare their performance on the MNIST dataset, and both troubleshoot and discuss the computational impacts of history dependence in fractional spiking neurons.

1 Introduction

Spiking neural networks are an emergent form of computing which employ a more biologically-inspired approach to computation. Thanks to a vast array of emergent hardware elements currently being explored and implemented in neuromorphic engineering [13], the discovery of spiking network models and co-requisite learning algorithms may break through the performance bottlenecks seen in the modern Von-Neumann computing architectures we use today, namely what is known as the memory wall [17, 12].

Out of the many learning mechanisms found in biological systems, Spike-Timing-Dependent Plasticity (STDP) has shown promise as researchers have been able to show that spike-timing dependant reinforcement learning rules can alter network dynamics and improve performance of action networks, see the improved reach-to-target performance in Neymotin et. al. [10], and decision-making networks of

Holden Lee
Portland State University, 1930 SW 4th Ave., Portland, OR 97201

Daniel Monahan
Portland State University, 1825 SW Broadway, Portland, OR 97201

Skorheim and Lonjers [14]. There have also been suggestions of STDP characteristics in memcapactive systems, see Pershin and Di Ventra's work [11], which gives STDP candidacy for hardware implementation. It is worth noting that Pershin and Di Ventra have also shown that a memcapacitive neural network may serve as a low-power solution, promising significant efficiency benefits.

2 Related Work and Background

2.1 Spiking Neural Networks

Artificial neural networks (ANNs) have a long history in machine learning and power the most advanced and well-known artificial intelligence tools in our time. They draw their inspiration from the structure of biological neural networks and are composed of a number of nodes, called artificial neurons, that compute some form of nonlinear activation function. These networks have proven to be highly successful in the approximation of arbitrary functions when an appropriate learning algorithm is applied to their learnable parameters. Their strength comes at a cost, however, as large networks required for solving complex tasks require a great number of expensive multiplication operations. For this reason, Spiking Neural Networks (SNNs) have been explored, as they replace the high-power multiplication operations with power efficient multiplexing circuits, for more details see the study performed by Han et al. [9].

The spiking neural network is an event-driven neural network model that replaces real-valued inputs with a series of event-based inputs known as spikes or spike-trains. Neurons receive these input spikes either as synaptic currents from pre-synaptic neurons, or as current injected into the cellular membrane via an experimenter or an extracellular pressure gradient. The charges delivered by these incoming currents cause the neuron's membrane potential to rise. When this membrane potential exceeds a certain threshold called an action potential, the neuron fires, sending an output spike to post-synaptic neurons. A spiking artificial neuron replicates this system. The artificial neuron receives input spikes from its pre-synaptic neurons, and updates its membrane potential, stored as a real number, accordingly. When its potential reaches or exceeds the threshold, it fires, sending a spike to its post-synaptic neurons. After firing, the neuron's potential resets to its resting value, and begins again[1]. In a SNN, rather than the constant flow of information between neurons in an ANN, the stream of data is both infrequent and discretized, as spikes are $\{0, 1\}$ values.

A major challenge presented by SNNs arises from the credit assignment problem, due to the interpretability of spikes which carry both spatial and temporal informa-

[1] In some models, the artificial neuron enters a brief refractory period after firing, during which it may not spike. This feature is also inspired by the refractory property of biological neurons.

tion. This dense encoding of information is believed to play an important role in biological information transmission and computation. For learning tasks that have an inherent time component, for example video processing, this may be trivial, but in cases where time is not an inherent property of the task, input data must be encoded as spike trains over time that can be fed to the SNN. This encoding is typically done by interpreting input data as commands for when or how often input spikes should be sent to the network. We will discuss these methods in greater detail in Section 3.2.

2.2 Leaky Integrate-and-Fire

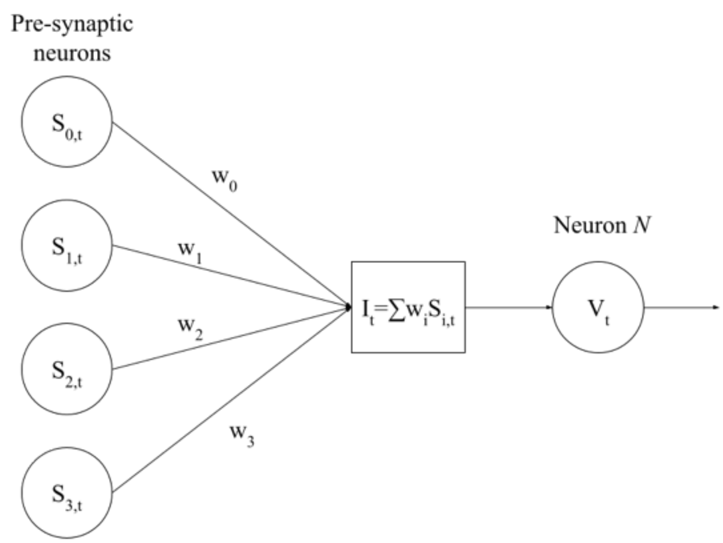


Fig. 1 Diagram of a spiking artificial neuron.

We now move our discussion to the mechanisms of spiking artificial neurons. Figure 1 shows a diagram of a spiking neuron. At any given time t, for every neuron N, the spikes $S_{i,t} \in \{0, 1\}$ of pre-synaptic neurons are multiplied by their synaptic weights w_i and summed to form the input current for N at time t, I_t. The input is then applied to N via a function to generate its membrane potential at time t, V_t. A common model for the dynamics of the neurons membrane potential is given by,

$$C \cdot \frac{dV(t)}{dt} = I(t),$$

where C is the capacitance of the membrane, and I and V are considered to be current and voltage as functions of time, respectively. The notation I_t in the previous paragraph is equivalent to $I(t)$ here. This model is known as the perfect integrate-and-fire, first proposed by Louis Lapicque [1]. In this model there is no diffusion of charges when the neuron is void of action potentials, however in biological neurons this is not the case. Due to 'leak' in the cell membrane caused by activation of ion channels there is diffusion of charge-inducing ions from within the cell membrane, causing its electrical potential to decay towards its resting value in the absence of stimulating current [5]. We can introduce this behavior by incorporating a decay term in the previous equation.

$$C \cdot \frac{dV(t)}{dt} = I(t) - gV(t),$$

Here, g, is the conductance of the leaky ion channels within the cell membrane. The $-gV(t)$ term introduces a 'leak' that mimics the diffusion of ions causing decay in membrane potential. The behavior of the model is this; in the presence of input currents, incoming synaptic signals are summed together and integrated by the cell membrane potential. In the absence of input currents the membrane potential decays according to the membrane conductance, g. For this reason the model is called leaky integrate-and-fire model (LIF). Neurons that exhibit this behavior are then called LIF Neurons, a popular choice for SNN computing units.

Despite its ubiquity, the leaky integrate-and-fire model has its limitations. As it is defined by a first-order differential equation, LIF may be described discretely using the Euler method:

$$V(t + dt) = dt \cdot \left(\frac{I(t) - gV(t)}{C}\right) + V(t),$$

where dt is a small change in time. As this formula suggests, the future value of the voltage is dependent only on its immediate previous value, and the input current. Therefore, the firing rate of an artificial neuron implementing leaky integrate-and-fire is restricted only by the magnitude of the input current.

However, it is well-documented that the membrane potential of biological neurons is history-dependent, meaning that a neuron's future response to input is influenced by its previous behavior ([6], [15], [2]). The LIF model's restriction to only requiring one previous value makes it a memory-less process, which limits its ability to accurately describe neuronal behavior. We attempt to capture this behavior using fractional calculus in upcoming section.

2.3 Fractional Order Calculus

To transform the LIF into a memory-driven model, we may introduce a **fractional order** derivative. The fractional order derivative is an abstraction of the notion of multiple derivatives. For a function $f(x)$ and a positive integer n, $\frac{d^n f}{dx^n}$ is defined as differentiating f n times with respect to x. In the realm of fractional calculus, we no longer restrict n to the positive integers, and allow n to be any real number. We interpret $n > 0$ to be the n-th order derivative, and $n < 0$ to be the n-th order integral.

There are several different definitions for fractional-order derivatives/integrals, which draw inspiration from different ideas in standard calculus. Here, we will discuss the Caputo fractional derivative, which is advantageous for its ability to be approximated numerically[2]. To understand it, we must first describe the Riemann-Liouville fractional integral.

2.3.1 Riemann-Liouville Fractional Integral

The integration operator I is the first-order integral of a function $f(x)$, i.e.,

$$(If)(x) = \int_0^x f(t)dt.$$

The Cauchy formula for repeated integration tells us that for $n \in \mathbb{N}$, the n-th order integral is given by

$$(I^n f)(x) = \frac{1}{(n-1)!} \int_0^x (x-t)^{n-1} f(t)dt.$$

The reason we cannot simply allow this definition to apply to any real number n is the fact that the factorial function ! is only defined on the natural numbers. However, the gamma function, defined to be $\Gamma(n) = \int_0^\infty t^{n-1} e^{-n} dt$, is an extension of the factorial function to the real (and complex) numbers, and hence has the property that if $n \in \mathbb{N}$, then $\Gamma(n) = (n-1)!$.

So the Riemann-Liouville fractional integral of order n is

$$(J^n f)(x) = \frac{1}{\Gamma(n)} \int_0^x (x-t)^{n-1} f(t)dt.$$

2.3.2 Caputo Fractional Derivative

Recall that our definition for orders dictates that negative orders describe integrals and positive orders describe derivatives. Let $\alpha \in \mathbb{R}, \alpha > 0$ be the order of the

[2] We will discuss this approximation in Section 2.4

fractional derivative of interest. Let $n = \lceil \alpha \rceil$, i.e., the smallest whole number greater than α. We have

$$n - (n - \alpha) = \alpha \iff -(n - \alpha) + n = \alpha.$$

Recall that higher order derivatives are simply the result of differentiating over and over again, i.e.,

$$\frac{d^m f}{dx^m} = \frac{d}{dx}\frac{d}{dx}\dots\frac{d}{dx}f(x).$$

We can extend this idea to prove that for positive integers a, b, c such that $a + b = c$,

$$\frac{d^c f}{dx^c} = \frac{d^a}{dx^a}\Big(\frac{d^b f}{dx^b}\Big).$$

Hence, with D as the differentiation operator, we have that

$$D^{-(n-\alpha)}D^n f(x) = D^\alpha f(x) \implies I^{n-\alpha}D^n f(x) = D^\alpha f(x),$$

as $-(n-\alpha) < 0$ indicates that it represents the $(n-\alpha)$-order integral. $D^n f(x)$ is simply the n-th order derivative $f^{(n)}(x)$ as $n \in \mathbb{N}$.[3] So the Caputo α-th order derivative [3] is therefore given by

$${}^C D^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-t)^{n-\alpha-1} f^{(n)}(t)dt.$$

2.4 Fractional Leaky Integrate-and-Fire

Now that we have some context for what a fractional derivative looks like, we may discuss its applications to the leaky integrate-and-fire model. Recall that LIF uses a first-order differential equation. We shall now use a α-th order **fractional order differential equation** in its place:

$${}^C D^\alpha V(t) = \frac{I(t) - g(V(t) - V_L)}{C},$$

where we use the Caputo derivative rather than the classical, first-order derivative, and introduce a leak reversal voltage V_L, as used in [16].

The fractional order leaky integrate-and-fire differs from its classical counterpart in that fractional order differential equations are known to represent non-Markovian (i.e., history-inclusive) processes, so the fractional LIF model is able to integrate its history, while the classical model cannot. To see this more clearly, we will briefly

[3] There are many different notations for derivatives. Please note that $f^{(n)}(x)$, $D^n f(x)$, and $\frac{d^n f}{dx^n}$ all refer to the same thing.

discuss the method by which we approximate the fractional LIF model.

Given a very small, discrete time step dt and the value of V at N such time steps, the L1 scheme [16] allows us to approximate the Caputo fractional derivative at time t_{N-1} as

$$\frac{d^\alpha V(t_{N-1})}{dt^\alpha} \approx \frac{(dt)^{-\alpha}}{\Gamma(2-\alpha)}\Big(\sum_{k=0}^{N-1}[V(t_{k+1})-V(t_k)][(N-k)^{1-\alpha}-(N-1-k)^{1-\alpha}]\Big) \quad (1)$$

Note that in this equation we use the derivative notation $\frac{dV}{dt}$, but we are referring to the Caputo derivative.

Consider the sum in equation (1). The $(N-1)$-th element of the sum is

$$[V(t_N - V(t_{N-1})][(N-(N-1))^{1-\alpha} - (N-1-(N-1))^{1-\alpha}] = V(t_N) - V(t_{N-1}).$$

As such, we can rearrange the equation to obtain

$$V(t_N) - V(t_{N-1}) \approx \Gamma(2-\alpha)(dt)^\alpha\Big(\frac{d^\alpha V(t_{N-1})}{dt^\alpha}\Big) - \sum_{k=0}^{N-2}[V(t_{k+1}) - V(t_k)][(N-k)^{1-\alpha} - (N-1-k)^{1-\alpha}],$$

and hence

$$V(t_N) = \Gamma(2-\alpha)(dt)^\alpha\Big(\frac{d^\alpha V(t_{N-1})}{dt^\alpha}\Big) + V(t_{N-1}) - \sum_{k=0}^{N-2}[V(t_{k+1}) - V(t_k)][(N-k)^{1-\alpha} - (N-1-k)^{1-\alpha}]. \quad (2)$$

But recall that $\frac{d^\alpha V(t_{N-1})}{dt^\alpha}$ is exactly the fractional leaky integrate-and-fire. So Equation (2) may be rewritten as

$$V(t_N) = \Gamma(2-\alpha)(dt)^\alpha\Big(\frac{I(t_{N-1}) - g[V(t_{N-1}) - V_L)]}{C}\Big) + V(t_{N-1}) - \sum_{k=0}^{N-2}[V(t_{k+1}) - V(t_k)][(N-k)^{1-\alpha} - (N-1-k)^{1-\alpha}] \quad (3)$$

So we now have a way to compute the membrane potential at the Nth time step, given the values at the previous $N-1$ steps. In sharp contrast to the first-order approximation, a Markovian process, the fractional model requires the integration of voltages at all previous time steps. This inclusion allows the model to adapt its firing rate in response to the density of previous spikes, for a small fractional order $0 < \alpha << 1$[16]. A fractional LIF neuron that has spiked more frequently recently becomes more excitable and has an increased firing rate, while a neuron that has spiked rarely or not at all becomes less inclined to spike and has a lower firing rate.

However, when applying the fractional LIF to SNN simulations, the history component, known as the memory trace, also introduces a steep physical memory cost during simulation which may be avoided on appropriate neuromorphic hardware. The derived equation also requires at least two previous time steps of history to function, so after defining an initial state $V(t_0)$, we apply the first-order leaky model to obtain a value for $V(t_1)$.

2.5 Learning with SNNs

Now that we have discussed the dynamics of spiking neurons, we can now explain the mechanisms for learning in spiking neuron networks. As Section 3.2 suggests, spiking neural networks can be treated the same way as artificial neural networks, just with an additional conversion step at both the input and output layers of the network, to translate between numeric values and spikes.

As a result, some learning rules used for ANNs can be adopted for SNNs. Backpropagation, which combines gradient descent with an error (loss) function to perform weight updates, can be applied to spiking neural networks. However, backpropagation is an unpopular choice, primarily because it is not a biologically realistic learning rule [4]; as the inspiration for spiking neurons is bio-inspired, it is often thought that bio-inspired learning rules are more effective or interesting.

Furthermore, backpropagation has limited compatibility with some architecture choices. It may be valuable to have a mix of excitatory and inhibitory connections between neurons, such as in [4]. The former send positive signals to post-synaptic neurons, increasing their potential, while the latter, as their name suggests, send negative signals, decreasing the potential of post-synaptic neurons. Within a network, the polarity of this signal is encoded in the weight between the neurons, which is then modified during training by the backpropagation algorithm. Because backpropagation only seeks to make changes resulting from gradient calculations based on the loss, it may or may not preserve inhibitory and excitatory connections.

Instead, research on SNNs often uses a learning rule called Spike Timing Dependent Plasticity (STDP), a property observed in biological neurons. STDP changes synaptic weights based on the spike timing of pre- and post-synaptic neurons. If a post-synaptic neuron fires shortly after a pre-synaptic neuron, the connection between the two is considered more 'relevant', and increases in strength. Conversely, if a post-synaptic neuron fires *before* a pre-synaptic neuron, the connection is considered less important, and should be decreased in strength. We can further this idea to have a spiking network learn a specific task by introducing a critic that modulates the scale of weight updates performed through STDP, based on the task at hand [10].

3 Data and Methodologies

3.1 Data Collection

To assess the differences between the dynamics of the first-order and fractional-order leaky-integrate-and-fire models, we built two neural networks, one for each model, and trained them on the classical MNIST handwritten digit classification task. We chose to use $\alpha = 0.2$ for our fractional order, as it yields a lower first spike latency than $\alpha = 0.15$, the other order used in [16]. Decreasing the delay to the first spike also decreases the total number of steps the network needs for evaluation, which in turn decreases the memory cost of simulation.

3.1.1 MNIST Classification

Images in the MNIST dataset are usually 28x28 pixels, but to reduce costs of simulation they have been scaled down to 10x10 using PyTorch's Torchvision library. Both neuron models used an identical network architecture. We used a fully-connected, feedforward network consisting of an input, hidden, and output layer. The input layer has 100 neurons, determined by the dimension of our inputs, the hidden layer has 1000 neurons, and the output layer has one neuron for each possible class, for 10 total. We shuffled the MNIST dataset and used batch learning to train the network.

In each case, the network was allowed to train until reaching peak performance at testing. The first-order model needed only 2 epochs to reach convergence, but the fractional-order model required 5 epochs to reach maximum performance. We used cross-entropy (CE) loss function for backpropagation for both models, as well as the mean squared error (MSE) loss function for the fractional-order model, both of which are described in Section 3.2.2.

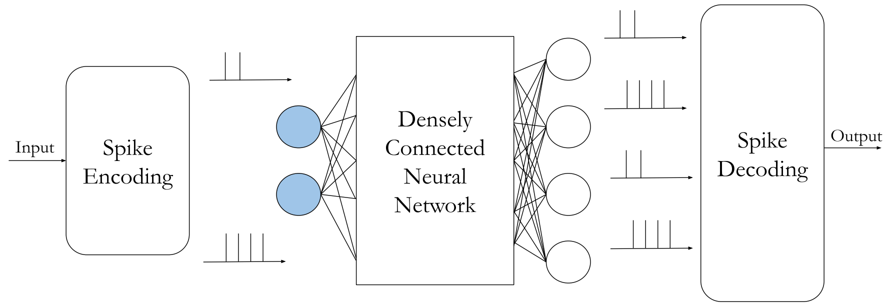


Fig. 2 Architecture of the feedforward network.

3.1.2 Robustness Testing

We then looked to see if the fractional-order network was more robust than the first-order one. After training both models, we tested their resistance to failure across two metrics: their ability to overcome 'dead' inputs and the introduction of noise into the data. 'Dead' inputs are ones artificially set to 0 (and hence generate no spikes when converted to input layer spikes), so the network overall had fewer spikes to process. We introduced noise by artificially perturbing the values of some inputs, which resulted in different input spike rates.

For both robustness metrics, we performed a 100-sample Monte Carlo simulation. In each sample of a simulation, n randomly selected inputs were permuted according to the metric used, for each $n \in \{10x \mid 0 \leq x \leq 10\}$. We then calculated the network's classification accuracy on the perturbed dataset, and took the average accuracy over all 100 samples.

3.2 Methods

We built our neural network models using the snnTorch library, which extends the functionality of PyTorch to spiking neural networks. Naturally, our models were built in Python. snnTorch evaluates SNN models through a discretization of time into steps, which means that inputs and outputs are expressed as spikes at each time step.

3.2.1 Spike Encoding

We perform this encoding in two ways. In either case, we assume the inputs are normalized and are within the open interval $(0, 1)$. We also assume a predetermined number N of time steps in which spikes appear.

The first approach is rate encoding, under which the input value S_i at the i-th input neuron is used to define a binomial distribution $B(N, S_i)$. We then sample from this distribution to obtain a spike train over N steps for the i-th input neuron.

The second is latency encoding, in which each input neuron fires only a single spike, and the magnitude of the input is interpreted as the timing of that spike. More specifically, a larger input corresponds to an earlier spike time, and a smaller input leads to a later spike.

It is clear that to maximize power efficiency, latency encoding should be the better choice, as the presence of fewer spikes corresponds to less power dissipation, but more spikes means more learning. Sparse activity at the input layer produced by latency encoding proved problematic during backpropogation learning, as activity throughout the network was decreased, and training was inhibited greatly. Rate encoded spikes always led to very dense spiking in the network, and so was the most consistent way to guarantee outputs.

We can apply these encoding schemes in reverse to decode spikes at the output of our networks; the network's output is the output neuron with the most frequent spikes under rate decoding, or the neuron that spikes earliest under latency decoding.

3.2.2 Loss Functions and Credit Assignment

As rate encoding proved to be more effective for network activity, we focused on encoding and decoding via spike frequency, which was reflected in our learning rules.

For MNIST classification, we trained our networks using backpropagation, which requires the use of a loss function. We trained the both the classic leaky integrate-and-fire and fractional LIF networks using cross entropy loss (CE), which measures the error based on the number of times the incorrect and correct output neurons fired: spikes at incorrect classes contributed positively towards the value of the loss, while spikes from the correct class contributed negatively. This function was provided by snnTorch.

We also used another loss function which takes as input desired firing rates for correct and incorrect classes, and measures the mean squared error between the network's output firing rates and the desired firing rates. This mean squared error loss (MSE) performed *better* for MNIST classification on a network using the fractional LIF model. We believe that the more complex dynamics of the fractional-order model made it more suitable for a loss function that allows for a greater margin of error in the firing rates of correct and incorrect output neurons.

4 Results

4.1 MNIST Classification

Ultimately, we found that first-order model performed better, despite training over fewer epochs. Its final accuracy was ≈ 95%. Meanwhile, the fractional-order model performed poorer overall, with ≈ 82% accuracy. After switching to the MSE loss function, the fractional LIF network increased in performance to ≈ 90%.

The MSE loss appeared to generate better spiking behavior at the output layer; Figures 3 and 4 illustrate this fact. The output spikes for the MSE network are much 'cleaner'. In Fig 4, the title indicates the network was given an image of the number 1, and it is clear that the output neuron corresponding to 1 fires significantly more than any other output neuron. On the other hand, in Fig 3, a 5 was shown to the network, but it is visually unclear which output neuron fires the most[4]. Furthermore, the hidden layer of the MSE network also displays less dense spiking behavior, which might indicate that it is creating fewer useless spikes, and hence the output layer becomes less inundated with incorrect spikes.

[4] In fact, the neurons with the highest firing rates here correspond to 3, 5, and 9, and it appears that the network has actually misclassified this image as a '3'.

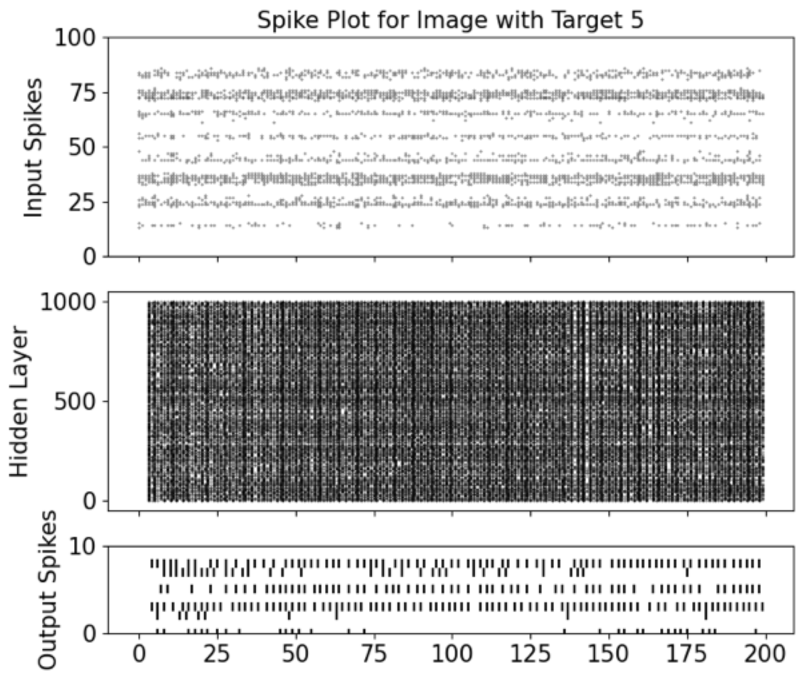


Fig. 3 Spike raster plot for the fractional-order network using cross-entropy loss. Each tick corresponds to a spike

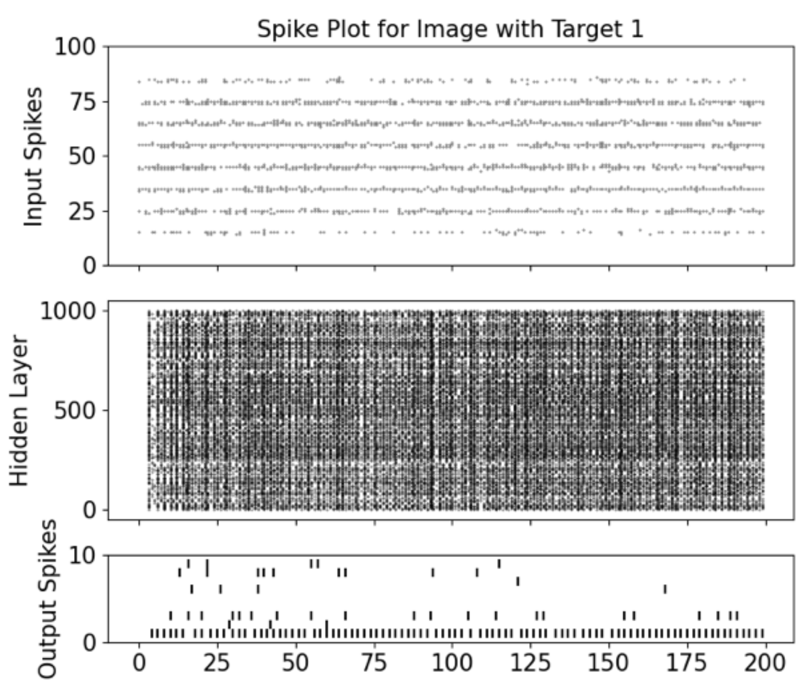


Fig. 4 Spike raster plot for the fractional-order network using mean squared error loss.

However, both fractional-order models performed worse than the classical LIF, which is reflected in the spike plot for the first-order model. Figure 5 shows significantly less noisy behavior in the output layer, and incorrect classes rarely fire. We believe that the spike adaptation property of the fractional leaky integrate-and-fire, which differentiates it from the classical LIF, is at least partially responsible for the increase in output noise. As mentioned in Section 2.4, the presence of previous spikes encourage future spikes, and the absence of spikes discourages it. While the latter helps suppress incorrect classes, the former does the opposite; when incorrect classes spike, they become more prone to spiking, leading to more noise in the output.

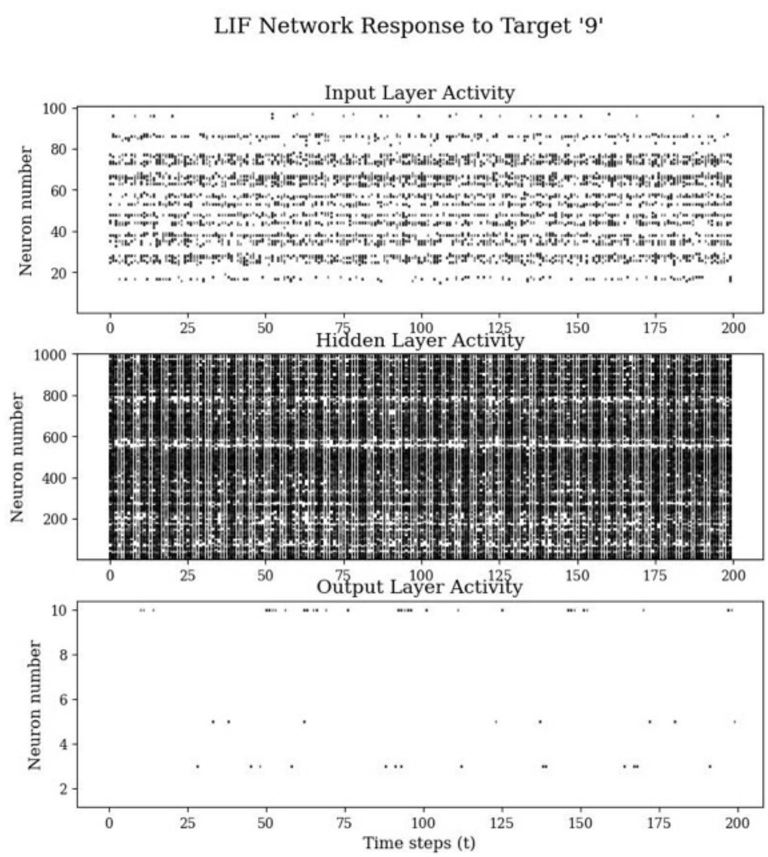


Fig. 5 Spike raster plot for the standard LIF network.

4.2 Robustness Testing

We analyzed the ability of the classical LIF network, as well as both fractional-order networks, to respond to removed and noisy inputs (separately). Figures 6-9 show the results of our Monte Carlo experiments on the fractional-order network. Interestingly, both show the same behavior in response to both types of perturbation, and, despite the initial difference in performance, converge to the same curve. We see that randomly removing inputs has little effect for small amounts, but this is likely due to the fact that several inputs would never fire anyways and are uninformative[5]. On the other hand, introducing noise leads to a drastic decrease in performance, which slows down as the network approaches random chance.

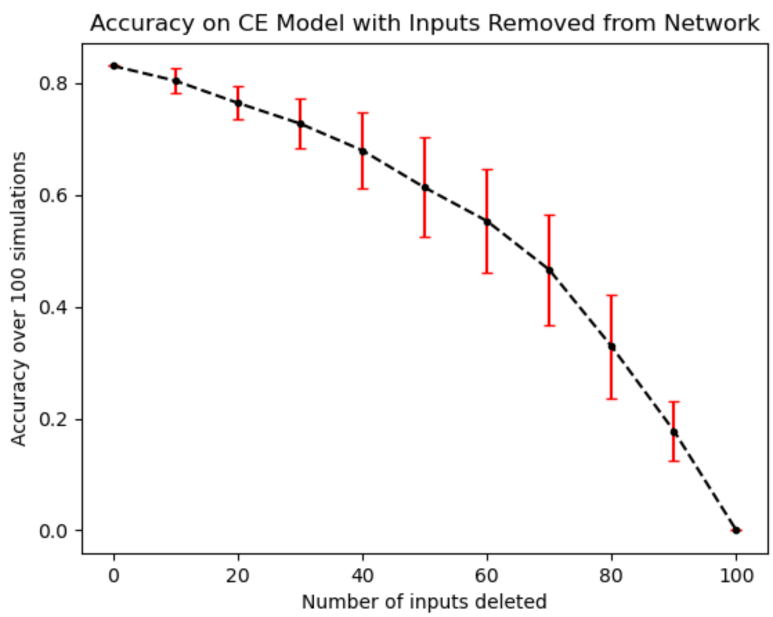


Fig. 6 Input Removal on the fractional-order network with CE loss

[5] For example, border pixels are always 0 valued

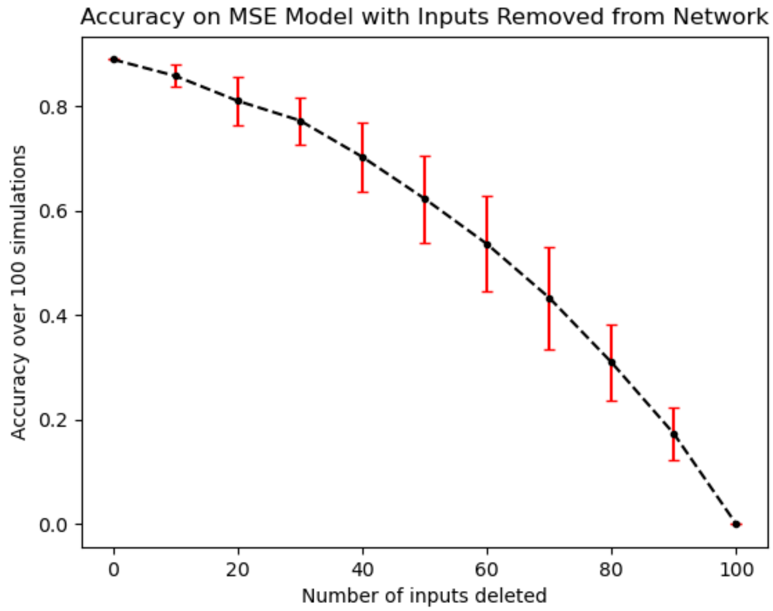


Fig. 7 Input Removal on the fractional-order network with MSE loss

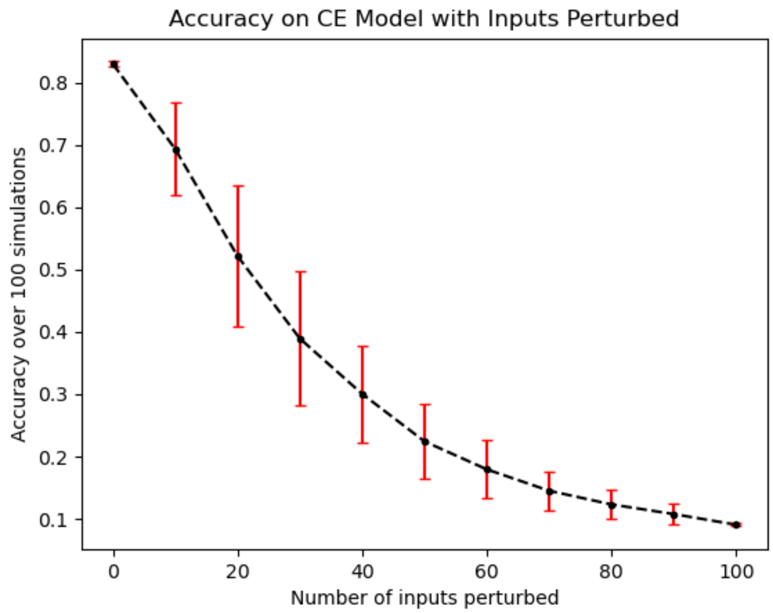


Fig. 8 Input Perturbation on the fractional-order network with CE loss

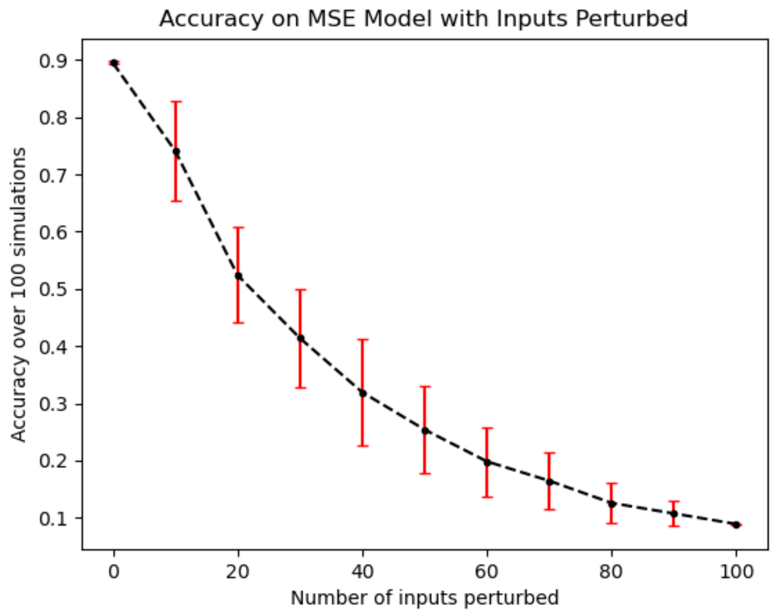


Fig. 9 Input Perturbation on the fractional-order network with MSE loss

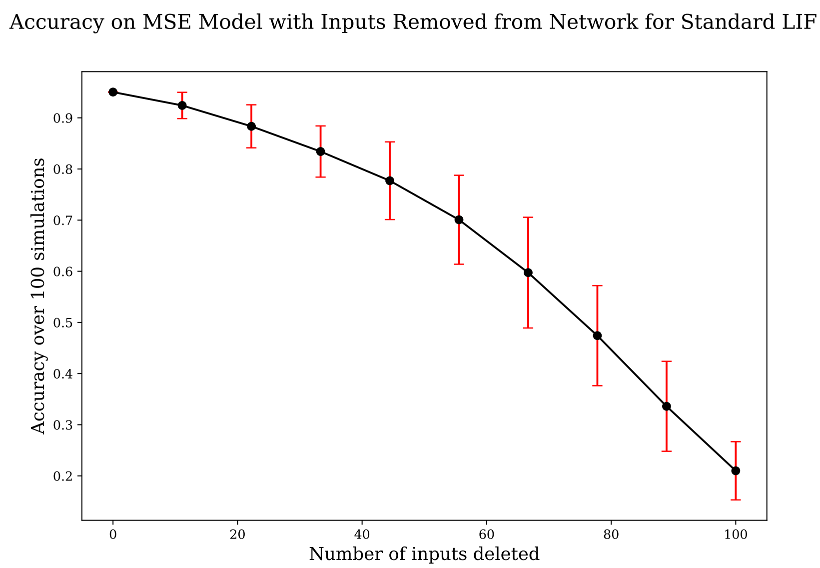


Fig. 10 Input removal on standard LIF network with MSE loss

5 Conclusion

In conclusion we found that fractional LIF dynamics did not improve classification accuracy when using backpropagation and rate encoding. This is not to say that adaptive dynamics are not worth exploring, as future research interests are to explore temporal and population coding schemes, as well as the dynamics of fractional LIF neurons in recurrent networks. Benefits of understanding fractional LIF dynamics are important, as memcapactive and supercapacitive devices currently being explored have been accurately represented using fractional models [7]. It is possible that an awareness of fractional effects will be necessary for further implementation of spiking neural networks if such memcapacitive or memresisitive systems prove efficient.

5.1 Discussion

Here we attempt to justify our implementation and provide reason for the fractional order LIF model presented, and address various choices made in our methodology. We hope not to dismiss the critic within the reader but to provide a basis for inquisition and analysis in our design choices.

5.1.1 Why Fractional Order Neurons?

Fractional ordered dynamics have been explored for many control system applications as well as in neuroscience as it forms a method of introducing chaotic behaviour and state dependence seen in many natural systems. For instance, the dynamics of a non-fractional Leaky Integrate-and-Fire neuron cannot explain spiking adaptation seen in the central nervous system explored in [8], where a state-dependence defined by the presence of Ca^{2+} ions is shown to modulate the neuronal dynamics. As for the introduction of fractional dynamics into the LIF model, Santamaria [16] shows that the level of spike adaptation is controlled by the order of the fractional derivative.

5.2 Concerns and Next Steps

As it is clear that the fractional leaky-integrate-and-fire is an inferior model for classification despite being more biologically-inspired than the classical LIF, we believe that it may be better suited both to tasks and architectures that make better use of its memory capabilities. We plan to extend our work to the Santa Fe trail, a reinforcement learning task in which an agent must learn to follow a fixed trail of food pellets[6]. As the trail must be traversed over time, we hope that the adaptation property of the fractional LIF will make it perform better than the first-order LIF. We also plan to introduce recurrence into our network architecture, again to try and make use of the fractional LIF's memory capacity.

Acknowledgements Thanks to Dr. Chrisof Teuscher for his indelible support and insight throughout our research, to our PI Dr. Fidel Santamaria for providing a basis for our work and exploration, and to the National Science Foundation for supporting our work.

References

1. L.F. Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res Bull*, 50:2, 1999.
2. Fabiano Baroni, Joaquín J. Torres, and Pablo Varona. History-dependent excitability as a single-cell substrate of transient memory for information discrimination. *PLoS ONE*, 5(12), Dec 2010.
3. M. Caputo. Linear models of dissipation whose q is almost frequency independent–ii. *Geophysical Journal International*, 13(5):529–539, Nov 1967.
4. Peter U. Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9, Aug 2015.
5. Wulfram Gerstner, Werner Kistler, Richard Naud, and Liam Paninski. *Part I: Foundations of Neuronal Dynamics*. Cambridge University Press, 2014.

[6] Or pheromone particles, depending on who you ask.

6. Gail Gilboa, Ronen Chen, and Naama Brenner. History-dependent multiple-time-scale dynamics in a single-neuron model. *The Journal of Neuroscience*, 25(28):6479–6489, Jul 2005.
7. José Francisco Gómez Aguilar. Behavior characteristics of a cap-resistor, memcapacitor, and a memristor from the response obtained of RC and RL electrical circuits described by fractional differential equations. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, 24:1421–1433, 2016.
8. Go Eun Ha and Eunji Cheong. Spike frequency adaptation in neurons of the central nervous system. *Experimental Neurobiology*, 26(4):179–185, 2017.
9. Bing Han, Abhronil Sengupta, and Kaushik Roy. On the energy benefits of spiking deep neural networks: A case study. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 971–976. IEEE, 2016.
10. Samuel A. Neymotin, George L. Chadderdon, Cliff C. Kerr, Joseph T. Francis, and William W. Lytton. Reinforcement learning of two-joint virtual arm reaching in a computer model of sensorimotor cortex. *Neural Computation*, 25(12):3263–3293, 2013.
11. Y. V. Pershin and M. Di Ventra. Memcapacitive neural networks. *Electronics Letters*, 2013.
12. Ashley Saulsbury, Fong Pong, and Andreas Nowatzyk. Missing the memory wall: the case for processor/memory integration. *ACM SIGARCH Computer Architecture News*, 24(2):90–101, 1996.
13. Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7):529–544, 2020.
14. Steven Skorheim, Peter Lonjers, and Maxim Bazhenov. A spiking network model of decision making employing rewarded STDP. *PLoS ONE*, 9(3):e90821, 2014.
15. Daniel Soudry and Ron Meir. History-dependent dynamics in a generic model of ion channels - an analytic study. *Frontiers in Computational Neuroscience*, Apr 2010.
16. Wondimu Teka, Toma M. Marinov, and Fidel Santamaria. Neuronal spike timing adaptation described with a fractional leaky integrate-and-fire model. *PLoS Computational Biology*, 10(3), Mar 2014.
17. Wm A Wulf and Sally A McKee. Hitting the memory wall: Implications of the obvious. *ACM SIGARCH computer architecture news*, 23(1):20–24, 1995.

Integrating Deep Q-Networks and Reinforcement Learning for Autonomous Navigation in the Santa Fe Trail Problem

Hussein Khodor

Abstract The Santa Fe Trail Problem, a standard benchmark in genetic programming, involves guiding an artificial agent to collect food pellets scattered along a predefined trail. A neural network based approach was developed for this task. In order to train the agent to navigate the trail autonomously, a reinforcement learning (RL) algorithm, specifically a Deep Q-Network (DQN) employed in PyTorch, was implemented together with Pygame for simulation and Matplotlib for data visualization. The model's performance was analyzed in terms of navigation efficiency, focusing on the collection of food pellets in as minimal time steps as possible. The results indicate that DQNs can effectively handle the complexities of dynamic and unpredictable environments such as the Santa Fe Trail.

1 Introduction

In artificial intelligence and machine learning, navigation and pathfinding are pivotal challenges that drive much of the research in developing autonomous systems. The Santa Fe Trail Problem, a benchmark problem in AI, offers a controlled environment to address these challenges by guiding an agent to collect scattered food pellets along a relatively complex trail. Traditionally tackled through genetic programming, this problem provides a valuable opportunity to apply and test reinforcement learning (RL) techniques, particularly Deep Q-Networks (DQNs).

Hussein Khodor
Portland State University, 1825 SW Broadway, Portland, OR 97201, e-mail: hkhodor@pdx.edu

1.1 Q-Learning

Q-Learning is a fundamental method in reinforcement learning, where an agent learns to make decisions by interacting with its environment. It operates in a finite state and action space, meaning there are specific actions the agent can take (e.g., move left, right, up, down) and specific states the agent can be in (e.g., different positions on a grid). The core idea of Q-Learning is to assign a value, called the Q-value, to each state-action pair. This value represents the expected cumulative reward of taking a specific action in a given state and then following the optimal policy thereafter. The value of each state-action pair is a combination of the immediate reward received after taking the action and the expected sum of all future rewards if the best actions are taken in subsequent states. Initially, when the agent has no information about the environment, the Q-values are essentially guesses. However, as the agent explores more states and actions, it updates these Q-values iteratively. This process is guided by the Bellman equation, which is a recursive formula used to calculate the Q-value for each state-action pair.

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

In this equation:

- $Q(s,a)$ is the Q-value of taking action a in state s.
- $r(s,a)$ is the immediate reward received after taking action a in state s.
- γ is the discount factor, which determines the importance of future rewards.
- $\max_{a'} Q(s',a')$ is the maximum Q-value for the next state s', considering all possible actions a'.

The Bellman equation helps the agent update its Q-values based on new experiences, gradually improving its policy for selecting actions to maximize cumulative reward.

1.2 Deep Q-Networks and Q-Learning: The Relationship

Traditional Q-Learning stores Q-values in a table, which becomes impractical for environments with large state and action spaces. This is where Deep Q-Networks come into play. DQNs use neural networks to approximate the Q-value function, enabling the agent to handle complex environments with high-dimensional state and action spaces. A DQN consists of a neural network that takes the current state as input and outputs Q-values for all possible actions. Instead of updating a table of Q-values, the neural network is trained using the Bellman equation, where the Q-values are adjusted based on the predicted rewards from the network itself.

1.3 Experience Replay

Experience Replay is a crucial technique in DQN that enhances training stability and efficiency by storing past experiences in a replay memory. In reinforcement learning, an agent's experience consists of lists detailing the state, action taken, reward received, and next state, denoted as (s, a, r, s'). Training on these experiences sequentially can lead to highly correlated data, which can destabilize learning due to the non-independent nature of consecutive experiences. To mitigate this issue, DQN uses a replay memory—a buffer that stores a collection of past experiences. Instead of learning from the most recent experiences alone, the agent samples random batches from this memory for training. This approach helps break the correlation between consecutive experiences, leading to more stable and effective learning. By incorporating a diverse range of experiences, the agent can learn from a broader array of scenarios, improving its overall performance. During training, the agent randomly selects a batch of experiences from the replay memory, which is used to update the Q-network. This method ensures that the network benefits from a varied set of experiences, enhancing its ability to generalize and make accurate predictions.

1.4 The Epsilon-Greedy Strategy

The epsilon-greedy strategy is employed in reinforcement learning to manage the trade-off between exploration and exploitation. Initially, the agent needs to explore the environment thoroughly to understand the consequences of different actions and gather diverse experiences. To facilitate this, this strategy uses a variable called epsilon (ϵ), which dictates the probability of selecting a random action versus the action with the highest Q-value. At the beginning of training, epsilon is set to a high value, which encourages the agent to explore a wide range of actions and learn about various scenarios. As training progresses, epsilon is gradually reduced to 0 or even less than 0. This reduction decreases the likelihood of selecting random actions and increases the agent's tendency to exploit what it has learned about its environment to maximize rewards. The epsilon-greedy strategy ensures that the agent does not become fixated on suboptimal actions by striking a balance between exploring new possibilities and exploiting known strategies that yield higher rewards. This gradual transition from exploration to exploitation helps the agent converge towards an optimal policy while avoiding premature convergence on suboptimal solutions.

1.5 The Justification

The decision to employ a DQN was driven by the need to explore the capabilities of modern neural networks in handling dynamic and non-linear environments. Unlike genetic programming, which relies on evolutionary algorithms, DQNs use deep learning to enable agents to learn optimal policies through continuous interaction with the environment. This approach not only tests the efficacy of DQNs but also demonstrates their potential in solving complex real-world scenarios. In other words, this task is not just about solving a seemingly useless and unimportant predefined problem, instead, it is about exploring the potential of neural networks and reinforcement learning to handle complex, real-world scenarios.

1.6 The Approach

The approach that was taken to solve this problem leverages the power of neural networks and RL, specifically focusing on DQNs. The objective is to develop a robust solution that enables an agent to learn and adapt to the trail, optimizing its pathfinding abilities through continuous interaction and feedback. The first step in the approach was to clearly define the Santa Fe Trail Problem and establish the simulation environment. This was achieved using Pygame, which allowed for an accurate simulation of the trail and the agent's movements within a dynamic and interactive setting. Next, the agent's characteristics were defined, detailing its movement capabilities and initial conditions to ensure it could navigate the environment effectively. The core of the method involved implementing a DQN with PyTorch, chosen for its flexibility and robust features in building and training neural networks. The agent was trained using the DQN algorithm, continuously evaluating and optimizing its performance to enhance navigation efficiency. Throughout the training process, Matplotlib played a crucial role in visualizing both the agent's progress and key performance metrics, providing valuable insights into the agent's learning and adaptation over time.

2 Related Works

2.1 Introduction to Q-Learning

A comprehensive introduction to Q-Learning is provided by [4]. This resource offers an accessible explanation of Q-Learning, detailing the fundamental concepts such as the Bellman equation, action-value function, and the iterative process by which an agent learns the optimal policy. It is particularly useful for understanding the theoretical basis of Q-Learning that underpins more complex algorithms such as Deep Q-Networks.

2.2 Deep Q-Learning Overview

In a subsequent article [5], which extends the basic Q-Learning framework by incorporating deep neural networks. This resource explains how Deep Q-Networks use neural networks to approximate Q-Values and how techniques such as experience replay and target networks are employed to stabilize training. This resource is useful for understanding the specific enhancements and challenges associated with implementing Deep Q-Networks.

2.3 Application to Game Environments

The application of Deep Q-Networks to complex game environments is exemplified by [2]. This resource demonstrates how Deep Q-Networks can be used to train an agent to perform well in OpenAI's car racing game, highlighting practical aspects such as reward shaping, state representation, and training dynamics. This article provides insights into adapting Deep Q-Networks to different types of grid-based and continuous environments.

2.4 Introduction to Reinforcement Learning

An additional foundational resource is the article [6]. This article provides an overview of reinforcement concepts, including the exploration-exploitation trade-off, reward systems, and the role of agents and environments. This resource is valuable for understanding the basic principles and motivations behind reinforcement learning, which are essential for comprehending more advanced methods such as Deep Q-Networks.

2.5 Teaching AI to Play Games

The article [1] provides a broader perspective on using deep reinforcement learning for game playing. The author goes over various deep reinforcement learning techniques, including policy gradient methods and actor-critic approaches, offering a comprehensive overview of the strategies available for training game-playing agents. This resource is valuable for understanding the broader context and potential alternative methods to Deep Q-Networks.

2.6 Reinforcement Learning for the Snake Game

Finally, [3] presents a case study of applying reinforcement learning to a classic game that has a lot in common with the Santa Fe Trail environment. This resource details the implementation of Q-Learning and Deep Q-Networks to train an agent to play the snake game, emphasizing the practical challenges and solutions in state representation, reward design, and training stability. This article provides a practical example of applying reinforcement learning techniques to a well-known grid-based game, similar to the Santa Fe Trail Problem.

3 Method

In reinforcement learning, the training data for models is not pre-collected and static. Instead, the agent generates its training data through continuous interaction with its environment, which in turn influences the agent's future actions. This feedback loop enables the agent to iteratively improve its decision-making policy based on the rewards that it receives. The Deep Q-Network algorithm was chosen for this problem due to its effectiveness in handling environments with large state spaces and its ability to learn from high-dimensional sensory inputs directly, making it well-suited for the complex task of navigating the Santa Fe Trail.

The choice of libraries and frameworks for this problem was driven by their specific capabilities and suitability for the tasks at hand. Pygame was selected for environment simulation due to its simplicity and efficiency in creating 2D game graphics, making it an ideal choice for rendering the grid-based environment and providing real time updates of the agent's movements. PyTorch was selected for implementing the DQN model because of its flexibility and dynamic computation graph, which facilitates the development and training of complex neural network architectures. Matplotlib was chosen for data visualization because of its powerful plotting capabilities, which are essential for monitoring the agent's learning process over time.

3.1 Environment Simulation

The environment is a 32x32 grid representing the Santa Fe Trail. This grid contains 89 food pellets scattered along a path that the ant (agent) must follow to collect the food. Using Pygame, we created a visual representation of this grid, with the ant and food pellets displayed at specific coordinates. The ant's movement and the positions of the food pellets are updated in real-time, providing a dynamic setting for the agent to interact with. The environment is responsible for handling the ant's actions, such as moving straight, turning right, or turning left, and updating the game state accordingly. It also keeps track of the score and the number of time steps taken, which are critical for evaluating the agent's performance.

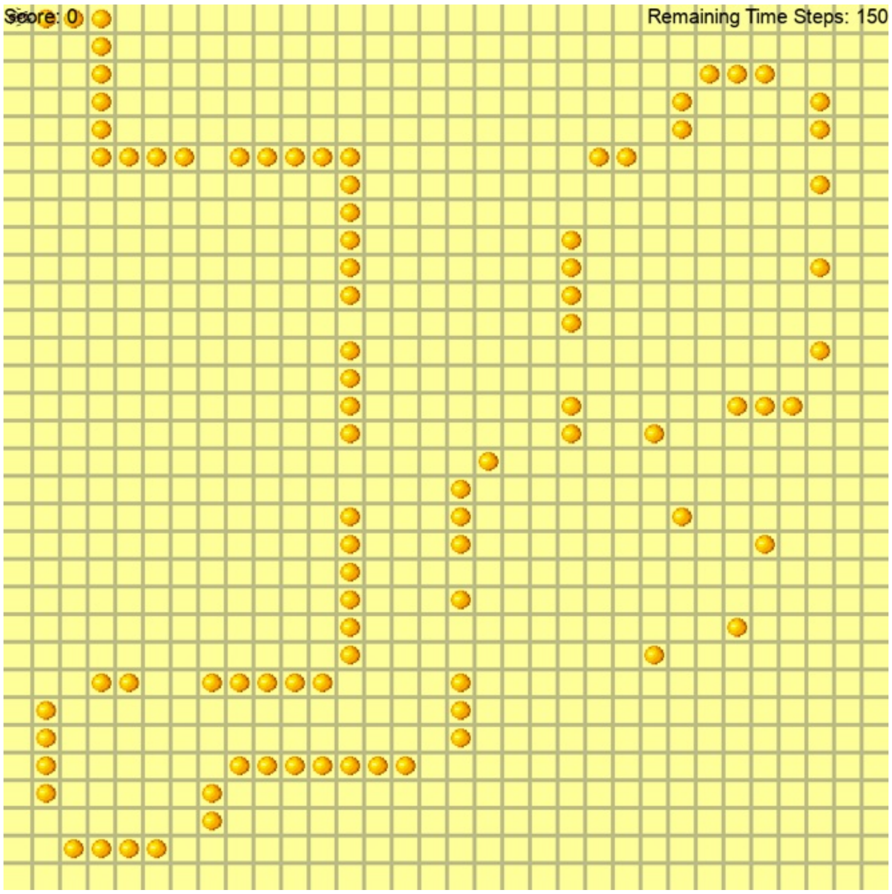


Fig. 1 Visual representation of the Santa Fe Trail environment used in the simulation.

3.2 Agent Behavior

The agent interacts with the environment and learns from these interactions. The agent's primary functions include deciding on the next action to take, remembering past experiences, and learning from them. The agent's decisions are influenced by an epsilon-greedy policy, balancing exploration (trying new actions) and exploitation (using known actions that yield high rewards). The agent decides on actions based on the Q-values predicted by the neural network model, choosing either the action with the highest Q-value (exploitation) or a random action (exploration) based on the current epsilon value. The agent collects data from its interactions with the environment, including the state before and after an action, the action taken, the reward received, and whether the episode is still running. This data is stored in the agent's memory and used for training the neural network model.

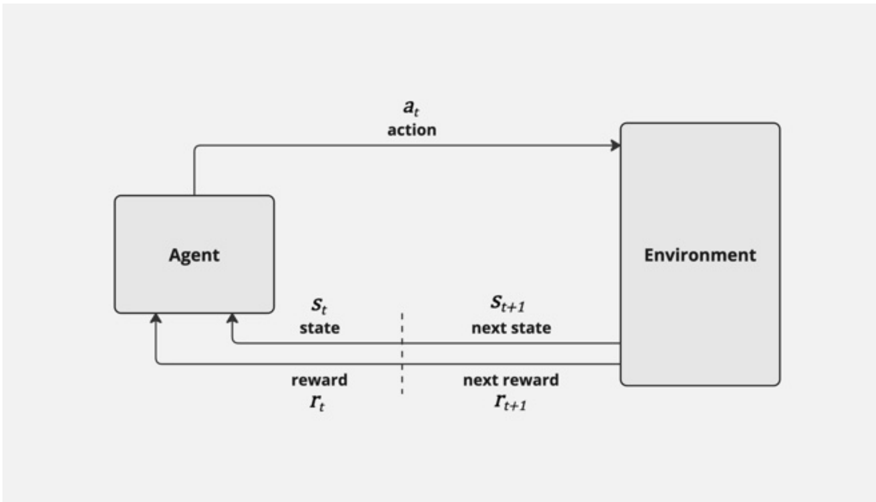


Fig. 2 Flowchart illustrating the training loop for the reinforcement learning agent.

3.3 Deep Q-Network Model and Training

At the heart of the system is the neural network model, implemented with PyTorch, which predicts the best actions to perform based on the current state of the environment. The neural network model is a basic feedforward neural network with three layers. The first layer is an input layer with eight nodes representing the state of the environment (a state vector with eight states is passed in as input). The second layer is a hidden layer with 256 nodes, chosen arbitrarily to provide sufficient capacity for the network to learn complex representations. The final layer is an output layer with three nodes, representing the Q-values for each of the possible actions: moving straight, turning right, and turning left. These Q-values represent the expected future rewards for taking each action. Each uses ReLU activation functions to introduce non-linearity and enable the network to learn complex patters. Training the model involves adjusting its parameters to minimize the difference between predicted Q-values and target Q-values, which are calculated based on the actual rewards received and the estimated future rewards. Additionally, the model is trained using experience replay, where the agent randomly samples a batch of experiences from its memory to update the network. This approach helps break the correlation between consecutive experiences, improving the stability of the learning process. A fixed Q-target network is used to calculate the target Q-values, which are updated less frequently than the main network to further stabilize training.

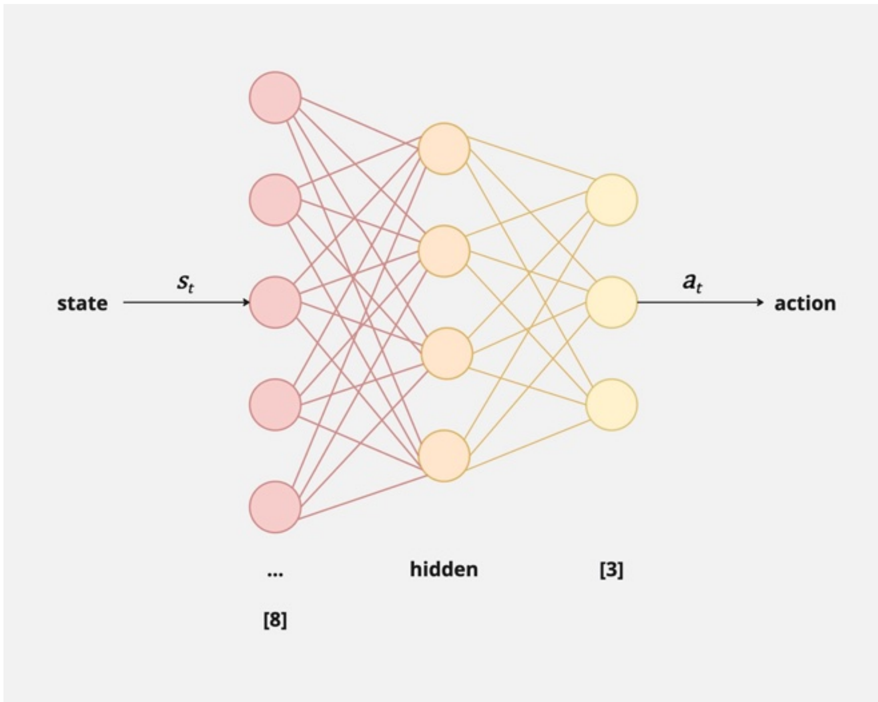


Fig. 3 Flowchart of the Deep Q-Network (DQN) model architecture.

3.4 Data Visualization

To monitor the agent's progress, a plotting system using Matplotlib was implemented to visualize the scores and time steps across multiple runs. This visualization helps in understanding how the agent's performance improves (or doesn't) over time. The scores plot shows the rewards the agent earns in each episode, as well as the average score across all episodes, while the time steps plot displays the number of steps taken to complete each episode, as well as the average time steps across all episodes. These plots include both the raw values and their moving averages, allowing for a clearer understanding of trends and the effectiveness of the learning process. This visualization is crucial for diagnosing issues and making adjustments to improve the agent's performance.

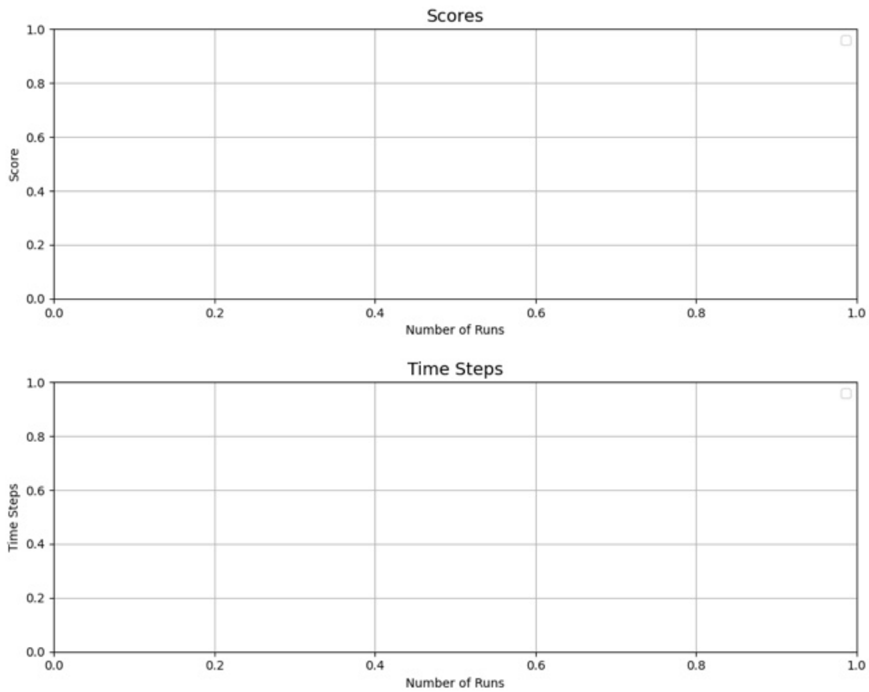


Fig. 4 Blank template for the training plot, illustrating the layout for visualizing the agent's scores and time steps during training runs.

4 Data

The effectiveness of the Deep Q-Network (DQN) model in navigating the Santa Fe Trail was assessed by collecting and analyzing data related to the agent's interactions with the environment. The core research question—whether a DQN can efficiently navigate a dynamic and unpredictable environment like the Santa Fe Trail—was explored through a series of experiments. The data collected provides insights into the agent's learning process, adaptation, and overall performance in meeting this challenge.

4.1 Data Collection and Experimental Setup

The primary data consists of the experiences generated by the agent while navigating the environment. Each experience, or tuple, is composed of the following elements:

- **State (S):** The current state of the environment, represented as a vector of features describing the agent's position and the relative positions of the food pellets.
- **Action (A):** The action taken by the agent in the current state. Actions include moving straight, turning right, and turning left.
- **Reward (R):** The immediate rewards received after taking the action. Rewards are designed to encourage the agent to collect food pellets and complete the trail efficiently.
- **Next State (S'):** The resulting state of the environment after the action is taken.
- **Run:** A boolean value indicating whether or not the episode has ended.

The training process involved multiple runs, where each run represents a complete episode from start to finish. Data collected across these runs were used to evaluate the agent's performance over time.

4.2 Performance Metrics

Key performance metrics were gathered, including:

- **Scores:** The cumulative rewards earned by the agent in each episode, directly reflecting the agent's efficiency in collecting food pellets.
- **Time Steps:** The number of actions taken to complete each episode, which is indicative of the agent's speed and path optimization.
- **Average Scores:** The mean of the cumulative rewards across all episodes, providing a measure of overall performance.
- **Average Time Steps:** The mean number of actions taken per episode, reflecting the agent's typical path efficiency.

The central question—how effectively the agent navigates the Santa Fe Trail—is directly measured by these metrics. An ideal agent would maximize its score (by collecting as many food pellets as possible) while minimizing the time steps required to do so.

4.3 Results and Interpretation

The results across different stages of training are summarized in Figures 1 through 6. These figures illustrate the agent's performance after varying numbers of runs, providing a clear picture of the learning process:

1. **Early Stages (0-50 runs):** The agent exhibits inconsistent performance with high variability in scores. This phase reflects the agent's exploratory behavior as it learns the environment and possible strategies. The time steps consistently reach

the maximum (150), indicating that the agent is still inefficient in optimizing its path.

2. **Mid Stages (50-150 runs):** The agent's performance begins to stabilize, with noticeable improvement in scores. Although the agent starts to exploit more efficient strategies for collecting food pellets, the time steps remain at the maximum (150), suggesting that while the strategy is improving, it is not yet optimized for speed.
3. **Later Stages (150-250 runs):** The agent demonstrates a significant and sustained improvement in performance, as indicated by a flattening learning curve. However, the agent consistently uses the maximum allotted time steps (150), showing that while its strategy for collecting food pellets has become efficient, there remains room for optimizing the speed of task completion.

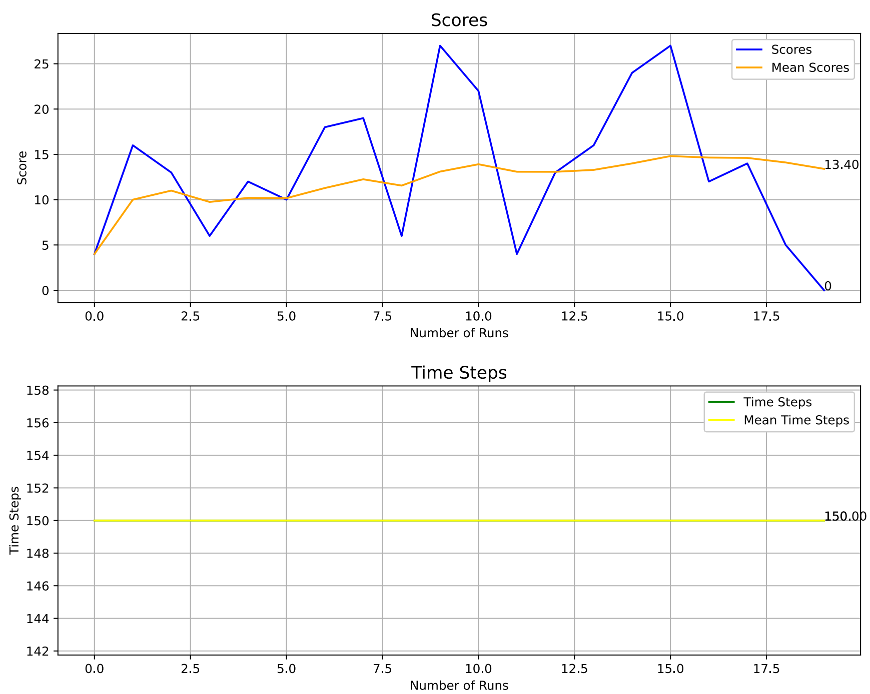


Fig. 5 Santa Fe Simulation showcasing the agent's performance after 20 runs. The agent is still in the early stages of learning, with noticeable fluctuations in performance as it explores different strategies.

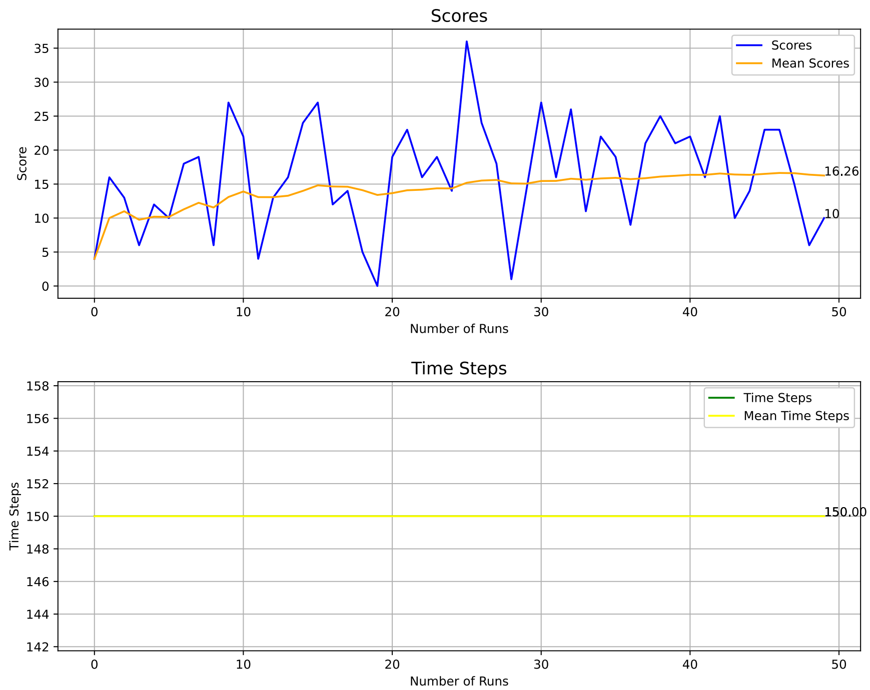


Fig. 6 Santa Fe Simulation showcasing the agent's performance after 50 runs. The agent begins to show signs of learning, with more consistent performance and higher scores.

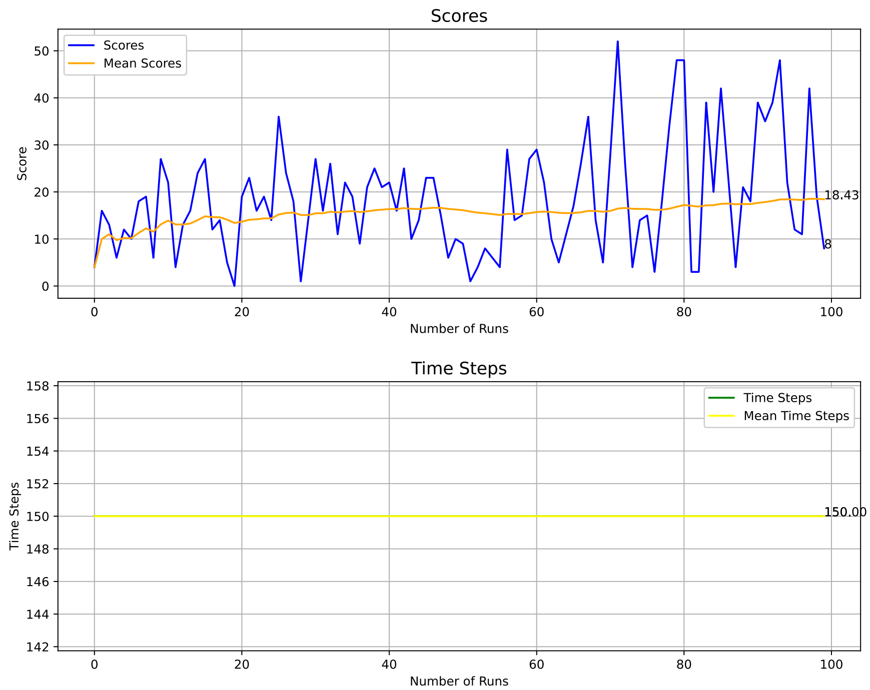


Fig. 7 Santa Fe Simulation showcasing the agent's performance after 100 runs. The learning curve is becoming more stable, and the agent's strategies are more refined.

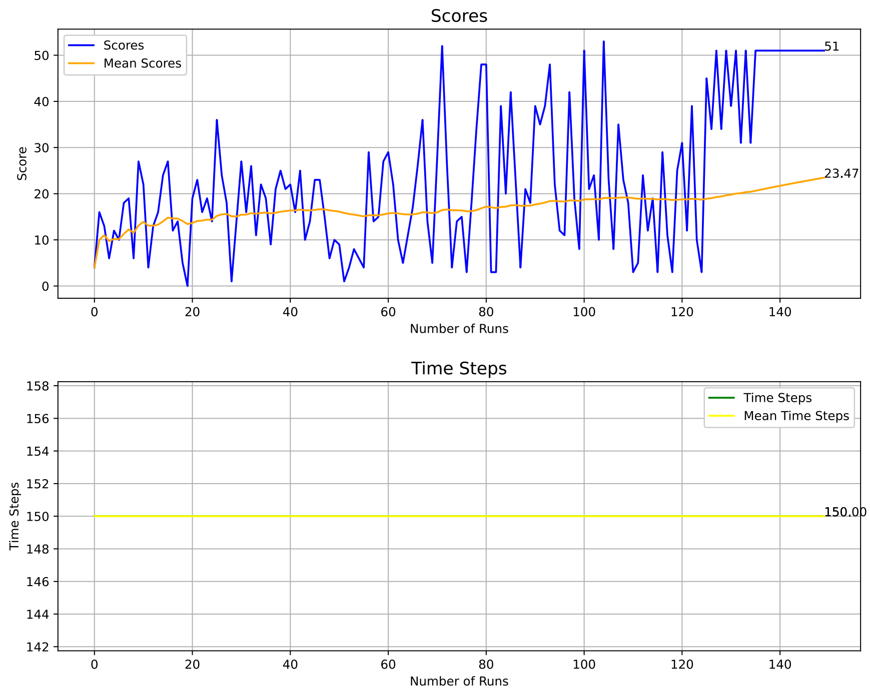


Fig. 8 Santa Fe Simulation showcasing the agent's performance after 150 runs. The agent demonstrates significant improvement, with higher cumulative rewards and more efficient paths to collect food pellets.

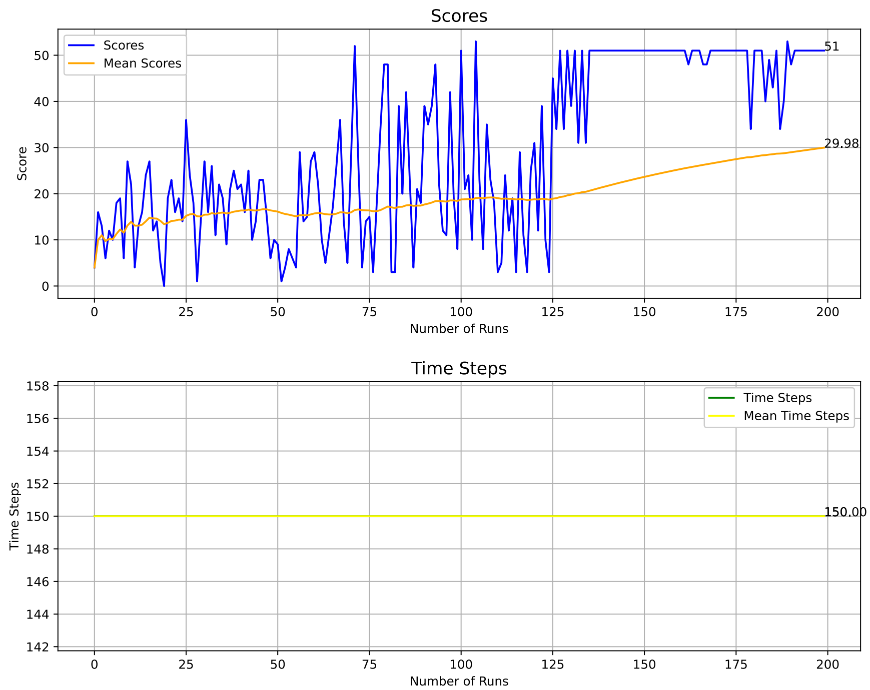


Fig. 9 Santa Fe Simulation showcasing the agent's performance after 200 runs. The performance has further stabilized, indicating a more reliable and effective learning process.

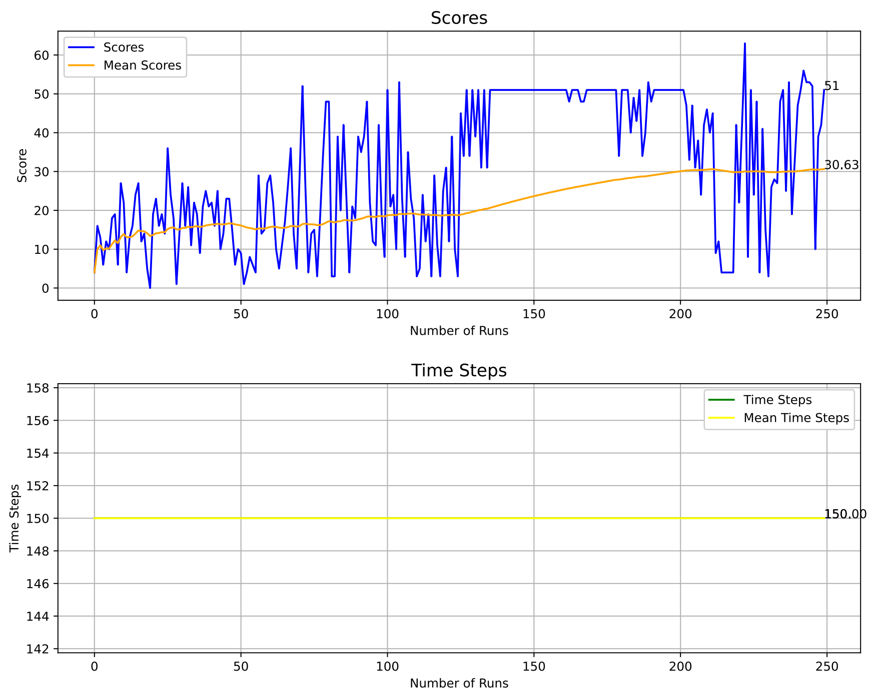


Fig. 10 Santa Fe Simulation showcasing the agent's performance after 250 runs. The agent's learning curve appears to plateau, suggesting that it has reached a near-optimal strategy for the given environment.

4.4 Linking Results to the Research Question

The results affirm that the DQN model is capable of handling the complexities of the Santa Fe Trail environment. The consistent improvement in scores indicates that the agent learns to navigate the trail more effectively over time. However, the persistent use of the maximum time steps highlights a limitation in the current model—while the agent becomes proficient at finding food, it does not yet optimize for speed. This suggests that further refinement of the model or the reward structure may be necessary to fully achieve the research goal of minimizing both time and maximizing efficiency.

5 Conclusion

The implementation of a Deep Q-Network (DQN) for the Santa Fe Trail Problem has demonstrated the efficacy of reinforcement learning techniques in a controlled, grid-based environment. This approach effectively uses feedback from the environment to iteratively improve an agent's performance, showcasing the potential for such models in solving complex decision-making problems.

5.1 Applications

The methods and findings from this reinforcement learning implementation extend beyond the Santa Fe Trail Problem to various fields. In robotics, the dynamic nature of reinforcement learning algorithms can be applied to autonomous navigation, where agents learn to operate in complex, unstructured environments. Similarly, in game development, reinforcement learning can create intelligent and adaptive non-player characters (NPCs), enhancing player engagement and experience. In industrial settings, reinforcement learning can optimize processes like inventory management by enabling agents to make informed decisions based on real-time data. The healthcare sector can benefit from these techniques by developing personalized treatment plans, where agents adjust strategies based on patient responses and outcomes.

5.2 Concerns and Next Steps

While the implementation has shown promising results, several concerns need to be addressed. One primary issue is the scalability of the model. As the complexity of the environment increases, the computational resources and time required for training the agent also increase. Another concern is the robustness of the agent's performance. The agent might perform well in the specific environment used for training but may not generalize well to different or more complex environments. To address these concerns, future work will focus on enhancing the model's scalability and generalizability. This can be achieved by exploring more sophisticated neural network architectures and incorporating advanced techniques such as transfer learning and curriculum learning. Additionally, experimenting with different reward structures and action spaces can help in fine-tuning the agent's behavior. Future steps also include applying the developed methods to real-world problems in robotics and autonomous systems, testing the agent's performance in various simulated and real environments, and integrating additional sensory inputs to improve the agent's decision-making capabilities.

Acknowledgements The work was supported by the Daimler Truck Portland Professorship. Special thanks are due to Dr. Christof Teuscher for his invaluable guidance and support throughout the internship. His expertise and encouragement played a crucial role in the successful completion of this research.

References

1. Mauro Comi. How to teach an AI to play Games: Deep Reinforcement Learning. *Medium*, June 2022.
2. Ali Fakhry. Applying a Deep Q Network for OpenAI's Car Racing Game. *Medium*, July 2021.
3. Nancy. Teaching an AI to Play the Snake Game Using Reinforcement Learning! *Medium*, January 2023.
4. Jeremi Nuer. Q-Learning — An Introduction. *Medium*, December 2021.
5. Jeremi Nuer. A Comprehensive Guide to Deep Q-Learning. *Medium*, May 2022.
6. Marco Del Pra. Introduction to Reinforcement Learning. *Medium*, November 2020.

Innovations in Educational NLP: Creating a Syllabus-Specific LLM

Iris Ta

Abstract The development of machine learning models grows every day, but how can machine learning help the educational system? It may seem contradictory to implement artificial intelligence in education, but with the right ethical practices, It can be extremely beneficial for students and teachers. Students who use artificial Intelligence models can promote efficient learning while teachers can strengthen their course teachings for the learners. I created a language model that assists students and professors with syllabus management to promote efficiency and optimize syllabus content. Using various machine learning libraries and Python, I trained a model that achieves 99.907% accuracy. This application can summarize key sections of an uploaded syllabus and compare it against hundreds of other syllabi. With modifications, this application has the potential to assist individuals across different fields with document management.

1 Introduction

Large language models are gaining popularity in the tech world. Educational figures, including students and professors, can reap significant benefits from a specialized syllabus model that promotes efficiency and transparency in the classroom. With the rise of the digital age, most high schools and colleges now post their syllabi online. While many professors rely on the honor code, expecting students to read the course syllabus, others use mandatory quizzes on syllabus content. I aimed to create an application that benefits both students and professors in managing syllabus content. For students, this application can summarize their uploaded PDFs of syllabi and highlight the differences between the syllabus of one class and another. Professors can compare their own syllabus with hundreds of others, which can help them identify missing key components or streamline text on certain topics.

Iris Ta
Portland State University, 1900 SW 4th Ave 500, Portland, OR 97201, e-mail: irista244@gmail.com

2 Related Work and Background

LLMs, or Large Language Models, are used almost daily by most people. Whether it be Google Maps, or customer service chatbots, LLMs are one of the largest technological advancements currently [2]. They are paving the way for a future full of technology. with advanced natural language processing like question and answer queries or con- In summarization, the possibilities are endless. With more basic NLP being tasks like tokenizing (identifying single words and/or punctuation) to more complicated tasks like identifying grammatical errors within preprocessing (breaking up text into parts) and fixing it [3].

But how do large language models fit into the educational aspect of society? LLMs can provide students with personalized learning paths while also assisting educators in catering to everyone's needs [1] making artificial intelligence seem like a no-brainer to most. Since artificial intelligence is still developing, many questions the ethical concerns of its use in the classroom. One of the main problems that affect artificial intelligence is bias since if models are trained on biased datasets, The results will come back biased as well, furthering singling out students coming from marginalized backgrounds [4]. With all these concerns, it is important to see why LLMs can assist the educational world. Many students who use some sort of Interactive technology in their educational life can benefit from the personalization. that technology offers them [5]. Whether it be choosing the pace at which you learn at or comprehending a topic in a style that fits your needs, it is hard to look away. from that.

3 Data and Methodologies

This section will cover the techniques I applied to process data from my large language model.

3.1 Data Collection

Data was collected in various ways and converted to fit the standards of a large language model. This section outlines the techniques used to process and utilize the data from the model. To obtain the necessary syllabi, I reached out to Professor Teuscher for guidance on accessing PSU syllabi.

Professor Teuscher was able to gather syllabi from numerous past professors and provided me with an extensive zip file containing hundreds of syllabi. These syllabi covered a wide range of subjects, from health and happiness to discrimination in Portland. The zip files were extracted and consolidated into a single large folder for easier accessibility and modifications.

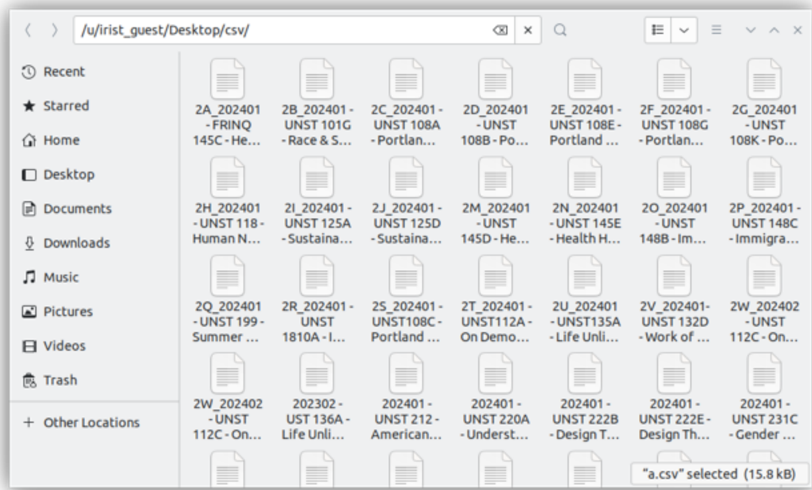


Fig. 1 The CSV files that were later converted into one large CSV file.

Please do not use quotation marks when quoting texts! Simply use the `quotation` environment – it will automatically be rendered in line with the preferred layout.

3.2 Methods

This subsection will cover the development stages of creating my large language model.

3.2.1 Development for Professor Usage

The development of the syllabus feature for professors involved several stages.

The backend was coded using Python (version 3.12.0) on the IDE VSCode. I began by converting all the PDFs into CSV files and then combined these smaller CSV files into a single large CSV file using the Pandas and Py2PDF libraries. I identified the most common sections within each syllabus and compiled a list of the top twenty sections typically found in a syllabus. The most common sections included instructor name, course description, academic integrity statement, and others. Using these keywords, I sorted through the large CSV file to locate instances of each keyword and created a new CSV file with three columns: 'Label' for the keyword, 'Text' for the full statement containing the keyword, and 'Summary' for a shortened summarized version of the 'Text' column.

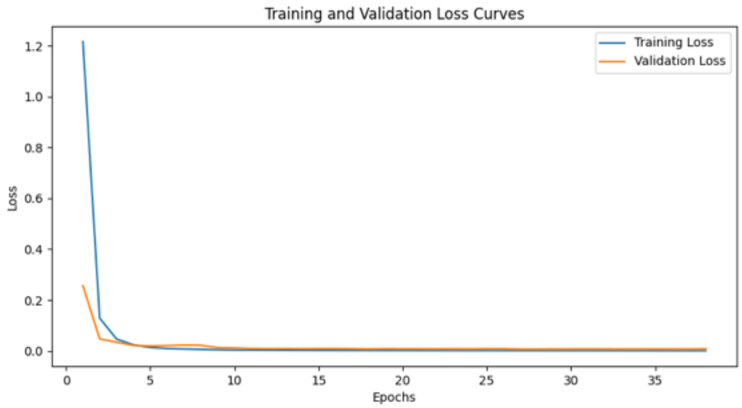


Fig. 2 The Training and Validation Loss Curves with Loss on the Y-Axis and the number of Epochs on the X-Axis.

Once the data was preprocessed, it was split into training, validation, and testing sets. The model was then built and trained using machine learning libraries such as PyTorch for model creation, Sklearn for data preprocessing, DistilBertTokenizer for tokenizing the text, Gradio for testing the question-and-answer feature, and TQDM to monitor progress after each epoch during training. After a few training sets, I noticed my loss was not optimized. I ended up cleaning up my data using PDFPlumber to ensure that there were no odd space issues between characters and that words were parsed correctly. After implementing those changes, I achieved an optimal training loss curve.

The final stage involved connecting the backend to the frontend using Flask. App routes were created for summarizing every instance of a keyword and finding every instance of a keyword. The Flask server ran parallel to the front-end application for local operations. For front-end development, I researched the most suitable applications for my needs and chose React.js, incorporating CSS, HTML, and JavaScript. The majority of the front-end work involved using JavaScript to display components from the back end, including the keyword summary input for professors and the keyword output summary.

3.2.2 Development for Student Usage

The development of the student's functionality was similar to that of the professor's, as it utilized similar machine learning libraries and coding languages.

The back-end development used Python (version 3.12.0) on the IDE VSCode. Since the student functionality focused on summarizing the uploaded syllabus, the PDFs did not need to be trained on my LLM. The LLM was not well-suited for summarization tasks, so I opted to use a pretrained model, specifically Hugging

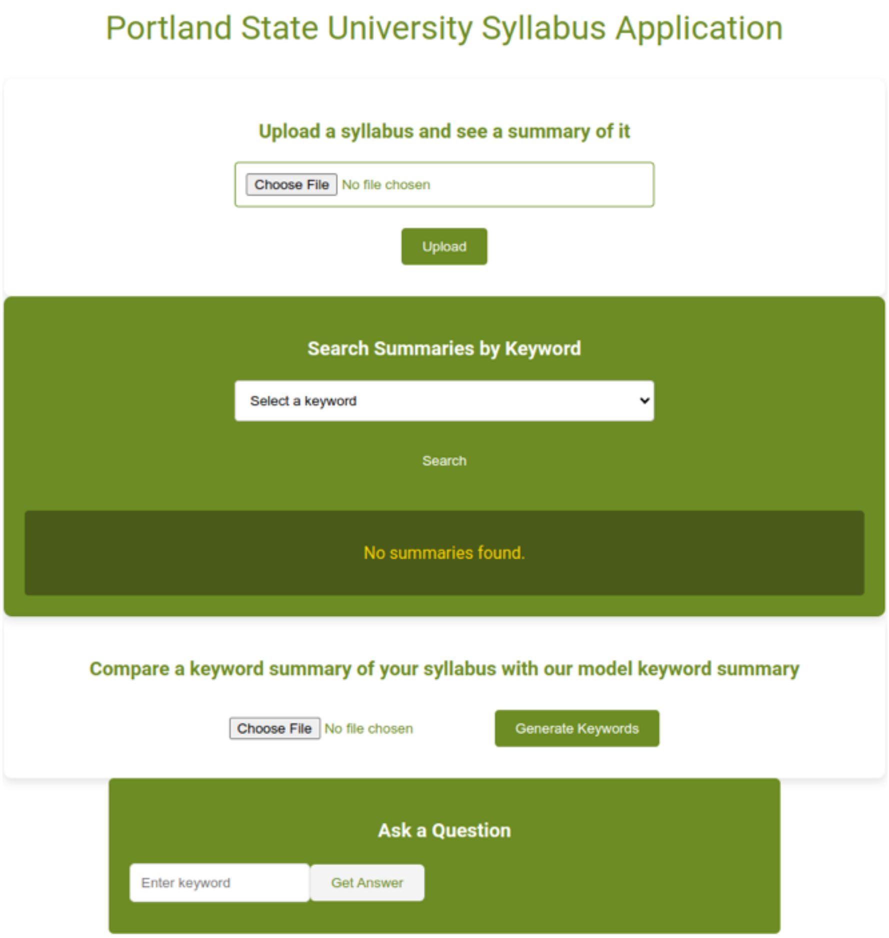


Fig. 3 The Frontend of my Application that can summarize the many syllabi it is trained on.

Face's pipeline summarizer, for these tasks. The LLM model was utilized for keyword identification and sentence separation within the PDFs.

In terms of machine learning library usage, I employed many of the same libraries as in the professor's development: PyTorch for invoking the saved trained model, SkLearn, and DistilBertTokenizer for preprocessing and cleaning the text. Additionally, I used other libraries such as Pandas for reading the CSV file format, io and re for handling input and output, and PDFPlumber for cleaning differently formatted syllabi. To store the text from the uploaded PDFs, SQLite3 was used instead of a CSV file, as it is easier to manipulate and manage data in a database than in a CSV file.

4 Results

This section will cover the results of both the professor side and student side. Subsection 1 is dedicated to the findings for the professors and the subsection 2 will cover the findings for the students.

4.0.1 Results for Professors

The leading question this project tried to answer is how can large language models assist professors in the educational field. After much trial and error between the communication of the backend and the frontend, an application was created. Professors were ultimately able to upload their own created syllabus and compare it to hundreds of other syllabi. This model helped professors identify missing key sections within their own syllabus and how their syllabus compares to others in various sections—from course descriptions to office hours. They were also able to see an elongated summary of certain key sections and shortened key sections.

4.0.2 Results for Students

Students being the main audience to this model can utilize it in many ways.

Students can apply the professor side by uploading a syllabus from a certain class they have and comparing their uploaded syllabus with hundreds of other syllabi, but students main use comes from the fact that the model can accurately summarize key parts of their uploaded syllabus. These functions can assist students in preparing for syllabi quizzes, understanding a long syllabus, and overall promoting efficiency for students with many classes.

5 Conclusion

My application was a success, but there are still many modifications that can be made to make the application more useful to educators and students alike. Though both parties benefit from using the application, with more additions, the model could be used outside of just syllabus management.

5.1 Applications

Professors using this application can reap the reward of comparing their own created syllabus with many others to ensure that their syllabus meets not only their educational standards but also institutional standards. Students, on the other hand, can summarize their syllabi in hopes of getting a better understanding of what the course covers and what the professor values enough to highlight in their syllabus. For a more broad point of view, this application can be useful for administrators at PSU so they can see trends within their school's curriculum and make necessary changes based on what professors emphasize to students taking their course. This could dive deeper into cross-referencing mountains of syllabi across different institutions to truly see the difference between what each university values to put in their syllabus.

5.2 Concerns and Next Steps

Currently my application is trained on a plethora of PSU syllabi from various subjects, but it may not be entirely representative of all the syllabi that PSU has. With access to more syllabi and hours to train another model, my application could better represent PSU's syllabus collection as a whole. I plan to dive deeper into my application with the goal of having professors upload hundreds of syllabi and letting the application train them and summarize the key sections just how they can do with one. This could learn further into professors being able to cross-reference the school's overall syllabus trends with other schools. I also plan to explore the bias in my model since a lot of the preprocessing within the model was done by hand and I could have easily missed some extra spaces.

Acknowledgements Thank you Professor Christof Teuscher for this wonderful opportunity and your underlying support and knowledge throughout this process. Thank you to the undergraduate and graduate students as well as the other student interns for their support throughout this project.

References

1. Tariq Alqahtani, Hisham A Badreldin, Mohammed Alrashed, Abdulrahman I Alshaya, Sahar S Alghamdi, Khalid bin Saleh, Shuroug A Alowais, Omar A Alshaya, Ishrat Rahman, Majed S Al Yami, et al. The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research. *Research in Social and Administrative Pharmacy*, 19(8):1236–1242, 2023.
2. Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.
3. Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.
4. Andy Nguyen, Ha Ngan Ngo, Yvonne Hong, Belle Dang, and Bich-Phuong Thi Nguyen. Ethical principles for artificial intelligence in education. *Education and Information Technologies*, 28(4):4221–4241, 2023.
5. Long V Nguyen. Technology-enhanced efl syllabus design and materials development. *English Language Teaching*, 1(2):135–142, 2008.

SunMaps: Sustainble Urban Planning Insights for Phoenix

Sofia Vargas and Ishita Deshpande

Abstract

Phoenix, Arizona, often dubbed as ’America’s hottest city’, is notorious for its extreme climate. We have created a web application catered towards Phoenix urban planners to help build cooling infrastructure for the city. Utilizing the Python library Folium and datasets from Arizona State University, we displayed local climate zones (LCZ), urban zoning types, and mean radiant temperatures (MRT) on a map of metro Phoenix at 1000m spatial resolution. LCZs are split into 7 urban zones and 7 vegetation zones. Within each LCZ, we calculated the mean and distribution of MRTs. When a user clicks on any location in the map, they are given the local LCZ, local MRT, and local zoning type. Based on this data, we developed an algorithm to recommend what cooling infrastructure is needed to reduce the MRT. The options for cooling infrastructure is based upon research papers from Arizona State University, and include solutions such as trees or permeable pavement.

1 Introduction

Phoenix is no stranger to extreme climate. Widely considered as the hottest city in the U.S., the city’s summer temperatures spill over a hundred degrees on a regular basis, making it difficult for Phoenix residents to not just walk in the open but even function at all. Science American’s article ”America’s Hottest City Is Having a Surge of Deaths” describes the effects of Phoenix’s heat, stating that 2023 hit a record-breaking 645 heat-related deaths, the highest in Arizona. [5]

In the face of extreme climate, Phoenix and Maricopa County have made significant efforts to combat the heat and thus make the region more livable. A handful

Sofia Vargas
Sofia Vargas, Arizona State University, e-mail: sofiamyrav@gmail.com

Ishita Deshpande
Ishita Deshpande, Arizona State University e-mail: ideshpa2@asu.edu

of effective cooling infrastructure tactics have been introduced, whether it be artificial shade, reflective pavement, or tree canopies. Furthermore, there have also been projects mapping Phoenix's heat vulnerability, temperatures, and climates.

Our goal is to draw from these efforts and create a program that will show which cooling infrastructure is best for which type of scenario. The resulting project was an interactive map that allows residents to predict what changes they can make to their residential areas to make a difference in their local climate. For example, a user should be able to zoom into their location on the map and receive a detailed description of their current climate, including factors such as humidity and temperature, alongside a suggestion to plant a few trees in order to reduce the CO2 levels in their area by a certain percentage or implement cooling pavements to reduce surface temperature.

2 Related Work and Background

2.1 Background

SunMaps mainly builds upon the knowledge of Local Climate Zones (LCZ)[4] and Mean Radiant Temperature [3]. Mean radiant temperature (MRT) is a measure of how radiant heat is experienced by humans, or rather a measure of how heat affects personal comfort. Its consideration for personal comfort is the main reason we decided to use MRT, particularly instead of air temperature which does not consider factors such as the heat from other nearby surfaces. [3]

LCZs are categorized based on factors of concrete and pavement, building heights, construction materials, and vegetation height. Most importantly, LCZs are not based on factors of temperature, and as such each LCZ has its own distribution and range of MRTs.

2.2 Related Work

The papers we searched for initially were mainly centered on urban heat influenced by socioeconomic conditions. A paper on Portland Urban Development had utilized NASA Earth Observations to overlap environmental and social vulnerabilities for the city, creating a heat vulnerability index to identify where infrastructure would be most effective. The specific infrastructure they were considering was depaving, which is replacing pavement with greenspace [6].

Another paper had also mapped heat vulnerability, this time focusing on New York City. Ocellus XR, a mixed reality program, visualizes heat data on an interactive map of New York City that also compounds socioeconomic data to identify opportunities where the city can adapt to climate change [11].

When we later decided to focus more specifically on just urban heat infrastructure, we relied on a paper by Arizona State University detailing the effect of the built environment and heat exposure on the average person's pedestrian journey. They ran two separate simulations, one which prioritized making the built environment cooler and one which prioritized the traveler's behavior, whether it be taking shorter routes or limiting heat exposure[8]. This paper used the method of combining LCZ with MRT data, which we eventually adopted for our project.

In addition to the foundational work on heat vulnerability and urban infrastructure, recent studies have provided more targeted insights into the effectiveness of specific cooling interventions in urban environments. One such study, ”Modelling the impact of increased street tree cover on mean radiant temperature across Vancouver's local climate zones” [10] by Aminipouri et al. (2019), explored the potential for increased street tree coverage to reduce mean radiant temperature (MRT) in Vancouver, Canada. The study utilized the SOLWEIG model to simulate the cooling effects of a one percent increase in street tree cover across different local climate zones (LCZs). The findings indicated that tree cover had a significant impact on reducing MRT, particularly in lower-density residential areas. The study's method of integrating LCZs with MRT data and its focus on tree cover as a cooling strategy provided a valuable framework that aligns closely with the goals of our project in Phoenix, where similar urban heat mitigation strategies are being considered.

Complementing this, the study ”50 Grades of Shade” [1] by Middel et al. (2021) conducted in Tempe, Arizona, closely examined the effectiveness of various shade types, including natural shade from trees, engineered structures, and urban forms, in mitigating heat. The study utilized biometeorological data to quantify the cooling performance of these shade types and developed characteristic shade performance curves. The research highlighted the significant variations in cooling potential across different shade types and urban contexts, underscoring the importance of selecting the right shade infrastructure for specific urban environments. The study's focus on urban form and engineered solutions, alongside natural shading, and provides a comprehensive view of the potential cooling strategies available to urban planners in hot, arid regions like Phoenix.

Furthermore, the study ”Effects of white roofs on urban temperature in a global climate model” [7] by Oleson et al. (2010) examines the effectiveness of cool roofs as a strategy to mitigate urban heat islands. This research assesses the impact of increasing roof albedo globally using an urban canyon model coupled with a global climate model. The study found that increasing roof albedo can reduce the urban heat island effect by lowering daytime maximum temperatures more significantly than nighttime temperatures. However, it also notes that the effectiveness of this strategy varies by region, with reduced benefits in high-latitude areas during winter due to increased heating demands. These findings suggest that cool roofs can be an effective component of urban cooling strategies in Phoenix, particularly given the city's high solar radiation and need for cooling during hot summer months.

These studies collectively informed the development of our web application, which integrates LCZ and MRT data to provide actionable insights for urban planners in Phoenix. By leveraging the methodologies and findings from these research papers, our application not only maps the current heat landscape of Phoenix but also offers evidence-based recommendations for cooling infrastructure tailored to the unique characteristics of each LCZ. Our project aims to build upon this existing body of work, providing a practical tool that makes data-driven decisions more accessible in the fight against urban heat.

3 Data and Methodologies

3.1 Data Collection

3.1.1 Local Climate Zone Data

Our local climate zone data was provided by a research paper by Arizona State University's heat resilience team,"Urban forestry and cool roofs: Assessment of heat mitigation strategies in Phoenix residential neighborhoods", [2] in which climate zones were mapped on a 1000m spatial resolution. There are fourteen local climate zones in total, which includes seven urban zones and seven vegetation zones. The zone definitions themselves were adopted from a research paper detailing the local climate zones of Washington. As stated before, these zones are determined on factors including construction materials, a ratio of how much sky is visible, width between buildings or trees, and zone purposes.

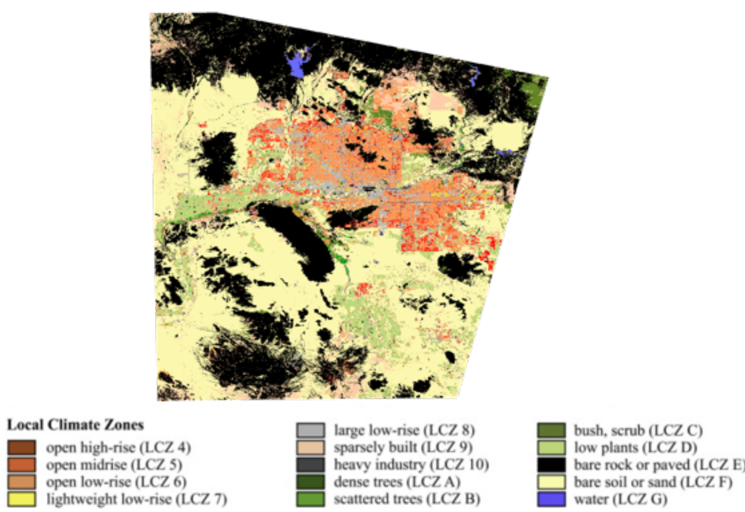


Fig. 1 Map of Local Climate Zones with Legend

3.1.2 Mean Radiant Temperature Data

To obtain a more accurate representation of the thermal environment, we utilized mean radiant temperature instead of regular air temperature. Our mean radiant temperature dataset came in the form of a GEOTIFF file provided by the same ASU heat resilience research team, with mean radiant temperature measured at a 1m spatial resolution. While we originally wanted to preserve such detailed spatial resolution, the file proved too large, and instead we converted the file into a JSON containing the mean radiant temperature data at a 100m spatial resolution.

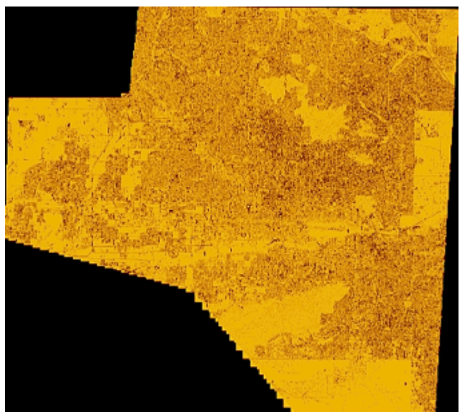


Fig. 2 Phoenix mapped by Mean Radiant Temperature

Mean radiant temperature is a measure of how an individual experiences heat. As such, the values of MRT can significantly deviate depending on whether the point was measured in the shade or in the sun. Simple changes to shade can result in an MRT reduction of around twenty degrees, despite the air temperature not varying as much. To make the variations in MRT more visible to users, we edited the GEOTIFF picture so that it would have higher contrast and saturation levels.

3.1.3 Zoning Data

Our zoning data came from City of Phoenix Open Data /citezoning, which is a website accessible to the public and contains a wide range of city data. The city is divided into eight zones: commercial, industrial, SF (single-family) residential, MF (multi-family) residential, planned unit development (PUD), downtown code, county, and non-developable. For our purposes, we combined the SF, MF, and PUD zones into one general residential category. Additionally, we did not factor in county and non-developable zones as there are too little areas designated as these zones.

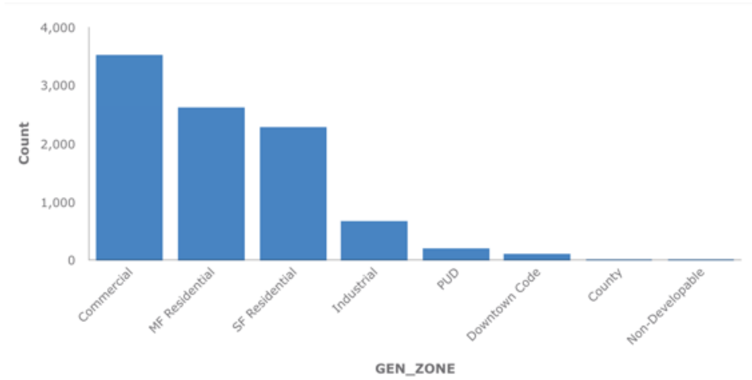


Fig. 3 Histogram of Different Zoning Type occurences

3.1.4 Infrastructure Data

Finally, our infrastructure data was gathered from multiple research papers (see related work). We had decided on five different types of infrastructure – artificial shade, tree canopy, cool roofs, reflective pavement, and water misters. These five infrastructure types have all been implemented to some capacity in Phoenix, and have research papers discussing their effectiveness, which we used in developing our own algorithm. These five infrastructure types were also all the most simple to implement, without requiring massive amounts of planning and infrastructure changes, while also functioning as the most effective methods of heat mitigation.

3.2 Methods

3.2.1 User Interface

Our user interface was a simple combination of Python, Javascript, HTML and CSS. We relied on the Folium and Geopandas libraries to create our interactive map as well as the image overlays of our data. Javascript was used to generate a popup displaying all relevant climate info. Finally, the map, side menu, and a tab listing our cited sources was added through HTML and CSS. The user interface retrieves its data from the decision tables via a server created through React.JS.

3.2.2 Decision Tables

We created an algorithm in Python based on five decision functions for each of our five infrastructure solutions. We first utilized the Numpy and MatPlotLib libraries to generate the distributions of MRT values for each Local Climate Zone, to which we used the mean for each climate zone in our functions. Each function has three parameters – the LCZ, the MRT, and the zoning type. With this combination of parameters, we used a series of if/else statements to determine the capacity at which each infrastructure would be used. For example, in our tree canopy function if the input MRT was higher than the mean MRT for the input LCZ, we would suggest a higher increase in tree canopy percentage for that area. The function's output would come in the form of a string message, recommending whether the infrastructure is good for this environment and its predicted MRT reduction.

4 Results

Users are able to navigate through the interactive map of Phoenix, and can choose to see a visual overlay of LCZs, MRTs, and zoning.

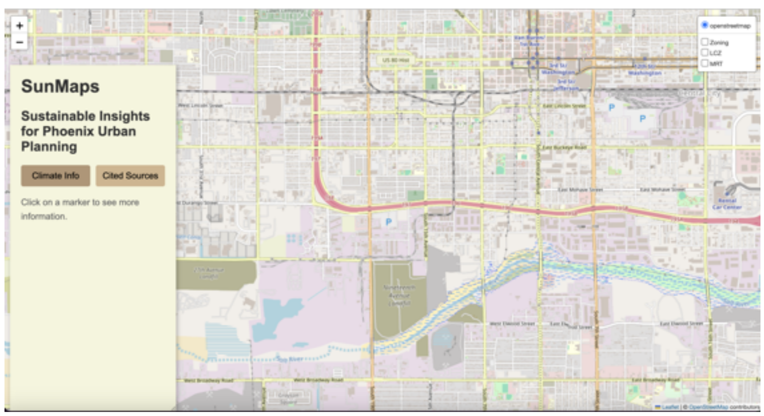


Fig. 4 Interface Home Screen, no data overlayed

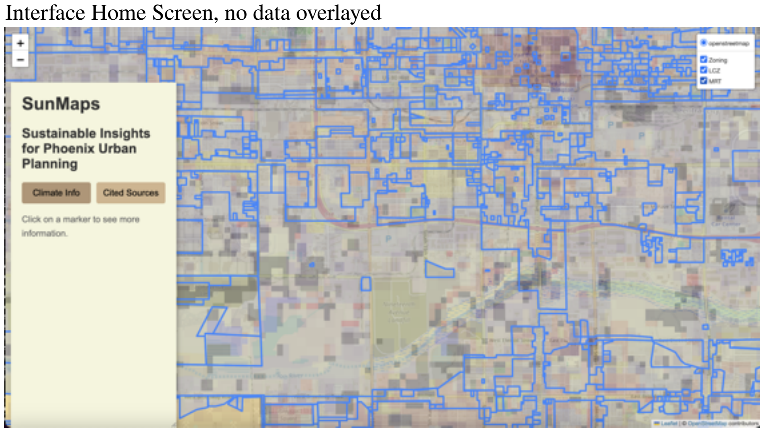


Fig. 5 Interface Home Screen, data overlayed

The user can then click on a location on the map, and a popup will generate showing the location's coordinates. An accompanying side menu will display the location's LCZ, MRT, and zoning type, along with a list of solutions to use and its projected MRT reduction. Finally, users may also navigate to the 'cited sources' tab, where they can see the papers with which we gathered our data from.

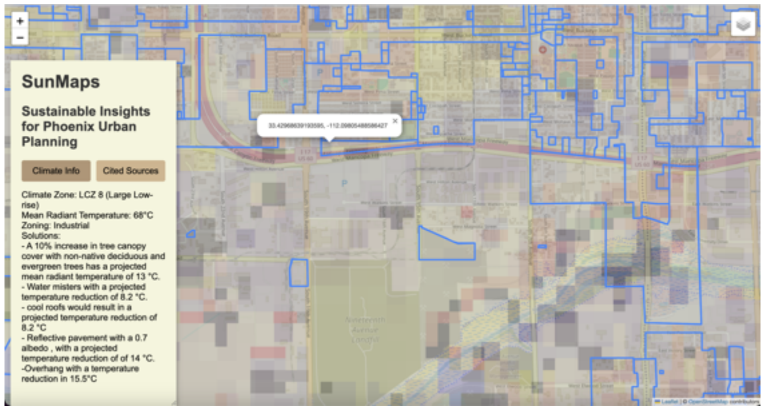


Fig. 6 Example Popup

In the process of making our decision functions, we also created 14 histograms for the different MRT distributions of each LCZ. An example distribution is shown of LCZ 6, the most common LCZ in Phoenix.

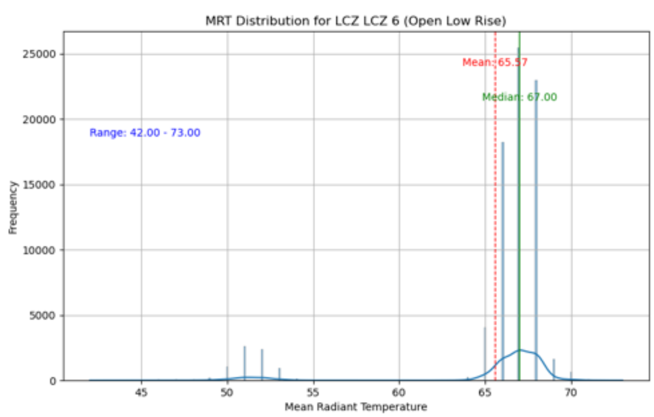


Fig. 7 MRT Distribution of LCZ 6

The distributions show just how widely MRT values deviate from conventional air temperatures; most LCZ distributions have mean values within the 65 to 68 degrees celsius range.

5 Conclusion

5.1 Applications

Our project is mainly geared towards those implementing cooling infrastructure, namely urban planners and the city government. Rather than having urban planners look at zoning, heat and climate research separately, our application should provide easier insight and access into what would work best. However, we also hope it can be used as a general awareness tool for any resident in Phoenix who wants to know how they can combat heat. Having residents generate attention about what infrastructure they want in their local community will help these changes actually get implemented, rather than just be suggested on a website.

Heat mitigation strategies in urban planning are very important to ensuring that excess energy consumption in urban environments is curtailed. Residents and urban workers are proven to be in dramatically better health and conditions when residing in areas that make use of heat mitigation strategies [9]

5.2 Concerns and Next Steps

5.2.1 User Interface

One small feature we would like to add in the future is a way for users to search up specific locations they would like to look at. As of now, they can only find locations they want by manually scrolling and/or zooming on the map. We could also add an option where instead of clicking on the map, they can type in their own LCZ, Zoning, and MRT, and have an output that is not dependent on the interactive map. Finally, the user could also benefit from an additional tab describing the definitions of different Local Climate Zones and Zoning Types. This would allow them to better understand the justification behind our infrastructure suggestions.

5.2.2 Decision Algorithm

In the future, we are planning to integrate machine learning into our decision algorithm. We had originally explored sk-learn's decision tree and random forest models, as well as a Llama-3 model, we will continue to explore other methods that could work for our purposes. We would also like to integrate cost as a factor, whether it be sorting the list of solutions by cost or just filtering the most expensive solutions out to begin with. Finally, we would like to see how our model could change if instead of a list of separate infrastructure solutions, we provided a solution that combined different infrastructures for the largest total temperature reduction.

References

1. F. A. Schneider B. Hagen A. Middel, S. AlKhaled and P. Coseo. 50 grades of shade. *International Journal of Biometeorology*, 102(9):E1805, 2021.
2. N. Chhetri A. Middel and R. Quay. Urban forestry and cool roofs: Assessment of heat mitigation strategies in phoenix residential neighborhoods. *Urban Forestry Urban Greening*, 14(1):178, 2015.
3. I. Buo and A. Middel. Hourly mean radiant temperature distribution on a summer day (2012-06-27) in the greater phoenix, arizona (usa) metropolitan area. *Environmental Data Initiative*, 2022.
4. S. W. Myint S. Kaplan A. J. Brazel C. Wang, A. Middel and J. Lukasczyk. Assessing local climate zones in arid cities: The case of phoenix, arizona and las vegas, nevada. *ISPRS Journal of Photogrammetry and Remote Sensing*, 141:59, 2018.
5. A. W. News EI&E. America's hottest city is having a surge of deaths. *Scientific American*, 2024.
6. J. Larson A. Nayak K. Kessler, H. Gross and J. Liu. Portland urban development quantifying and visualizing urban heat with compounding vulnerabilities to support community depaving initiatives. *https://ntrs.nasa.gov/api/citations/20230013297/downloads/2023Sum*$_{V}EJ_{P}ortlandUrban_{T}echPaper_{V}2.pdf$, 2023.
7. J. Feddema K. W. Oleson, G. B. Bonan. Effects of white roofs on urban temperature in a global climate model. *https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2009GL042194*, 2010.
8. Middel Ariane Vanos Jennifer K. Hernandez-Cortes Danae Buo Isaac Hondula David M. Li Rui, Chester Mikhail V. Effectiveness of travel behavior and infrastructure change to mitigate heat exposure. *Frontiers in Sustainable Cities*, 5, 2023.
9. Travis Anderson Margaret C. Morrissey-Basler, Gabrielle J. Brewer. The effect of heat mitigation strategies on thermoregulation and productivity during simulated occupational work in the heat in physically active young men. *https://www.frontiersin.org/journals/sports-and-active-living/articles/10.3389/fspor.2023.1274141/full*, 2023.
10. E. Scott Krayenhoff Kirsten Zickfeld Ariane Middel Mehdi Aminipouri, Anders Jensen Knudby. Modelling the impact of increased street tree cover on mean radiant temperature across vancouver's local climate zones. *Urban Forestry Urban Greening*, 39:9, 2019.
11. McPhearson T. Peng X. E. Wu Y. Fan P. Bowe E. Outwater J. Kennedy C. Sauter, D. Ocellus xr. *https://ntrs.nasa.gov/api/citations/20230013297/downloads/2023Sum*$_{V}EJ_{P}ortlandUrban_{T}echPaper_{V}2.pdf$, 2022.

RunTempo: Dynamic music system based on cadence for enhanced running experience Title

Srijan Kunta and Jason Zhou

Abstract The RunTempo project aims to enhance the running experience by matching the song tempo to the runner's step cadence. Many studies show that the effects of music on running performance positively impact exercise by elevating heart rate, improving endurance, and enhancing how the runner feels about the overall workout. In addition, it's shown that the tempo of the music is able to affect a runner's step cadence since runners frequently adjust their pace to match the music they listen to. The problem with existing programs is that the tempo range is too large, and the recommendation system is not dynamic. We sought to build a solution that dynamically gave users songs in real-time based on their running cadence by utilizing the Spotify API and data from the phone's accelerometer. RunTempo dynamically adjusts the music to synchronize with the runner's pace, thereby improving motivation and performance. This paper describes the development process, methodologies, and the impact RunTempo will have on the running

1 Introduction

Running is one of the most popular and fundamental forms of exercise worldwide. According to a 2021 study by World Athletics, 40% of surveyed people consider themselves runners.[1] Music is often used by runners to improve their performance and motivation, and studies have shown that music increases endurance, elevates heart rate, and overall improves the running experience.[2]

However, the effectiveness of music is dependent on the synchronization between

Srijan Kunta e-mail: srijankunta@gmail.com · Jason Zhou e-mail: independence106@gmail.com

[1] *Recreational Running Consumer Research Study - World ...*, Nielsen Sports & World Athletics, Apr. 2021, assets.aws.worldathletics.org/document/60b741d388549ceda6759894.pdf

[2] Thakare, Avinash E et al. "Effect of music tempo on exercise performance and heart rate among young adults." International journal of physiology, pathophysiology and pharmacology vol. 9,2 35-39. 15 Apr. 2017

the music's tempo and the runner's cadence[Fig 1]. When researchers examined how runners responded to small changes in the tempo of the music they listened to, they found that "Imperceptible shifts in musical tempo in proportion to the runner's self-paced running tempo significantly influenced running cadence".[3]

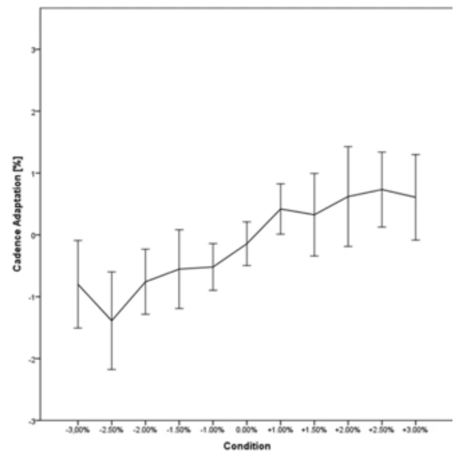


Fig. 1 Condition and Cadence adaptation: when researchers changed the tempo by a certain percentage, the runners would also adjust their running cadence by a proportional amount.

Since it was shown how music tempo is so important towards running, it is no brainer that many solutions already exist. However, the large tempo range and static recommendation system in these solutions cause negative changes in running pace resulting in a lower quality run. The RunTempo project addresses this by creating a dynamic music system that plays songs with a tempo that matches the runner's step cadence in real time. By leveraging the Spotify API(Application Programming Interface) and data from the phone's sensors such as an accelerometer, RunTempo ensures that the music played is in sync with the runner's pace, leading to an optimized and enjoyable running session.

The Spotify API is a powerful tool that allows developers to access and control Spotify's vast music catalog and playback features programmatically. With the API, and the specific tools built for iOS and Android, our app can fetch detailed information about tracks, artists, and albums, as well as create and manage playlists, control playback, and retrieve real-time user data such as currently playing tracks and recommendations.

[3] Van Dyck, E., Moens, B., Buhmann, J. et al. Spontaneous Entrainment of Running Cadence to Music Tempo. Sports Med - Open 1, 15 (2015). https://doi.org/10.1186/s40798-015-0025-9https://doi.org/10.1186/s40798-015-0025-9

2 Related Work and Background

There has been extensive research on using various devices to determine a user's running cadence, such as a trunk-fixed accelerometer[4], magnetometers in phones[5], and wireless earbuds[6].

We used the included pedometer libraries in our phones(apple core motion, and android health) to build reliable algorithms to calculate the running cadence for our app.

By integrating these components, RunTempo can provide a personalized and engaging running experience that changes with the user's cadence.

3 Methodologies

Data collection:

Data collection primarily consists of using the iphone's built in step tracker through core motion and the android's step tracker through google fit. These return steps and steps per minute which match up against bpm. These are found using the phones' built in accelerometers and timers. Using the accelerometers with a tuned pedometer model, the algorithm gets the steps and step counts with their respective time intervals. Using this, the algorithm divides the number of steps by the time interval to calculate the steps per second. We then simply multiplied this value to get a number that would match with songs' BPM, or Beats per Minute.

Music selection using tempo:

Music selection was done through Spotify's SDKs for Ios and Android, as well as the combined use of Spotify's web API for more tailored recommendations. SDKs (Software Development Kits) are collections of tools, libraries, and documentation that help developers build applications for specific platforms. Spotify's SDKs for iOS and Android provide easy-to-use libraries that allow developers to integrate Spotify's music streaming capabilities directly into their apps.

The Spotify App Remote is a key component of Spotify's mobile SDK that allows developers to control Spotify playback directly from their app. It provides a high-level interface for sending commands to the Spotify app running on the user's device, such as playing, pausing, skipping tracks, and adjusting the volume. The

[4] G. Prigent, E. Barthelet, Aminian, K., & A. Paraschiv-Ionescu. (2022). Walking and running cadence estimation using a single trunk-fixed accelerometer for daily physical activities assessment. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. https://doi.org/10.1109/embc48229.2022.9871713

[5] Umran Azziz Abdulla, Taylor, K., Barlow, M., & Naqshbandi, K. Z. (2013). *Measuring Walking and Running Cadence Using Magnetometers*. https://doi.org/10.1109/trustcom.2013.176

[6] Nijs, A., Beek, P. J., & Roerdink, M. (2021). Reliability and Validity of Running Cadence and Stance Time Derived from Instrumented Wireless Earbuds. *Sensors*, *21*(23), 7995. https://doi.org/10.3390/s21237995

App Remote enables apps to interact with Spotify without needing to handle the streaming of music themselves, making it easier to manage playback and maintain a smooth user experience[Fig 2].

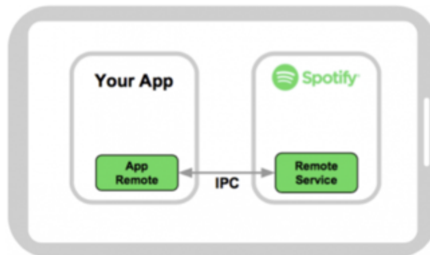


Fig. 2 Spotify App Remote library

Spotify uses OAuth 2.0, which is an industry-standard protocol for authorization. It allows apps like RunTempo to securely access Spotify resources on behalf of users without needing to share or store their login credentials. OAuth 2.0 works by issuing access tokens after the user grants permission, enabling the app to perform authorized actions like controlling playback or retrieving music data. This method ensures that user data and Spotify's systems remain protected while enabling seamless interaction between the app and the Spotify platform[Fig 3].

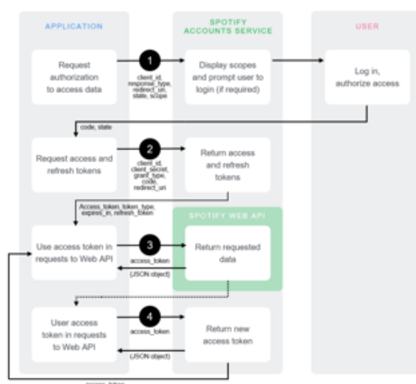


Fig. 3 Spotify's authorization code flow

Using the Spotify API's more advanced recommendation system, we were able to get songs that best matched the user's running cadence. This was achieved through API calls, which are requests sent from the app to Spotify's servers. These requests ask for specific information or trigger certain actions, such as retrieving song recommendations based on the runner's tempo. The Web API processes the request searches its database for the best matches, and then returns a response containing the data the app needs, like a list of songs. Our app then takes the responses and chooses the best songs to play for the user. By leveraging these API calls, the app can dynamically update the music to ensure that it aligns with the user's running cadence in real time.

Testing, Tuning, Additional Features:

Once we developed the backbone of our app, we tested it by going on multiple short runs and observing how our model recommended songs. We noticed that our running cadence algorithm worked perfectly, displaying our running cadence in real-time and transmitting it to our music algorithm. However, the music algorithm had a few problems. Although the songs that it chose always matched the given cadence, it would sometimes recommend only a few select songs repetitively for each tempo. We realized that this was due to the API likely finding multiple songs and putting the best-fitting songs at the top of the list. Since our algorithm originally only asked for one song, Spotify would only give us the top song which is often the same song. Once we realized this we decided to implement more randomization to our algorithm by requesting multiple songs at a time from Spotify and then choosing a random song from that list. We then also decided to randomize multiple other audio features for every API call, such as our preferred danceability, energy, and loudness. By tweaking these variables and introducing randomization we were able to get more diverse recommendations as we ran. Additionally, by using a different seed for each request, we were also able to achieve a more diverse recommendation for the next song.

Extra features were also added by implementing a settings page including an option where users could manually set the tempo of songs they wanted while they ran as well as allowing the user to choose up to 5 preferred genres for more customized song recommendations[Fig 4].

4 Results

After developing the algorithms to find and queue the songs, we built a user interface for our app where users could view currently playing songs, their current running cadence, and additional functionality to pause, play, and skip songs[Fig 5].

We conducted a run with our final iteration and during the run, the app successfully tracked our step cadence and continuously adjusted the tempo of the music to match our pace. The integration of the Spotify API and the app's cadence-tracking algorithm worked seamlessly, providing song recommendations that not only fit our cadence but also adhered to our genre preferences.

Overall, the final test demonstrated that RunTempo is capable of delivering on its goal of optimizing the running experience through tempo-matched music. The combination of accurate cadence detection, responsive song recommendations, and

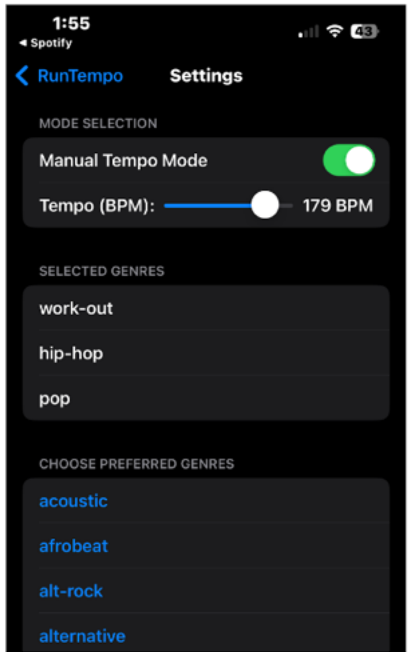


Fig. 4 Settings page with options to choose the preferred music genre

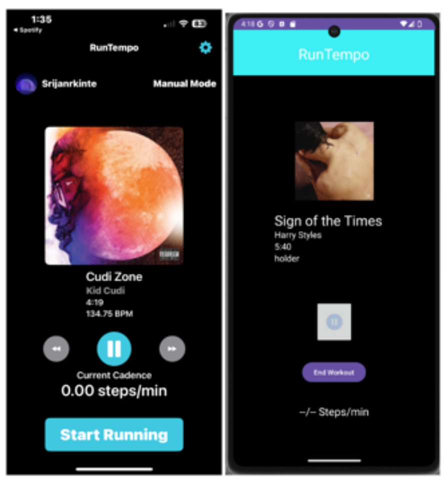


Fig. 5 UI with functionality

real-time adjustments showed the app's potential to enhance the runner's performance and enjoyment.

5 Conclusion

The RunTempo project successfully achieved its goal of enhancing the running experience through the dynamic coupling of song tempo and step cadence. By using data from the phone's accelerometer and the Spotify API toolkit, RunTempo can deliver a personalized and engaging running experience. The app's ability to adjust music tempo in real-time based on the user's step cadence not only improves runner motivation and enjoyment but also has the potential to positively impact running performance.

Our testing showed that the app is capable to delivering diverse and appropriate song recommendations during a run, and the dynamic nature is something not seen in existing solutions offering a more tailored workout experience.

While RunTempo performs very well, we acknowledge that there are further enhancements able to be made such as incorporating heart rate and improving the cadence algorithm. Additionally, expanding the app towards other forms of exercise is something that can be achieved as well. Overall, RunTempo demonstrates the potential to improve exercise experiences, opening new possibilities for music and fitness integration as well as encouraging people to workout more, raising overall health.

Acknowledgements: This work was supported by the altREU program at Portland State University. Special thanks go to Dr. Christof Teuscher for his support during this project, in addition to the guidance from Dr. Bart Massey, Dr. John Lipor, and Yogeshewar Landge from Portland State University.

References

1. *RECREATIONAL RUNNING CONSUMER RESEARCH STUDY NIELSEN SPORTS*. (2021). https://assets.aws.worldathletics.org/document/60b741d388549ceda6759894.pdf

2. Thakare, A. E., Mehrotra, R., & Singh, A. (2017). Effect of music tempo on exercise performance and heart rate among young adults. *International Journal of Physiology, Pathophysiology and Pharmacology*, *9*(2), 35–39. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5435671/

3. Van Dyck, E., Moens, B., Buhmann, J., Demey, M., Coorevits, E., Dalla Bella, S., & Leman, M. (2015). Spontaneous Entrainment of Running Cadence to Music Tempo. *Sports Medicine - Open*, *1*(1). https://doi.org/10.1186/s40798-015-0025-9

4. G. Prigent, E. Barthelet, Aminian, K., & A. Paraschiv-Ionescu. (2022). Walking and running cadence estimation using a single trunk-fixed accelerometer for daily physical activities assessment. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*.
https://doi.org/10.1109/embc48229.2022.9871713

5. Umran Azziz Abdulla, Taylor, K., Barlow, M., & Naqshbandi, K. Z. (2013). *Measuring Walking and Running Cadence Using Magnetometers*.
https://doi.org/10.1109/trustcom.2013.176

6. Nijs, A., Beek, P. J., & Roerdink, M. (2021). Reliability and Validity of Running Cadence and Stance Time Derived from Instrumented Wireless Earbuds. *Sensors*, *21*(23), 7995. https://doi.org/10.3390/s21237995

Linear Forecasting: Predictive Stock Market Trends

Rohan Shah and Joshua Kannagala

Abstract This research explores the application of machine learning techniques to predict stock market prices, integrating historical stock data with sentiment analysis. We examine various machine learning models, including Long Short-Term Memory (LSTM) networks and Principal Component Analysis (PCA), to assess their effectiveness in predicting stock trends. After careful evaluation, sentiment analysis using social media and news data was incorporated to capture market sentiment, providing a more comprehensive approach to stock price prediction. The model developed in this study demonstrates a 3.2% improvement in accuracy when sentiment analysis is included, as measured by the reduction in Mean Absolute Error (MAE). Despite the model's enhanced accuracy, challenges such as the limited availability of historical sentiment data and the potential for overfitting are acknowledged. The findings suggest that combining quantitative financial data with qualitative sentiment insights can lead to more accurate stock price predictions. This research contributes to the growing body of work focused on improving predictive models in finance by incorporating both traditional and sentiment-based analysis techniques.

Rohan Shah
University of Illinois, Champaign, IL, e-mail: rohan.m.s1211@gmail.com

Joshua Kannagala
University of Illinois, Champaign, IL e-mail: joshuakann23@gmail.com

1 Introduction

Predicting stock prices is a complex and challenging task, yet it remains such a fundamental objective for investors and financial analysts aiming to make informed decisions. The motivation for this project arises from the constantly changing nature of the stock market, influenced by economic indicators, market sentiment, and global events. In a world where financial markets can turn on a dime, harnessing the power of machine learning presents an exciting opportunity to enhance the accuracy of stock price predictions [1].

This research explores the potential of machine learning techniques to predict stock prices, providing a thorough analysis of various models and their performance. By combining historical stock data with sentiment analysis, the study aims to uncover patterns and insights that can improve prediction models.

2 Related Work and Background

Previous research on stock price prediction has implemented many different different machine learning methods, including LSTM models, PCA for reducing data complexity, and sentiment analysis. We combined the best ideas and positive outcomes of these approaches to build a model that uses both past data and sentiment scores to improve prediction accuracy.

2.1 Stock Prediction with High Accuracy using Machine Learning Techniques

In this article, authors Bansal, Goyal, and Choudhary examine and compare the use of five different machine learning algorithms to accurately predict stock market trends in the Indian Stock Market. These five models include Linear Regression, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) networks, K-Nearest Neighbors, and Decision Tree Regression networks. Through a comparative analysis of the performances of each method, the study concludes that deep learning algorithms proved to be the most accurate [2].

2.2 Stock Price Prediction using LSTM and its Implementation

In this article, Siddharth M explores how accurate Long Short-Term Memory (LSTM) networks are in analyzing complex patterns in historical stock price data. The article provides a detailed tutorial on using LSTM networks for stock price prediction, covering the theoretical background of LSTM, the challenges of traditional methods, the architecture of LSTM, and a practical implementation using Python. By leveraging LSTM's ability to capture long-term dependencies, the study demonstrates how LSTM networks can effectively analyze time-series data and predict stock prices [3].

2.3 Stock Price Prediction using Principal Components (PCA)

In this article, authors Ghorbani and Chong investigate the application of Principal Component Analysis (PCA) in predicting stock market prices. The study uses PCA to reduce dataset dimensionality, focusing on principal components that capture the most variance. The authors apply PCA to historical stock data and use these components in machine learning models to predict future prices. Their findings demonstrated that PCA enhances prediction accuracy by isolating key patterns and trends, making the models more robust compared to those without PCA integration [4].

2.4 Stock trend prediction using sentiment analysis

In this article, Xiao and Ihnaini introduce a novel weighted sentiment index for predicting stock trends using Twitter and news data, analyzed with VADER and FinBERT, respectively. They compare the predictive performance of tweets and news across different time divisions, employing algorithms like K-Nearest Neighbors, Decision Trees, SVM, Random Forests, Naïve Bayes, and Logistic Regression. The study finds that Naïve Bayes consistently delivers the best accuracy, particularly with skewed datasets, demonstrating that combining news and tweet sentiment scores improves prediction accuracy [5].

3 Data and Methodologies

To build a comprehensive stock price prediction model, various tools and methods were utilized to ensure accuracy and relevance. Google Cloud APIs played an integral role in efficiently retrieving data using Python. This allowed large volumes of information from historical stock data sources to be automated onto spreadsheets. Yahoo Finance served as the primary source for historical and real-time stock data, providing extensive and reliable market information. Additionally, sentiment scores and data were gathered using VADER and BeautifulSoup to incorporate market sentiment into the model, enhancing its predictive power by combining both quantitative financial data and qualitative insights [6].

3.1 Data Collection

After gathering historical stock data from many companies over the past ten years, the data was organized in Google Sheets. The spreadsheets focused on technology companies, including major companies like Apple, Microsoft, and Google. Following the collection of basic information from Yahoo Finance, additional metrics were calculated using Python and subsequently stored in the spreadsheets for further analysis with machine learning.

3.1.1 Company Historical Stock Price Data

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	Symbol	SMA_20	EMA_20	Volatility_20	BB_upper	BB_lower
2024-08-13 00:0	219.00999450	221.88999938	219.00999450	221.27000427	44155300	0.0	0.0	AAPL	nan	221.27000427	nan	nan	nan
2024-08-12 00:0	216.07000732	219.50999450	215.60000610	217.52999877	38028100	0.25	0.0	AAPL	nan	220.91381327	nan	nan	nan
2024-08-09 00:0	211.85479245	216.52937448	211.72493788	215.99000549	42201600	0.0	0.0	AAPL	nan	220.44487919	nan	nan	nan
2024-08-08 00:0	212.86361928	213.95235545	208.58856870	213.06338500	47161100	0.0	0.0	AAPL	nan	219.74187975	nan	nan	nan
2024-08-07 00:0	206.66078565	213.39299866	206.15138077	209.57742309	63516400	0.0	0.0	AAPL	nan	218.77383626	nan	nan	nan
2024-08-06 00:0	205.06264094	209.74722094	200.83753582	206.99040222	69660500	0.0	0.0	AAPL	nan	217.65160444	nan	nan	nan
2024-08-05 00:0	198.85982239	213.25316621	195.77339849	209.02806091	119548600	0.0	0.0	AAPL	nan	216.83031458	nan	nan	nan
2024-08-02 00:0	218.89661861	225.33917349	217.45829631	219.60580444	105568600	0.0	0.0	AAPL	nan	217.09464695	nan	nan	nan
2024-08-01 00:0	224.11059000	224.22046343	216.76909684	218.10754394	62501000	0.0	0.0	AAPL	nan	217.1911133347	nan	nan	nan
2024-07-31 00:0	221.18398273	223.56123595	220.37492166	221.82324218	50036300	0.0	0.0	AAPL	nan	217.63226846	nan	nan	nan
2024-07-30 00:0	218.93659134	220.07527275	215.87013333	218.54704284	41643800	0.0	0.0	AAPL	nan	217.71938983	nan	nan	nan
2024-07-29 00:0	216.70916725	219.04645819	215.50055949	217.98768615	36311800	0.0	0.0	AAPL	nan	217.74494186	nan	nan	nan
2024-07-26 00:0	218.44714249	219.23623765	215.76025015	217.70800781	41601300	0.0	0.0	AAPL	nan	217.74142433	nan	nan	nan
2024-07-25 00:0	218.67687825	220.59467187	214.37186367	217.23855590	51391200	0.0	0.0	AAPL	nan	217.69353210	nan	nan	nan
2024-07-24 00:0	223.74101699	224.54009510	216.87896478	218.28732299	61777600	0.0	0.0	AAPL	nan	217.75008361	nan	nan	nan
2024-07-23 00:0	224.11058796	226.67762395	222.42253943	224.74984741	39960300	0.0	0.0	AAPL	nan	218.41672778	nan	nan	nan
2024-07-22 00:0	226.74754193	227.51665598	222.83207578	223.70108032	48201800	0.0	0.0	AAPL	nan	218.91999945	nan	nan	nan
2024-07-19 00:0	224.56007929	226.53778582	223.02185124	224.05065917	49151500	0.0	0.0	AAPL	nan	219.40863371	nan	nan	nan
2024-07-18 00:0	230.01376046	230.17357913	222.01302671	223.92080688	66034600	0.0	0.0	AAPL	nan	219.83836449	nan	nan	nan
2024-07-17 00:0	229.18472003	231.19240594	226.37797123	228.61538696	57345900	0.0	0.0	AAPL	218.38961334	220.67427139	5.5706223801	229.53085810	207.24836858
2024-07-16 00:0	234.72830752	235.99684350	232.06139624	234.54852294	43234300	0.0	0.0	AAPL	219.05353927	221.99562868	6.6237316370	232.30100255	205.80607600
2024-07-15 00:0	236.20659488	236.95572779	232.82051476	234.12899780	62631300	0.0	0.0	AAPL	219.88348922	223.15118765	7.4153983848	234.71428599	205.05269245
2024-07-12 00:0	228.65533061	232.37103092	228.41560260	230.27345275	53046500	0.0	0.0	AAPL	220.59766159	223.82949861	7.7029243053	236.00351020	205.19181297
2024-07-11 00:0	231.12247556	232.12131940	225.50897806	227.30690002	64710600	0.0	0.0	AAPL	221.30983734	224.16067970	7.6277557056	236.56534875	206.05432593

Fig. 1 AAPL Stock Data Metrics

3.1.2 Sentimental Scores

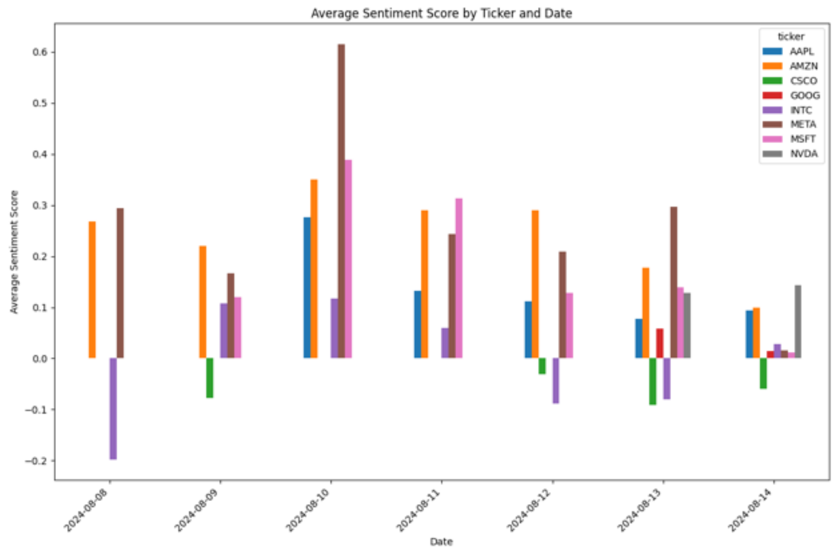


Fig. 2 Average Sentiment Scores of Several Technology Companies in a week

3.2 Methods

3.2.1 Visual Studio Code

Visual Studio Code served as the main coding platform due to its flexibility and extensive support for Python libraries. As a result, Python was also chosen as the main programming language, as many of these libraries were needed for machine learning. Key libraries included:

- **Pandas**: Used for data manipulation and analysis, particularly for organizing stock price data into data frames for easy computation.
- **Yfinance**: A Python library providing easy access to historical market data from Yahoo Finance, allowing the retrieval of stock prices, volume, and other important financial metrics.
- **Scikit-learn (Sklearn)**: Employed for implementing machine learning models, particularly for preprocessing data, training models, and evaluating their performance.

3.2.2 Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable (in this case, stock prices) and one or more independent variables (such as historical prices and sentiment scores). This approach was employed to predict future stock prices by identifying the linear relationship between these variables. The model calculates a best-fit line that minimizes the difference between predicted and actual stock prices [7].

3.2.3 Mean Absolute Error

Mean Absolute Error (MAE) is a metric used to evaluate the accuracy of regression models. It measures the average magnitude of errors between predicted and actual values, providing an indication of how close the predictions are to the real stock prices. In this study, MAE was utilized to assess the performance of the linear regression model, serving as a key metric to determine the model's accuracy and effectiveness [8].

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} | x_i - x |$$

Fig. 3 Mean Absolute Error Formula

3.2.4 Sentiment Methods

To incorporate sentiment analysis into the stock price predictions, several tools and methodologies were utilized:

- **VADER Sentiment Analysis**: Used to analyze the sentiment of text data from social media posts and news articles. VADER is particularly well-suited for analyzing short, informal text from sources like Twitter, making it ideal for this project.
- **BeautifulSoup**: A Python library used for web scraping, which allows the collection of news articles from various financial news websites. This helped create a comprehensive dataset of news sentiments to be analyzed alongside stock prices.
- **Finviz**: An online stock screener that was used as the main source of additional sentiment data, news articles, and financial statements.

4 Results

To evaluate whether sentiment analysis had an affect on the model's accuracy, the mean absolute error was calculated on both the training and testing data sets for the eight technology stocks. Through the tables and column charts below, a comparison can be drawn between the MAE of the training and testing sets with and without sentiment analysis.

4.1 Model Evaluation

Stock	Training Dataset (MAE)	Testing Dataset (MAE)
APPL	1.72	1.64
MSFT	2.6	2.4
GOOG	1.15	1.15
AMZN	1.88	1.65
META	4.15	3.9
NVDA	0.66	0.64
INTC	0.63	0.63
CSCO	0.46	0.45

Fig. 4 MAE without Sentiment Analysis

Stock	Training Dataset (MAE)	Testing Dataset (MAE)
APPL	1.65	1.63
MSFT	2.2	2.5
GOOG	0.95	1.12
AMZN	1.73	1.54
META	3.78	3.71
NVDA	0.59	0.62
INTC	0.6	0.62
CSCO	0.41	0.43

Fig. 5 MAE with Sentiment Analysis

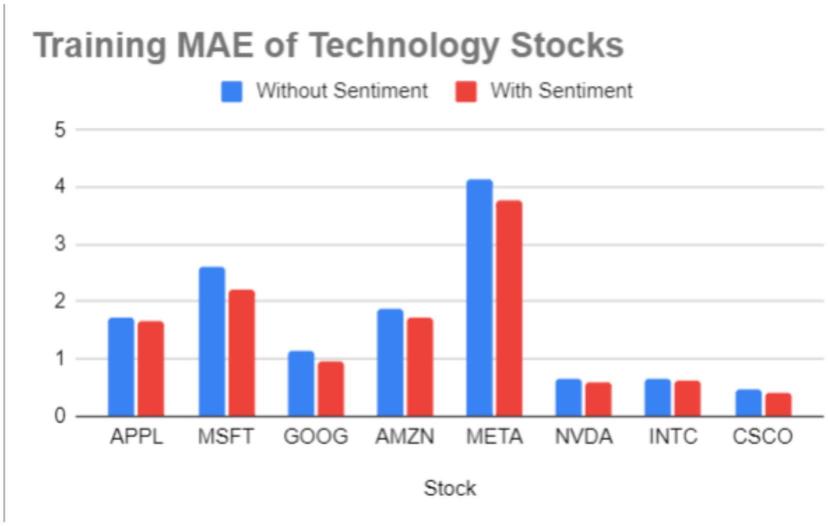


Fig. 6 MAE in Training Set

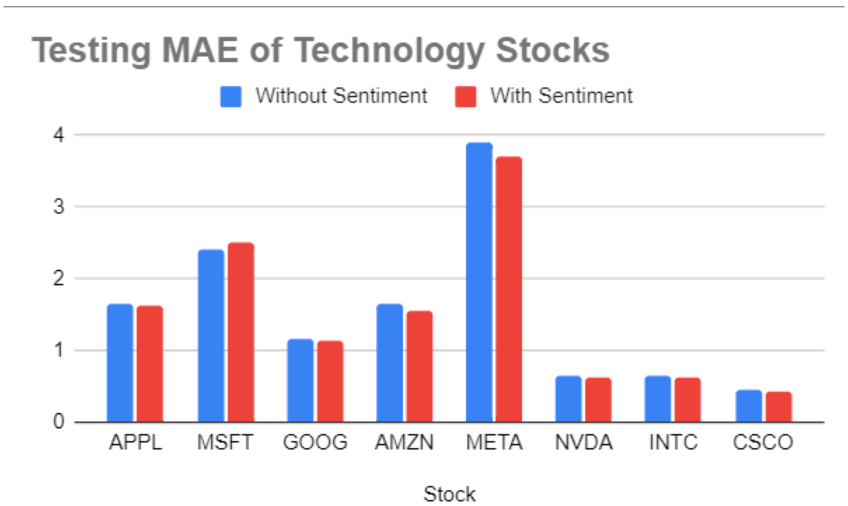


Fig. 7 MAE in Testing Set

Through observing the results, the integration of sentiment analysis led to a 3.2% reduction in Mean Absolute Error, enhancing the model's accuracy. Despite the challenges posed by the limited historical sentiment data, the inclusion of sentiment metrics provided valuable insights, particularly for stocks with high public engagement.

5 Conclusion

The research presented demonstrates that the integration of machine learning techniques with sentiment analysis can enhance the predictive accuracy of stock prices. The model developed showed a Mean Absolute Error reduction of 3.2% when sentiment analysis was utilized, indicating a concrete improvement in prediction accuracy. However, it was observed that the model performed better on the training set than on the testing set, suggesting potential overfitting. This discrepancy points to the need for further refinement in the model to ensure that it generalizes well to unseen data.

5.1 Applications

The methodology and findings from this research have several potential applications in finance. For example, investment firms can utilize the enhanced model to make more informed trading decisions by integrating real-time sentiment data with historical financial metrics [9]. Additionally, the approach could be adapted for use in predicting price movements in other financial markets, such as commodities or foreign exchange. The integration of sentiment analysis can also be extended to other domains where understanding the influence of public sentiment is critical, such as in consumer behavior analysis or political forecasting.

5.2 Concerns and Next Steps

Several concerns were identified during the research process. The limited availability of historical sentiment data restricted the analysis primarily to recent trends, which may have affected the model's overall performance. This is due to the fact that the model relied on historical stock price data from the past ten years, and the sentiment data was not available for every single one of those dates. Additionally, the model was only trained on stocks from the technology sector, which are inherently volatile and may not represent broader market dynamics [10]. Future work should focus on expanding the dataset to include a wider range of industries and a more extensive historical sentiment dataset. This would help limit overfitting and improve the model's generalization to unseen data. Additionally, exploring more sophisticated machine learning algorithms and refining sentiment analysis techniques could be a next step in enhancing the model's robustness and accuracy.

Acknowledgements We would like to thank Professor Christof Teuscher and the altREU faculty mentors for their helpful guidance and feedback throughout this research project.

References

[1] Y. Chen, S. H. Sim, S. Yan, and M. Dong, "Stock market prediction using artificial neural networks and sentiment analysis," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1061-1066, 2013, doi: 10.1109/SMC.2013.187.

[2] T. M. Dang et al., "A hybrid deep learning model for stock market prediction," *Procedia Computer Science*, vol. 207, pp. 1230-1242, 2022, doi: 10.1016/j.procs.2022.09.121.

[3] M. S. (2024, July 29). Stock price prediction using LSTM and its implementation. *Analytics Vidhya*.

[4] Ghorbani, M., & Chong, E. K. P. (2020, March 20). Stock price prediction using principal components. *National Institutes of Health*.

[5] Xiao, Q., & Ihnaini, B. (2023, March 20). Stock trend prediction using sentiment analysis. *National Institutes of Health*.

[6] Cristescu, M. P., Nerisanu, R. A., Mara, D. A., & Oprea, S.-V. (2022, November 14). Using market news sentiment analysis for stock market prediction. *MDPI*.

[7] Mo, H. (2023, December). Comparative analysis of linear regression, polynomial regression, and ARIMA model for short-term stock price forecasting. *ResearchGate*.

[8] Hodson, T. O. (2022, July 19). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*.

[9] Chen, Y., Zhao, H., Li, Z., & Lu, J. (2020, December 4). A dynamic analysis of the relationship between investor sentiment and stock market realized volatility: Evidence from China. *National Institutes of Health*.

[10] Rasiova, B., & Arendas, P. (2022, December 1). Copula approach to market volatility and technology stocks dependence. *Finance Research Letters*.

Forecasting Quarterly GDP Using Multivariate LSTM

Khuselbayar Bolor-Erdene

Abstract This paper explores the forecasting of US GDP using a Multivariate Long Short-Term Memory (LSTM) neural network combined with Wavelet Analysis (WA). By leveraging key macroeconomic indicators as inputs, we aim to enhance the predictive accuracy of GDP forecasts. The integration of WA with LSTM enables the model to capture both the temporal dependencies and the non-linear characteristics of the data. Our empirical results demonstrate that the WA-LSTM model significantly outperforms traditional forecasting models, such as simple linear regression and seasonal models like ARIMA, as well as LSTMs without wavelet analysis. This study is built upon the framework presented by Zhang, Yaling, et al. in their research on China's GDP forecasting, extending its application to the US context. The findings highlight the potential of advanced neural network architectures in improving economic forecasting accuracy, though further research is needed to fully explore the capabilities of LSTMs in modeling complex, non-linear economic indicators like GDP.

1 Introduction

The prediction of Gross Domestic Product (GDP) is a long-studied and crucial subject in economic research. Accurate GDP forecasting is essential for policymakers, investors, and businesses as it informs decisions and strategies that impact economic stability and growth. Traditionally, seasonal models like ARIMA (AutoRegressive Integrated Moving Average) and SARIMA (Seasonal AutoRegressive Integrated Moving Average) have been widely used for GDP prediction. However, with the advancement of artificial neural networks, new models such as Long Short-Term Memory (LSTM) networks have emerged, demonstrating superior predictive capabilities.

Khuselbayar Bolor-Erdene
University of Illinois Urbana Champaign, e-mail: kb.bolor@gmail.com

Our project focuses on utilizing a Multivariate LSTM neural network combined with Wavelet Analysis (WA) to forecast US GDP and other key economic indicators. This research was inspired by the need to explore methods of better economic forecasts and utilizing advanced neural network architectures in capturing complex, non-linear patterns in time series data.

LSTMs, a type of RNN, are particularly well-suited for time series forecasting due to their ability to capture long-term dependencies and avoid the vanishing gradient problem that plagues traditional RNNs. Unlike regular RNNs, which struggle to maintain information over long sequences, LSTMs use a gating mechanism to control the flow of information, allowing them to retain relevant information over extended periods. This makes LSTMs more effective in modeling the temporal dynamics of economic indicators, leading to more accurate predictions.

By integrating Wavelet Analysis with LSTM, our model enhances its ability to capture both temporal dependencies and non-linear characteristics of economic data. Wavelet Analysis decomposes the time series data into different frequency components, enabling the model to identify patterns at various scales. This combined approach not only improves the accuracy of GDP forecasts but also offers deeper insights into the underlying economic dynamics.

In this study, we compare the performance of our WA-LSTM model against traditional models like simple linear regression and ARIMA, as well as other neural network models, to demonstrate its superior predictive capabilities. Our findings highlight the potential of advanced neural network architectures in improving economic forecasting accuracy, paving the way for more reliable and informed decision-making in the economic domain.

2 Related Work and Background

The study builds upon the framework presented by Zhang, Yaling, et al. [2] in their work on China's GDP forecasting, extending the application to the US context. This study concluded WA-LSTM models had the best prediction capabilities compared to other models and predicted quarterly GDP with 7.64% accuracy. Also, the study heavily focused on Chinese energy sector data to build their model, while this paper explored different economic indicators, mentioned in the following sections, that fits better in the US context.

3 Data and Methodologies

The data used to build the model were from the Federal Reserve Bank [1]. All indicators were quarterly economic data since 1960 to 2020 containing 240 rows in total.

3.1 Data Collection and Processing

This subsection outlines the data collection and processing methods employed in our study. Our research focuses on forecasting US GDP using a range of key economic indicators, which were selected based on their anticipated impact on GDP changes. The indicators used in this study include:

1. Gross Domestic Product (GDP): The primary variable of interest, representing the total economic output of the country.

2. Consumer Price Index (CPI): An indicator of inflation that measures the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services.

3. Personal Consumption Expenditures (PCE): A measure of the value of goods and services consumed by individuals, reflecting changes in consumer spending patterns.

4. Industrial Production: An index that measures the output of the industrial sector, including manufacturing, mining, and utilities.

5. Exports: The total value of goods and services sold abroad, which affects GDP by influencing net exports.

6. Federal Reserve Rates (Fed rates): The interest rates set by the Federal Reserve, which impact borrowing costs and economic activity.

7. Unemployment Rate: The percentage of the labor force that is unemployed and actively seeking employment, serving as an indicator of labor market conditions.

These indicators were chosen due to their significant influence on GDP fluctuations.

3.1.1 Data Collection

The data processing involved several steps to ensure its quality and suitability for modeling:

The economic indicators were obtained from various sources, including the Federal Reserve Economic Data (FRED) database, the Bureau of Economic Analysis (BEA), in a consistent format to facilitate integration.

The collected data underwent a cleaning process to address missing values, outliers, and inconsistencies. Missing values were imputed using interpolation methods, and outliers were treated to maintain data integrity.

To ensure that the data was on a comparable scale, to speed up model training time and convergence, this paper used MinMax normalization technique.

The processed data was divided into training: 70% of data, validation: 15%, and test sets: remaining 15%. This partitioning ensured that the model could be trained effectively and evaluated on unseen data to assess its generalization performance.

This prepared data was then used to train and evaluate the Multivariate LSTM model combined with Wavelet Analysis, aimed at providing accurate forecasts of US GDP.

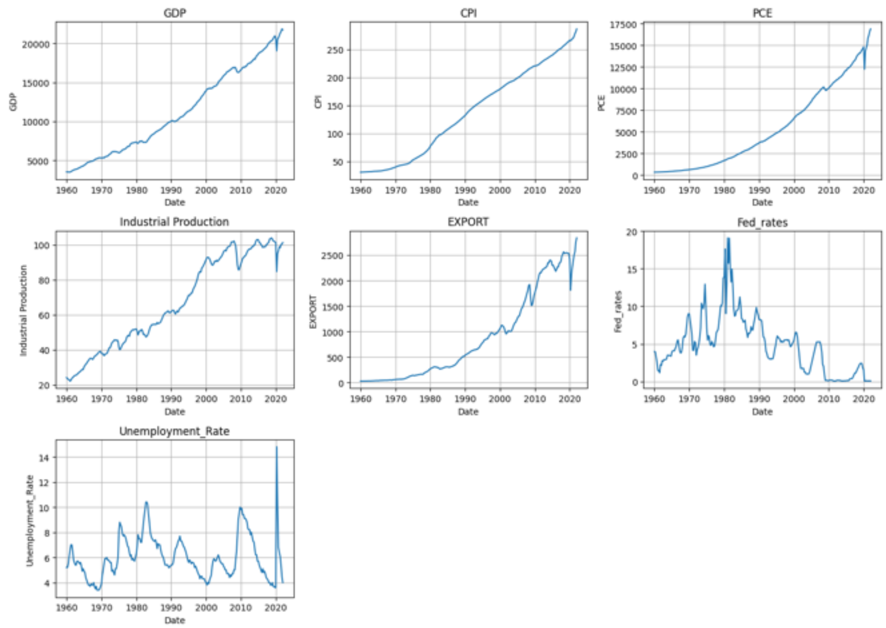


Fig. 1 Indicators Used as Inputs.

3.2 Methods

This paper leverages the strengths of wavelet analysis and LSTM networks to improve the accuracy of GDP forecasts. The inputs were first split into high and low frequency features using wavelet analysis and separate models were build for both and combined optimized through a unified loss function. This demonstrated superior predictive performance compared to traditional forecasting methods.

3.2.1 Wavelet analysis

Time series data, such as economic indicators, often exhibit complex patterns that include both high-frequency noise and low-frequency trends. Traditional methods like Fourier transform can decompose signals into frequency components but fail to provide temporal localization, meaning they cannot effectively capture when certain frequencies occur. Wavelet analysis, however, offers both frequency and time localization, making it an ideal choice for analyzing non-stationary time series data.

By applying wavelet analysis, we can decompose the data into different scales, allowing us to separately analyze and model the high-frequency components (which capture short-term dynamics) and low-frequency components (which capture long-term trends). This decomposition enhances the model's ability to learn and predict complex patterns in the data, leading to more accurate forecasts.

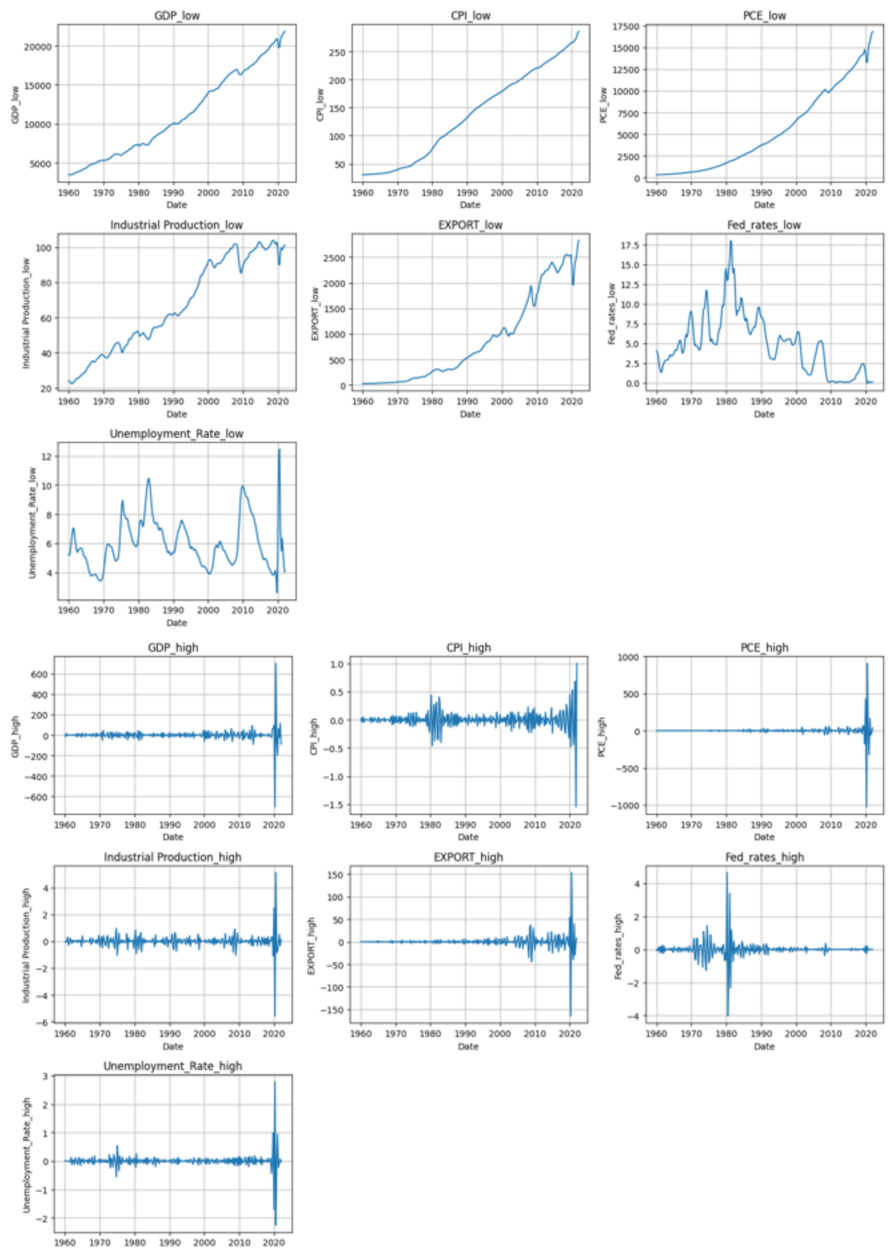


Fig. 2 Low and High Frequency Component Inputs.

3.2.2 Rolling Window Method

To capture the temporal dependencies in the time series data, this paper employed the rolling window method. This approach involved using a fixed-size window of past observations to predict future values. Specifically, we used one year (four quarters) of historical data to predict the next quarter's GDP.

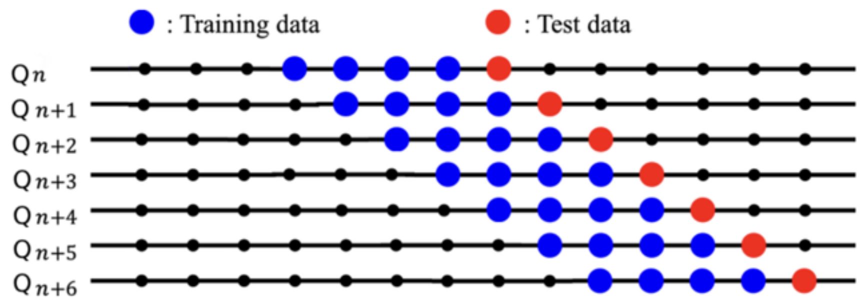


Fig. 3 Low and High Frequency Component Inputs.

3.2.3 Model Architecture

In this subsection, we describe the architecture of the combined low-frequency and high-frequency LSTM model used for forecasting US GDP.

The low-frequency model is designed to capture long-term trends and patterns in the data while the high-frequency model captures short-term fluctuations and noise. The architecture for both models are as follows:

The input shape corresponds to the low-frequency components of the training data, with dimensions representing the number of timesteps and features. Two LSTM layers with 32 and 16 units respectively, both using ReLU activation functions. The first LSTM layer returns sequences to feed into the next LSTM layer. Finally, a fully connected Dense layer with 8 units and ReLU activation to further process the output from the LSTM layers - seen in Fig. 3

To integrate the outputs from both the low-frequency and high-frequency models, the following steps are implemented. The outputs from the dense layers of both models are concatenated into a single tensor. A Dropout layer with a rate of 0.2 is applied to prevent overfitting by randomly setting a fraction of input units to 0 during training. Two additional Dense layers are applied. The first Dense layer with 8 units and ReLU activation to further process the combined input. The final Dense layer with 1 unit and linear activation to produce the single GDP prediction, which is the predicted value of the next quarter's GDP or other economic indicators.

Model Compilation The combined model is compiled using the Adam optimizer with a learning rate of 0.001, epochs of 100 and batch size of 32 when compiling.

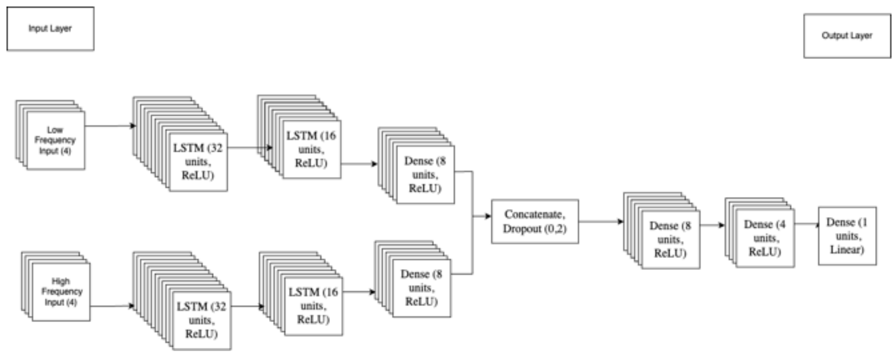


Fig. 4 Model Architecture.

The loss function used is Mean Squared Error (MSE), and the model is evaluated using Mean Absolute Error (MAE) as the metric. Also early stopping and learning rate scheduler were used.

4 Results

4.0.1 Results

The effectiveness of the WA-LSTM model in forecasting US GDP and other key economic indicators was evaluated by comparing its performance with several other models, including linear regression, traditional seasonal ARIMA, and a regular LSTM model. The primary metric used for evaluation was the Mean Absolute Percentage Error (MAPE).

Seen in Table 1, the WA-LSTM model achieved a MAPE of 6.28%. This result demonstrates the superior predictive performance of the WA-LSTM model, highlighting its ability to capture both short-term fluctuations and long-term trends in the economic indicators. The regular LSTM model, without wavelet analysis, achieved a MAPE of 8.44%. While better than linear regression and ARIMA, the regular LSTM still fell short of the WA-LSTM model's performance. This result underscores the added value of integrating wavelet analysis with LSTM networks. The linear regression model had a MAPE of 15.76%. This higher error rate indicates that linear regression, while simple and easy to implement, is less effective in capturing the complex, nonlinear relationships present in the economic data. The traditional seasonal ARIMA model achieved a MAPE of 15.45%. Despite being designed to handle seasonality and trends in time series data, the ARIMA model underperformed compared to the WA-LSTM model, likely due to its limitations in modeling non-linear relationships.

The results clearly indicate that the WA-LSTM model outperforms traditional forecasting models and even a regular LSTM model. The incorporation of wavelet analysis allowed the WA-LSTM model to better capture the intricate patterns in the data, resulting in a significantly lower prediction error.

Model	MAPE (%)
WA-LSTM	6.28
Regular LSTM	8.44
Linear Regression	15.76
Seasonal ARIMA	15.45

Table 1 Comparison of MAPE for different models on testing data.

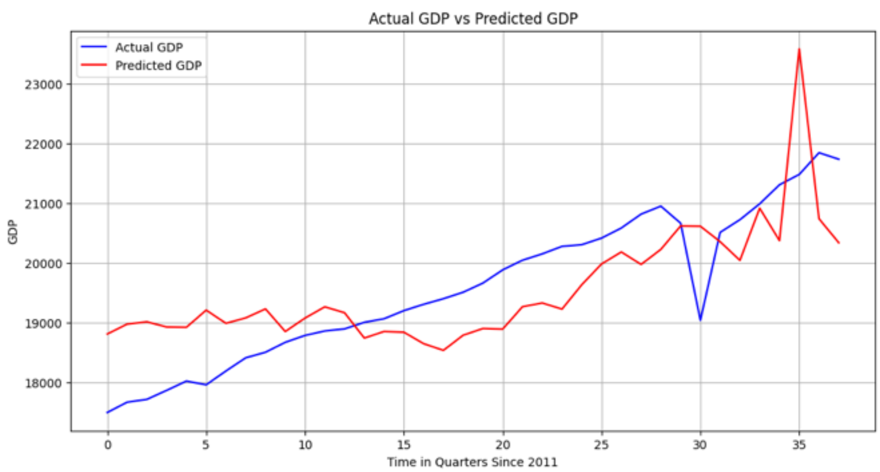


Fig. 5 Predictions of the testing data set.

The significantly lower MAPE of the WA-LSTM model demonstrates its robustness and accuracy in forecasting GDP and other economic indicators. The integration of wavelet analysis with LSTM networks provided a substantial improvement over both traditional and other neural network models, highlighting its potential for future economic forecasting applications.

5 Conclusion

In this study, we explored the efficacy of using a Multivariate LSTM with Wavelet Analysis (WA-LSTM) to forecast US GDP and other key economic indicators. By integrating wavelet analysis with LSTM networks, we significantly enhanced the model's ability to capture both short-term fluctuations and long-term trends in economic data. The WA-LSTM model outperformed traditional forecasting models such as linear regression and seasonal ARIMA, as well as a regular LSTM model, achieving the lowest Mean Absolute Percentage Error (MAPE) on testing data. This demonstrates the potential of advanced neural network architectures in improving the accuracy of economic forecasting.

The findings from this study have several potential applications in economic policy and planning, financial markets, and business strategy. The model's ability to predict economic indicators with higher accuracy can be applied to various other fields that rely on time series forecasting, such as energy demand prediction, weather forecasting, and supply chain management.

5.1 Concerns and Next Steps

Incorporate additional relevant economic indicators and data sources to enhance the robustness of the model. Experiment with different neural network architectures and hyperparameters to optimize the model's performance and reduce computational requirements. Apply the WA-LSTM model to economic data from other countries or regions to generalize to different economic contexts. Develop and implement techniques to enable real-time forecasting, making the model more practical for real-world applications.

In conclusion, the integration of wavelet analysis with LSTM networks presents a promising approach to improving the accuracy of economic forecasts. By pursuing the proposed next steps, we can further enhance the model's performance and broaden its applicability to various fields.

References

1. Federal Reserve Bank of St. Louis. Federal reserve economic data. https://www.stlouisfed.org/. Accessed: 2024-07-29.
2. W. Zhang N. Pan X. Huang B. Zhang, Y. Shang. Quarterly gdp forecast based on coupled economic and energy feature wa-lstm model. *Front. Energy*, 2023.

Develop an Agent-based Model for Assessing the Impact of Autonomous Vehicles on Travel and Land Use in Portland, Oregon

Faye, Linden & Wang, Liming

Abstract The advent of autonomous vehicles (AVs) presents a transformative force poised to reshape urban landscapes, influencing travel patterns, land use, and urban form. However, while the potential impacts of AVs are widely recognized, our understanding of their complex and interrelated effects remains limited. Traditional modeling approaches often struggle to capture the dynamic interactions between AV technology, human behavior, and the environment. In this work, we elaborate on these relations, employing an agent-based approach to explore the complex interactions between AVs and the urban environment. We developed an agent-based model (ABM) to simulate the behaviors of individual agents (households, vehicles) and their decision-making processes regarding travel mode choices and residential location preferences. Our model will eventually allow comparison of different scenarios of AV adoption and policies, and enable the assessment of the potential impacts of AVs on travel patterns, land use, and the effectiveness of various policies. The model will provide valuable insights for planners and policymakers to anticipate and proactively manage the transformative effects of AVs on urban areas like Portland.

1 Introduction

The advent of autonomous vehicles (AVs, or self-driving vehicles) represents a transformative force poised to reshape urban landscapes, with far-reaching implications for transportation systems, land use patterns, and urban form. As we stand on the cusp of this technological revolution, it is crucial to develop a comprehensive understanding of the potential impacts of AVs on our cities and communities. While the significance of AVs is widely acknowledged, our grasp of their complex and interre-

Linden Faye
Hofstra University, Address of Institute, e-mail: lfaye1@pride.hofstra.edu

Liming Wang
Portland State University, 506 SW Mill St, Portland, Oregon e-mail: lmwang@pdx.edu

lated effects remains limited, necessitating innovative approaches to modeling and analysis.

The integration of AVs into urban environments is expected to dramatically alter travel behaviors, mobility patterns, and residential preferences. These changes could lead to shifts in urban density, parking requirements, and the overall structure of cities [6]. However, the precise nature and extent of these impacts are subject to considerable uncertainty, given the complex interactions between technological advancements, human behavior, and environmental factors.

Traditional modeling approaches often struggle to capture the dynamic and multifaceted nature of these interactions. Static models or those based on historical trends may fail to account for the disruptive potential of AVs and the adaptive behaviors of individuals and institutions in response to this new technology [11]. Consequently, there is a pressing need for more sophisticated modeling techniques that can simulate the complex, emergent phenomena associated with AV adoption and its urban impacts.

In response to this challenge, this project develops an agent-based model (ABM) to explore the intricate relationships between AVs and the urban environment. ABM offers a promising solution by simulating individual agents and their adaptive decision-making processes, allowing us to explore the emergent effects of AVs on urban systems as complex adaptive systems [4] and study policies that have the potential to address AV's adverse effects.

Our agent-based model simulates the behaviors of individual worker agents and their decision-making processes regarding commuting mode choices. This granular approach allows us to account for heterogeneity in agent characteristics, preferences, and behaviors, providing a more nuanced understanding of how different segments of the population might respond to the introduction of AVs.

The primary objectives of this project are:

- To develop an agent-based model that enables simulating the adoption of AVs and their impacts on urban systems.
- To explore various scenarios of AV adoption and policy interventions, assessing their potential effects on travel patterns and land use.
- To evaluate the effectiveness of different policy approaches in managing the urban transitions associated with AV technology.

By achieving these objectives, our research aims to provide valuable insights for urban planners, policymakers, and researchers grappling with the challenges and opportunities presented by AVs. Understanding the potential impacts of this technology is crucial for developing proactive strategies to shape urban futures that are sustainable, equitable, and resilient. At this stage, the model is a proof-of-concept for a hypothetical monocentric city. We plan to continue to improve and eventually enable assessment of realistic scenarios for regions like Portland.

In the following sections, we will summarize the related work, detail the methodology underlying our agent-based model, present the results of our simulations, and discuss the implications of our findings for urban planning and policy. Through this work, we hope to contribute to the growing body of knowledge on the impacts of autonomous vehicles and to inform evidence-based decision-making in the face of this transformative technology.

2 Related Work and Background

The advent of autonomous vehicles (AVs) promises to revolutionize transportation systems globally. Understanding AV effects on traffic flow, safety, energy consumption, and urban dynamics is crucial for policy development, infrastructure planning, and societal preparation [5, 11, 7]. This imperative has driven diverse modeling approaches to predict and analyze AV impacts.

Current modeling efforts span a range of methodologies. Microsimulation tools like VISSIM have been adapted for AV behavior [14], while system dynamics approaches explore long-term adoption impacts [12, text]. Discrete choice models predict adoption rates and mode choice behaviors [1], and game theory provides insights into AV-human vehicle interactions [15]. Soteropoulos et al. [13, -] provide a comprehensive review of these modeling studies, highlighting the diverse impacts of AVs on travel behavior and land use across different scenarios.

Agent-Based Modeling (ABM) has emerged as a particularly promising approach due to its ability to capture complex, emergent phenomena in transportation systems. Huang et al. [8] offer an overview of ABM applications in transport simulation, emphasizing its advantages in modeling systems with heterogeneous agents. These characteristics make ABM well-suited for exploring the multifaceted impacts of AVs on transportation and urban environments[4].

The strengths of ABM in AV research are many-folded. By simulating individual agents with unique behaviors, ABM can reveal unexpected system-level outcomes. Fagnant et al. [6] demonstrated this capability in their study of shared AV fleet operations in Austin, Texas, revealing insights into vehicle relocation strategies and service efficiencies. Similarly, Chen et al. [3] used ABM to explore the potential impacts of AVs on urban parking demand, highlighting the method's utility in understanding secondary effects of AV adoption.

ABM's spatial explicitness allows for the incorporation of detailed geographic information, crucial for understanding localized AV impacts [2]. Zhang and Guhathakurta [16] leveraged this feature in their study of residential location choices in the era of shared AVs, illustrating how ABM can integrate transportation effects with broader urban systems.

Despite these advantages, current ABM applications in AV research face several challenges. Jing et al. [9] provide a systematic review of agent-based simulation of AVs, highlighting issues such as scalability, validation, and the need for more sophisticated models of human-AV interactions. Many models struggle to adequately account for uncertainties in AV technology development and adoption rates [11].

Addressing these gaps presents opportunities for advancing the field. Developing more robust, comprehensive agent-based models can provide valuable insights into

AV effects on transportation systems and urban environments. In this chapter, we describe the proof-of-concept agent-based model for assessing the effects of AV.

3 Methodologies

Agent-Based Modeling (ABM) is a computational approach used to simulate the interactions of multiple entities, or ”agents”, within a predefined environment. ABMs are known for their flexibility and heterogeneity, allowing them to model an innumerable range of novel and elaborate complex systems. This is accomplished through the individually-defined behaviors and independent learning and decision-making of each agent down to the macro level. Each agent operates on their own sets of rules and behaviors, and can be adapted to represent individuals, groups, organizations, and much more. This also allows for the revelation of certain emergent patterns and phenomena from the agents' unique interactions with one another, as opposed to from a more explicitly-defined procedure. The versatility of ABMs helps researchers relate the agent's individual actions with their conjugated outcomes, gaining insight into the underlying dynamics of increasingly complex systems.

In this section, we lay out our approach in utilizing ABM to comprise different scenarios of behaviors, policies, area sizes, and distribution of population among various roles of agents. We utilized the open-source python library Mesa to simulate the complex system of AVs and explore potential emergent behaviors of individuals in the model [10]. Mesa was used to define each agent type and have them interact with one another within the model environment.

Currently, there are two types of agents: an employer agent or a worker agent. They are situated in a grid-like environment, with each employer agent possessing its own unshared space and each worker agent occupying spaces by means of a monocentric density gradient given by a capacity formula, resembling the density gradient observed in most cities.

Let (x, y) represent coordinates in the grid and g represent the size (length of one side) of the grid. The formula for determining the capacity of a single space is as follows:

$$capacity(x, y) = \max\left(10 - \left|x - \frac{g-1}{2}\right| - \left|y - \frac{g-1}{2}\right|, 1\right)$$

The closer the space is to the center of the grid, the higher its capacity, allowing an increased number of worker agents to fill that same space. Additionally, the maximum function ensures that each space may contain at least one agent of any type.

The user can input the values for the model's grid size, amount of employers, and total density of the amount of initial agents, including both employers and workers, relative to the grid size. For convenience's sake, this allows for quick, consecutive generations of the model without resorting to excessive manual scaling for increasingly numerous amounts of agents in a larger environment. Upon receiving these variables, the model then creates and places the agents in accordance with the monocentric density gradient.

Once every agent has been fully generated and placed successfully in their model environment, the model prepares to initiate certain policies, simulating the advent of AVs and their potential impact on the behaviors of various households who plan to ride an AV to work.

Firstly, each worker agent is assigned to one employer agent, with each employer expected to have many different workers at a time. For every step in the model, the workers start by evaluating each of their respective distances to their employers, which will be a primary factor contributing to their decision of riding an AV to work. However, upon the advent of AVs, there will also be many new impediments that will also change their behaviors. For instance, AVs free our attention from driving, allowing us to have an added convenience of long distance travel without any effort and multitask on other important affairs. Therefore, in order to limit unprecedented amounts of road congestion and greenhouse gas emissions, it is likely that a toll will be put into effect, penalizing individuals who spend prolonged periods of time traveling in AVs. This scenario is simulated in the model as a cost-per-distance value inputted by the user, which is then utilized in a probability formula to determine the behavior of whether the worker agent decides to ride the AV to work.

Let c represent the cost-per-distance value in cents and d represent the added horizontal and vertical round trip distance, from the worker's home to their employer and back, measured in spaces. The formula determining the probability of the worker riding the AV to work is as follows:

$$Pr(driving) = \frac{1}{1 + e^{6cd-4}}$$

Lastly, a second probability formula is executed to determine whether the worker will relocate to a new home, situating it in a new starting position for the next step (or iteration) of the simulation:

$$Pr(relocate) = \frac{1}{1 + e^{5cd-5}}$$

Ultimately, upon realizing these steps, two variables will be produced for every worker agent: $distance$, representing their round trip travel distance in an optimized horizontal and vertical path between each worker and their employer, and vehicle distance traveled (VDT), which will either be the same as $distance$ if the worker rides the AV to work, or 0 if they choose otherwise.

In the near future, we plan to continue expanding upon this model. One main feature that we plan to implement is an adaptation for our model's compatibility with GIS, or more specifically, real data on the city of Portland. This will allow us to observe how the agents interact in a non-simulated, real environment, as well as distribute additional roles and behaviors in accordance with the new environment's geography.

4 Results

From our testing, certain user parameter adjustments did not yield a significant pattern in relation to other variables. For the sake of brevity, we therefore report our findings in a series of graphs and maps with fixed user input variables unless otherwise specified. Each simulation iteration will be conducted with an agent density of 2.5 (i.e., the total number of agents, including both employers and workers, will be 2.5 $\times$ # of grid spaces) and cost-per-distance of 0.01 cents, along with a fixed number of 5 employers in a 15×15 grid environment.

4.1 Round Trip Travel Distance

Round trip travel distance was determined by doubling the sum of the horizontal and vertical distances between the worker and the employer. This was achieved with the intention of reproducing the standard cross-sectional alignment of real roads.

Since each distance is represented with an even integer value (as per our horizontal and vertical summation for each round trip), this allows us to delineate these fixed values into a bar graph, showing the frequency of workers with a certain distance to be traveled to their employers and back. Figure 1 displays this relationship of round trip travel distance and frequency of workers.

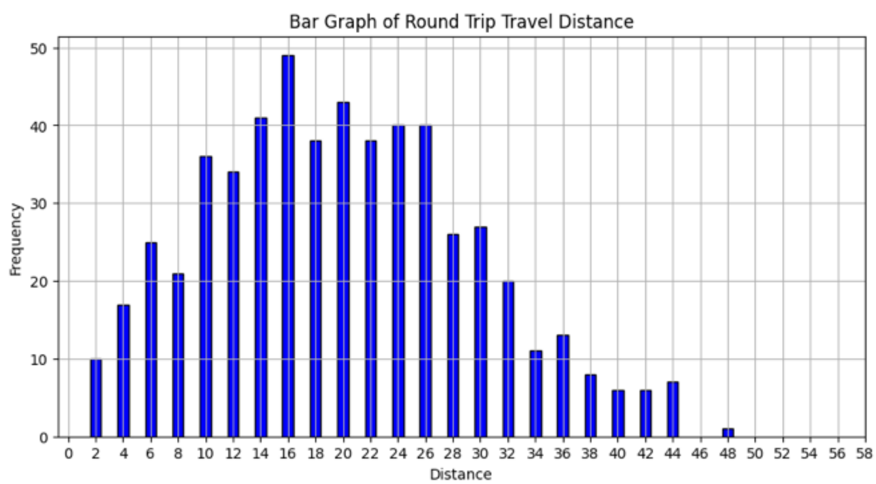


Fig. 1 Bar Graph of Round Trip Travel Distance

In addition to a bar graph representation, it would also be insightful to portray the data in a more generalized form, including a normal distribution of all round trip distance values. Figure 2 displays the worker's travel distance frequencies in a histogram, demonstrating the probability density of each cluster of values with a normal distribution line showing the highest ranges of occurrence.

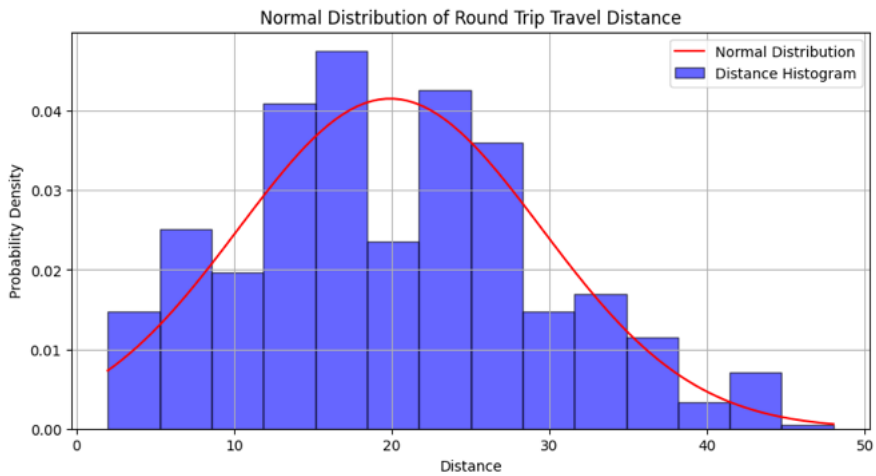


Fig. 2 Distribution of Round Trip Travel Distance

Lastly, it is also crucial to observe the agents and their interactions directly in their environment. Figures 3 and 4 display each grid space's summation of all distances of worker agents occupying their respective squares in a heat map. Employer spaces are colored green, empty spaces are white, and all other spaces follow the heat map color legend depending on the distances of the agents on each space. Figure 4 also shows a scenario where some worker agents in the previous step, Figure 3, have decided to move to a new space upon evaluating the current cost-per-distance, and these spaces have been updated accordingly.

4.2 Vehicle Distance Traveled (VDT)

VDT was used to represent each worker's decision of whether they choose to ride an AV to work or not, decided by the previously cited probability formula. Depending on each worker's distance to their employer and the current cost-per-distance fee, they can either choose to ride the AV for the full round trip distance, giving the agent a VDT equal to their distance, or choose to be absent or go to work by other means, giving the agent a VDT of zero.

Since the probability formula renders the majority of VDT values as non-integers, we move straight to the normal distribution representation of our data. Figure 5 displays each worker's VDT in a histogram, demonstrating the probability density of each cluster of values with a normal distribution line showing the highest ranges of

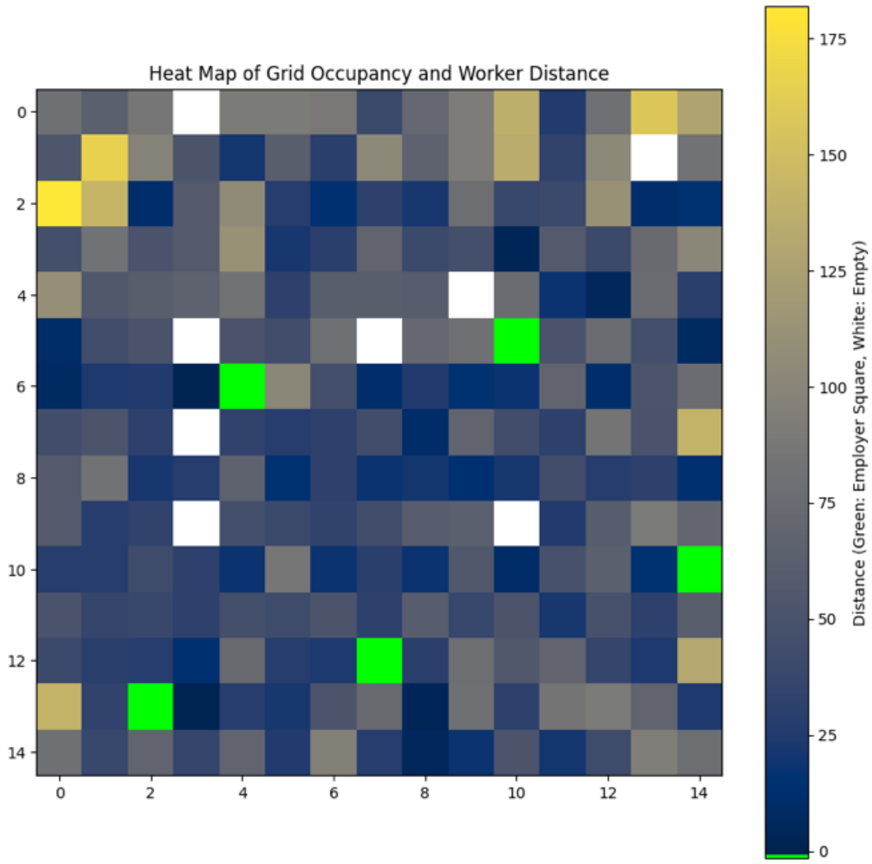


Fig. 3 Heat Map of Grid Occupancy and Worker Distance (Step 1)

occurrence. Note how in comparison to figure 2, the normal distribution histogram of round trip distance, Figure 5 has a portion of values allocated to zero VDT. This is due to how certain worker agents deciding not to ride the AV to work have not travelled their original distance in an AV, moving their distance to 0 for VDT.

This relationship is more clear in Figure 6, where each worker's distance and VDT are plotted against one another in a scatter plot. This graph shows how certain values of round trip distance have some amount of workers choosing not to drive to work, resulting in their VDT to be zero.

In Figures 7 and 8, we observe how a handful of spaces in the heat map have grown darker, falling down to the lower end of the color spectrum, in comparison to Figures 3 and 4. This is due to many worker agents adopting a VDT of zero due to expensive cost-per-distance pricing. Figure 8 also shows the worker agent relocation process following the events of Figure 7.

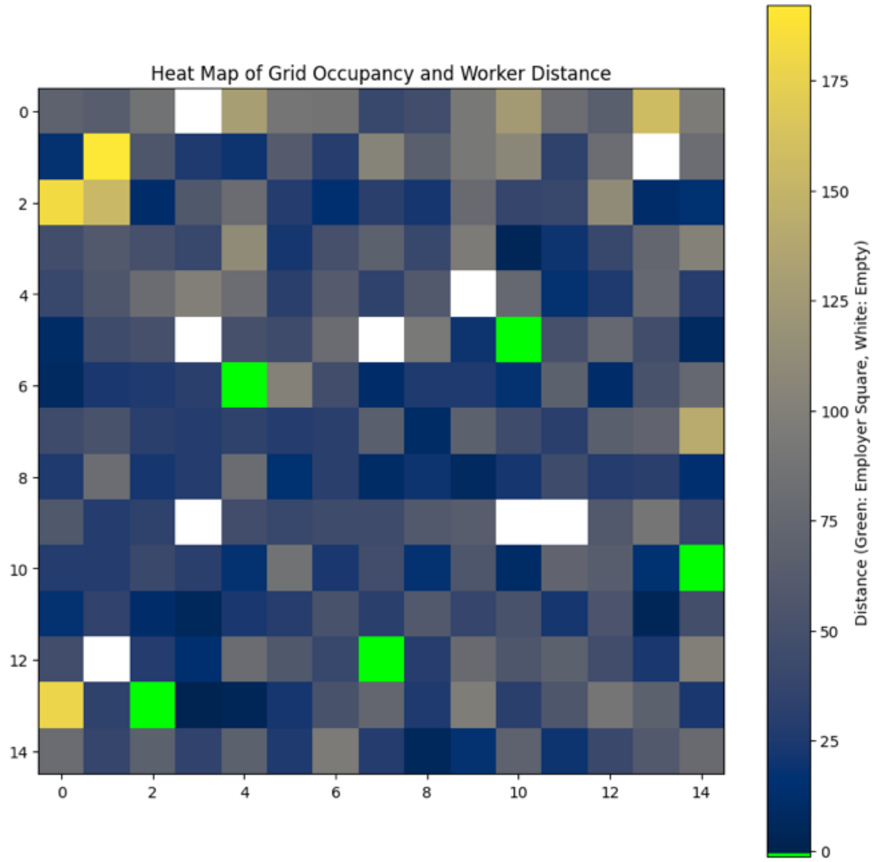


Fig. 4 Heat Map of Grid Occupancy and Worker Distance (Step 2)

4.3 Cost-Per-Distance vs VDT

In the process of testing different values and conditions for our initial variables, we have found an interesting trend. Figure 9 displays the results of a batch run simulation accumulating the data from multiple iterations of the model, showing the relationship between initial cost-per-distance prices and average VDT of all workers. It shows that as the cost-per-distance increases, VDT tends to decrease, as more workers are unenthusiastic about riding the AV to work.

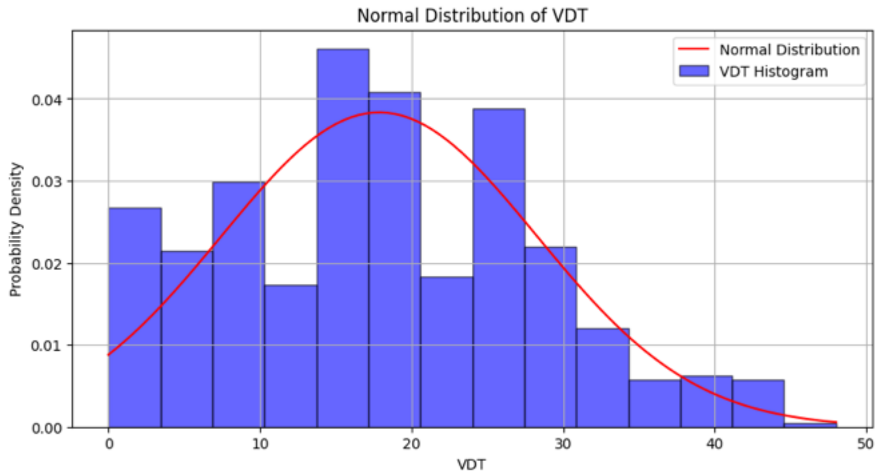


Fig. 5 Normal Distribution of VDT

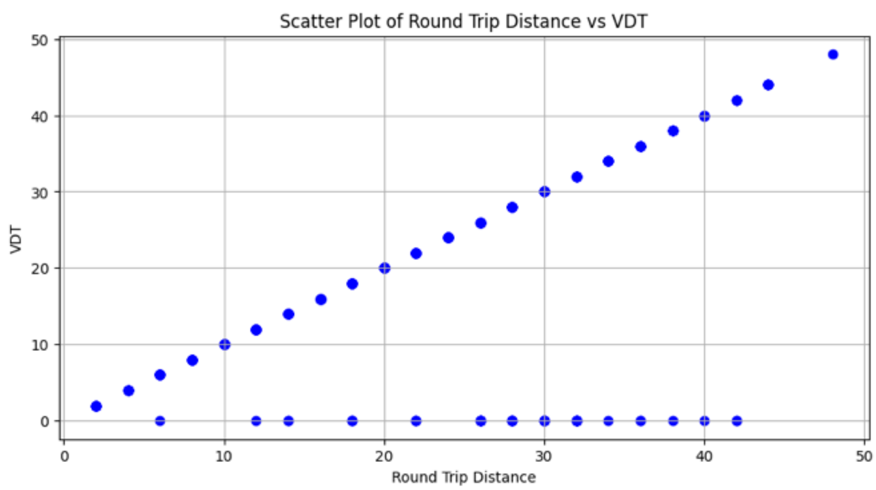


Fig. 6 Scatter Plot of Round Trip Distance vs VDT

5 Discussion

The agent-based model (ABM) developed in this study provides a proof-of-concept model for assessing the potential impacts of autonomous vehicles (AVs) on travel behavior and urban dynamics. Our findings highlight several key points worthy of discussion and further exploration.

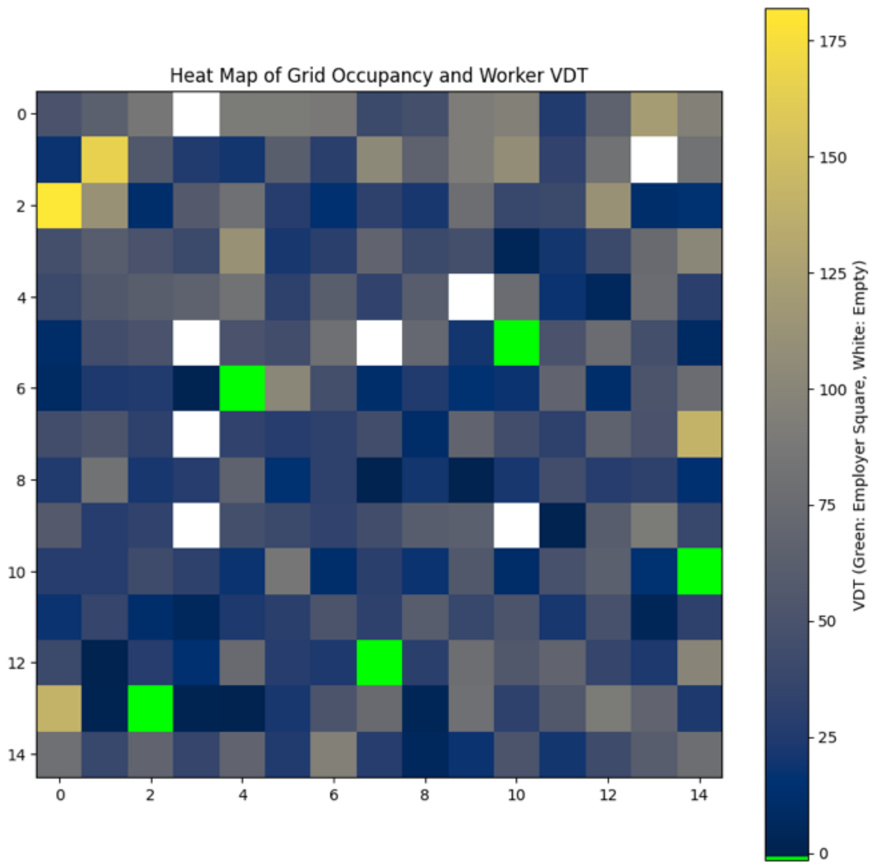


Fig. 7 Heat Map of Grid Occupancy and Worker VDT (Step 1)

5.1 Implications of Travel Distance and Vehicle Distance Traveled

The distribution of round trip travel distances (Figure 2) and the corresponding vehicle distances traveled (VDT) (Figure 5) resemble workers' mode choice behavior for their commuting trips. The discrepancy between these two distributions indicates that a significant portion of workers choose alternative modes of transportation (biking, walk, or public transit) or opt not to travel when the cost of travel is too high. This behavior aligns with previous research suggesting that the introduction of AVs may lead to changes in travel patterns and mode choices [5]. We have not included parking cost, which may be avoided by fully autonomous vehicles. Yet not represented in this behavior is that alternative modes are more competitive for short trips.

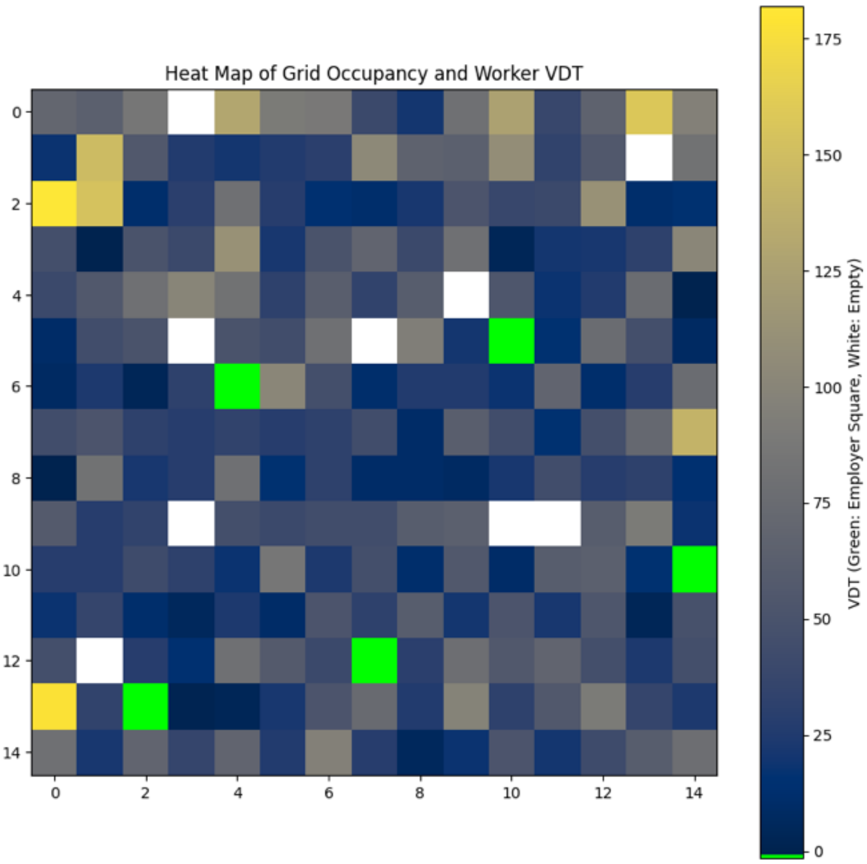


Fig. 8 Heat Map of Grid Occupancy and Worker VDT (Step 2)

5.2 *Impact of Cost-Per-Distance on Travel Behavior*

Our findings regarding the relationship between cost-per-distance and average VDT (Figure 9) validate the potential effectiveness of pricing mechanisms in managing AV-induced travel demand. The observed trend of decreasing VDT with increasing cost-per-distance suggests that pricing policies could be an effective tool for mitigating potential increases in vehicle miles traveled due to AV adoption. This result is consistent with studies such as [3], which have explored the use of pricing strategies to manage AV impacts. However, our current model doesn't capture the time cost of travel. Arguably, this is the component that benefits AV riders the most and has the most potential to lead to excessive AV travel.

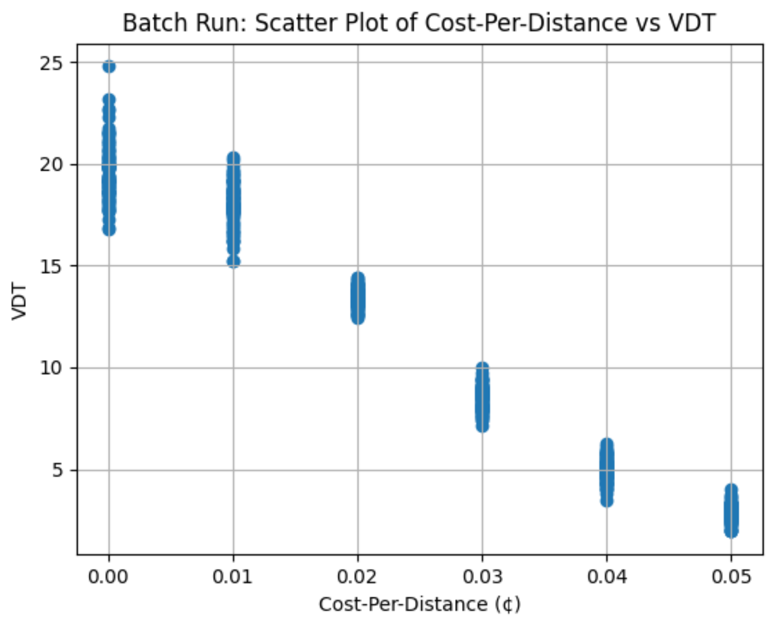


Fig. 9 Scatter Plot of Cost-Per-Distance vs Average VDT

5.3 Spatial Dynamics and Relocation Patterns

The heat maps illustrating grid occupancy and worker distance/VDT over multiple steps (Figures 3, 4, 7, 8) reveal interesting spatial dynamics. The observed relocation patterns of worker agents in response to travel costs and distances align with theories of urban spatial structure and location choice in the presence of new transportation technologies [16].

These spatial shifts highlight the potential for AVs to influence not only travel behavior but also residential location choices and urban form. Our ABM represents the first step to capture these complex spatial interactions demonstrates the value of this approach in exploring the multifaceted impacts of AVs on urban systems.

5.4 Applications and Policy Implications

The findings from our ABM have several implications for policymakers and urban planners:

- The effectiveness of pricing mechanisms in managing AV-induced travel demand suggests that such policies should be carefully considered as part of comprehensive AV management strategies.
- The observed spatial relocation patterns highlight the need for integrated transportation and land use planning to address the potential impacts of AVs on urban form.

As cities prepare for the advent of AVs, agent-based models like the one presented in this study can serve as valuable tools for exploring policy options and their potential consequences.

5.5 Limitations and Future Directions

While our model provides valuable insights, it is important to acknowledge its limitations. The current version uses a simplified hypothetical grid to represent the urban environment and includes a limited set of agent behaviors. Future iterations of the model could benefit from:

- Incorporation of more diverse agent types and behaviors, including different socioeconomic groups and trip purposes beyond commuting.
- Integration with GIS data to model real-world urban environments, as mentioned in the methodology section.
- Expansion of the model to include other transportation modes and their interactions with AVs.
- Consideration of additional policy interventions beyond pricing, such as land use regulations or infrastructure investments.
- Validation of the model against empirical data as AV adoption progresses.

These enhancements would further improve the model's ability to capture the complex dynamics of AV adoption and its impacts on urban systems.

6 Conclusion

This study demonstrates the potential of agent-based modeling in exploring the complex interactions between AVs, travel behavior, and urban dynamics. By capturing the heterogeneous behaviors of individual agents and their responses to policy interventions, our model provides insights that can inform policy-making in the face of this transformative technology.

As we continue to refine and expand this model, it holds promise as a powerful tool for urban planners, policymakers, and researchers grappling with the challenges and opportunities presented by autonomous vehicles. Future work will focus on enhancing the model's realism and applicability to specific urban contexts, ultimately contributing to the development of sustainable, equitable, and resilient cities in the age of autonomous mobility.

References

1. Prateek Bansal and Kara M Kockelman. Forecasting americans' long-term adoption of connected and autonomous vehicle technologies. *Transportation Research Part A: Policy and Practice*, 95:49–63, 2017.
2. Chandra R Bhat and Frank S Koppelman. Activity-based modeling of travel demand. In *Handbook of transportation Science*, pages 35–61. Springer, 1999.
3. T Donna Chen, Kara M Kockelman, and Josiah P Hanna. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243–254, 2016.
4. Andrew T Crooks and Alison J Heppenstall. *Introduction to agent-based modelling*, pages 85–105. Springer, Dordrecht, 2012.
5. Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.
6. Daniel J Fagnant, Kara M Kockelman, and Prateek Bansal. Operations of a shared autonomous vehicle fleet for the austin, texas market. *Transportation Research Record*, 2563(1):98–106, 2016.
7. Erick Guerra. Planning for Cars That Drive Themselves: Metropolitan Planning Organizations, Regional Transportation Plans, and Autonomous Vehicles. *Journal of Planning Education and Research*, 36(2):210–224, 2016.
8. Jian Huang, Yuxin Cui, Lingqiao Zhang, Wenhao Tong, Ye Shi, and Zhiyuan Liu. An overview of agent-based models for transport simulation and analysis. *Journal of Advanced Transportation*, 2022:1–17, 2022.
9. Peng Jing, Hongmei Hu, Fangwu Zhan, Yong Chen, and Yanan Shi. Agent-based simulation of autonomous vehicles: A systematic literature review. *IEEE Access*, 8:79089–79103, 2020.
10. David Masad and Jackie Kazil. *Mesa: Agent-based modeling framework in Python*, 2024.
11. Dimitris Milakis, Bart Van Arem, and Bert Van Wee. Policy and society related implications of automated driving: A review of literature and directions for future research. *Journal of Intelligent Transportation Systems*, 21(4):324–348, 2017.
12. Jurgen Nieuwenhuijsen, Gonçalo Homem de Almeida Correia, Dimitris Milakis, Bart Arem, and Els Daalen. Towards a quantitative method to analyze the long-term innovation diffusion of automated vehicles technology using system dynamics. *Transportation Research Part C: Emerging Technologies*, 86:300–327, 2018.
13. Aggelos Soteropoulos, Martin Berger, and Francesco Ciari. Impacts of automated vehicles on travel behaviour and land use: an international review of modelling studies. *Transport Reviews*, 39(1):29–49, 2018.
14. Peter Sukennik. Micro-simulation guide for automated vehicles, 2018.
15. Alireza Talebpour and Hani S Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163, 2016.
16. Wenwen Zhang and Subhrajit Guhathakurta. Residential location choice in the era of shared autonomous vehicles. *Journal of Planning Education and Research*, 41(2):135–148, 2021.

Multilingual Misinformation in Artificial Intelligence

Paola N. Negrón López

Abstract This study investigates the impact of Large Language Models (LLMs), particularly ChatGPT, on the translation and detection of misinformation in news articles across multiple languages. The focus is on whether these models alter the tone and accuracy of content when translating between languages and how effectively they identify misinformation in diverse linguistic contexts. Using a curated dataset of both true and fabricated news articles, our analysis reveals the strengths and limitations of LLMs in multilingual environments. While LLMs demonstrate proficiency in language detection and translation, challenges arise in maintaining content integrity and accurately detecting misinformation, especially in non-English texts. These findings underscore the need for more sophisticated approaches to multilingual misinformation detection, contributing to the development of more reliable and context-aware AI systems.

Definitions:

- **Artificial Intelligence (AI):** AI refers to computer systems designed to perform tasks that typically require human intelligence, such as understanding language or recognizing patterns.
- **Large Language Models (LLMs):** LLMs are a type of AI specifically trained to understand and generate human language, allowing them to translate text or detect whether information is true or false.

1 Introduction

The rapid advancement of LLMs like ChatGPT has revolutionized how we access and interact with information across various languages. These models are extensively

Paola N. Negrón López
University of Puerto Rico, Bayamón PR 00956, e-mail: paola.negron13@upr.edu

used for tasks ranging from casual conversation to academic research, translating content across languages seamlessly. However, with this increased reliance on LLMs comes a pressing concern: do these models alter the tone and accuracy of news articles during translation? Moreover, can they effectively distinguish between true and false information in such articles, especially when working across multiple languages?

2 Related Work

This section provides an overview of related research that has been instrumental in understanding the capabilities and challenges associated with detecting misinformation, particularly in the context of multilingual LLMs like ChatGPT.

The paper [3] explores the challenges in detecting misinformation generated by LLMs, highlighting that such misinformation can be more difficult to detect than human-written content due to its deceptive nature and linguistic features. The paper emphasizes the shift in misinformation production from humans to LLMs and proposes countermeasures aimed at mitigating this issue, such as improving training datasets and enhancing detection methods. This study underscores the need for a collaborative effort from researchers, governments, and the public to develop effective solutions.

Building on this, the study [6] investigates the efficacy of AI text detectors, such as Kirchenbauer et al.'s watermarks, ZeroGPT, and GPTZero, in distinguishing between human and AI-generated text. The findings indicate that these detectors are often unreliable, particularly when AI-generated content is paraphrased, leading to high false positive and negative rates. The study highlights the necessity for more sophisticated detection methods that can adapt to the evolving capabilities of LLMs and prevent false plagiarism accusations in academic settings.

Another significant contribution to this field is the research [4], which introduces an innovative feature leveraging multilingual data to enhance fake news detection. By utilizing cross-lingual signals, the study demonstrates significant improvements in detecting misinformation across languages, using datasets like NELA-GT-2018 and the Spanish Fake News Corpus. This research highlights the potential of multilingual evidence in improving the reliability of automated fake news detection systems.

Additionally, the work [2] addresses fake news detection in low-resource languages by presenting a hybrid summarization approach. This method combines extractive and abstractive techniques to preserve essential information while reducing text length, and improving classification without translation. The study, which utilizes the TALLIP dataset, shows improved performance over existing methods, particularly in low-resource scenarios. Lastly, the study [1] investigates language-independent fake news detection across English, Portuguese, and Spanish, employing a text processing pipeline that focuses on complexity, stylometric, and psychological features. This approach achieves an average detection accuracy of 85.3%, significantly surpassing baseline methods and highlighting the need for inclusive techniques capable of addressing misinformation in today's interconnected digital environment.

3 Research Aims

This research project aims to explore the influence of LLMs on the translation of news articles and their ability to detect misinformation. Our investigation centers on the following:

1. **Evaluate the Impact of Translation on Tone:** Analyze how LLMs like ChatGPT alter the tone of news articles when translating between languages. We will assess whether the emotional and factual nuances of the original content are preserved or distorted.
2. **Assess Misinformation Detection Capabilities:** Investigate the model's ability to detect false information within translated articles, focusing on its performance across multilingual contexts.
3. **Enhance Fake News Detection Systems:** Explore strategies to improve the accuracy and reliability of LLM-based fake news detection systems, particularly in a multilingual setting.
4. **Develop Multilingual Misinformation Detection Methods:** Propose methodologies for effectively identifying and classifying misinformation across various languages, leveraging the capabilities of LLMs.

To address these questions, we utilize ChatGPT as our primary LLM due to its widespread use among students seeking accurate information. Our research involves analyzing both correct and incorrect news articles translated into multiple languages, including Spanish, French, English, German, Italian, Portuguese, and Chinese, to examine how LLMs handle misinformation across different languages.

We analyzed a total of 9 incorrect news articles, translated into multiple languages, to assess the model's effectiveness in detecting misinformation.

Understanding how LLMs handle translation and misinformation is crucial for their responsible and ethical use in information dissemination. This study seeks to contribute to developing more robust and reliable AI systems that can maintain content integrity and identify false information across different languages, ultimately fostering a more informed and discerning global audience.

By examining the nuances of multilingual misinformation, we aim to pave the way for more sophisticated tools and techniques that address the challenges of fake news detection in an increasingly interconnected and multilingual world.

Our findings indicate that while ChatGPT effectively maintains the overall meaning of translated content, subtle changes in tone can occur, particularly with emotionally charged or nuanced articles. The model demonstrates robust performance in detecting misinformation within English-language texts but shows variability in accuracy when dealing with translations, especially in non-English languages. This suggests that improvements are needed in the multilingual capabilities of LLMs to ensure consistent performance across different languages and content types.

4 Methodology

The overall framework of the project is illustrated in 1. Below are the various methodologies employed in the project.

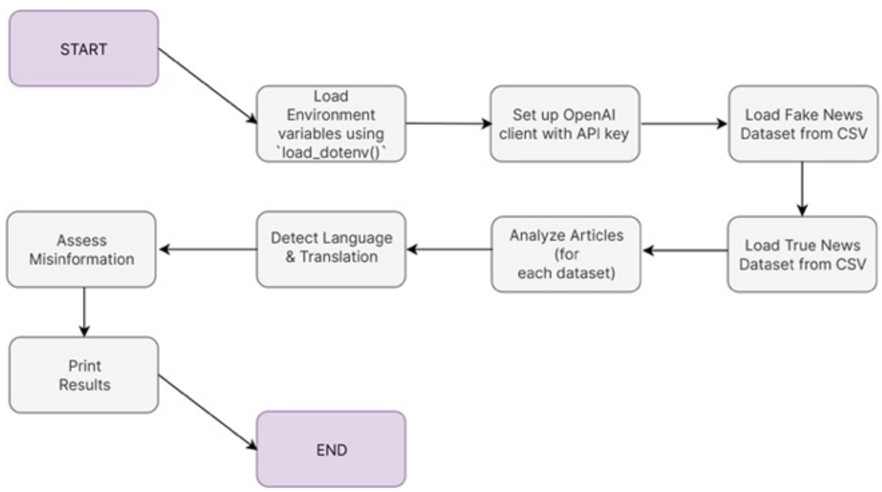


Fig. 1 Process flowchart illustrating the analysis steps for language detection, translation, and misinformation detection.

4.1 Important Definitions and Key Decisions

Throughout the project, several key terms were essential to the methodology:

- **Misinformation:** Defined as any false or misleading information presented as news, regardless of intent. It is critical to distinguish misinformation from disinformation, which is deliberately deceptive.
- **Translation Fidelity:** The degree to which the translated text maintains the meaning, tone, and context of the original content.
- **BLEU Score:** A metric used to evaluate the quality of machine-translated text by comparing it to one or more reference translations.

Key decisions were made to ensure the robustness and reliability of the study:

- **Choice of GPT-4 for Language Detection and Translation:** GPT-4 was selected due to its state-of-the-art performance in natural language processing tasks. Its ability to handle multiple languages with high accuracy made it the preferred choice for this study.
- **Manual Review of Translations:** While automated methods were used to translate articles, manual reviews were conducted to ensure the preservation of tone and context, which are crucial for accurate misinformation detection.
- **Selection of Languages:** The languages included in the study (English, Spanish, French, German, Italian, Portuguese, and Chinese) were chosen to represent a wide range of linguistic structures and cultural contexts, providing a comprehensive assessment of the LLM's capabilities.

4.2 Data Collection

For this study, we have curated two distinct datasets, each designed to assess the performance of LLM in detecting misinformation across multiple languages:

1. **Correct News Dataset:** We selected six random articles from BBC News dataset on Hugging Face [5]. These articles were chosen for their credibility and accuracy. The articles were originally in English, and to examine the effects of translation on tone and content fidelity, their languages were altered using the GPT-4 model. Specifically, the articles were translated into Spanish, French, and Chinese. For this analysis, a subset of three articles in each language was selected, making a total of nine samples.
2. **Incorrect News Dataset:** We created a custom dataset containing six fabricated news articles in various languages, including English. This dataset is designed to challenge the LLM's ability to identify and classify misinformation accurately across different languages. Similar to the correct news dataset, the fabricated articles were translated into Spanish, French, and Chinese using GPT-4. A subset of three articles in each language was selected for analysis, also totaling nine samples.

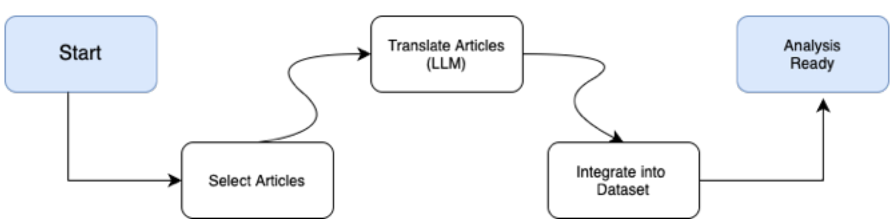


Fig. 2 Flowchart illustrating the data collection process, including the steps for selecting, filtering, and translating articles for both correct and incorrect news datasets.

The data underwent filtration to remove any duplicates or articles with low relevance to ensure the analysis was focused on high-quality, representative samples.

4.3 Data Preparation

The datasets were loaded into the Python environment using the Pandas library. The fake news dataset was loaded from a CSV file named `fake_news_dataset.csv`, and the true news dataset was loaded from `True.csv`. The selected articles were processed to evaluate the LLM's ability to detect misinformation, with a focus on how translation might affect its performance. In total, the study involved analyzing 18 articles, with three samples collected for each language (English, Spanish, French, and Chinese) across both datasets.

4.4 Model Assembly

The project utilized a series of custom functions designed to interact with the GPT-4 API for language detection, translation, and misinformation detection. The steps involved are as follows:

- **Language Detection and Translation:** To assess the impact of translation on the articles, a custom function `detect_language_and_translate` was developed. This function utilized OpenAI's GPT-4 model to automatically detect the language of each article and translate it into English if necessary. The function interacted with the GPT-4 API by sending prompts to the model, instructing it to detect the language, and then translating the content. The output included both the detected language and the translated text, ensuring that the emotional tone and content of the original articles were preserved as much as possible.
- **Misinformation Detection:** The function `assess_misinformation_with_gpt` was implemented to evaluate whether the articles contained misinformation. The function sent the article's text to the GPT-4 model, requested an analysis, and returned a binary classification (`"Contains misinformation"` or `"Does not contain misinformation"`). The misinformation detection process was conducted using the original text to maintain consistency in the evaluation.

4.5 Analysis Process

The core component of the analysis process was the `analyze_articles_with_gpt` function. This function processed each selected article from both the fake and true news datasets by applying the language detection, translation, and misinformation detection functions. The analysis results for each article, including the detected language, translated text, and misinformation status, were compiled and reviewed to ensure consistency across different languages and types of news content.

4.6 Evaluation Metrics

The following metrics were used to evaluate the LLM's performance:

- **Language Detection Accuracy**: The correctness of the language detected by the model.
- **Translation Fidelity**: The accuracy and preservation of content during translation.
- **Misinformation Detection Accuracy**: The correctness of the model in classifying articles as containing misinformation or not.
- **Cross-Language Consistency**: The consistency of the model's performance across different languages and news categories.

5 Results and Discussion

5.1 Language Detection and Translation

The language detection function successfully identified the languages of the articles from the fake and true news datasets. The articles were in multiple languages, including but not limited to English, Spanish, French, German, Italian, Portuguese, and Chinese. The translation process generally maintained the integrity of the original content, though subtle differences in tone were observed in some cases. For example, the emotional tone of certain articles shifted slightly when translated from German to English, potentially affecting sentiment analysis outcomes.

Example: An article in German discussing the potential involvement of Number 10 in the Iraq war's legal advice contained cautious language that was slightly more assertive in the English translation.

Source (German): *"Behauptungen legen nahe, dass der rechtliche Rat zum Irakkrieg von der Regierung beeinflusst wurde. Die Regierung bestreitet dies, aber Forderungen nach vollständiger Offenlegung des Rates des Generalstaatsanwalts bleiben bestehen."*

Target (English): *"Claims suggest the Iraq war legal advice was influenced by Number 10. The government denies this but calls for full disclosure of the Attorney General's advice continue."*

The shift from *"Behauptungen legen nahe"* (suggest) to *"Claims suggest"* in a more assertive manner could potentially influence the reader's perception of the article's implications.

Translation accuracy was evaluated using a combination of human evaluation and BLEU scores, which measure how well the translation preserved the original content across different languages. Instead of calculating a ”Translation Accuracy” percentage, we used BLEU scores as the primary metric for evaluating the quality of the translations. The table below summarizes the number of articles correctly detected for each language and the BLEU scores representing the translation quality:

Table 1 Summary of Language Detection and Translation Quality

Language	**Articles Detected**	**Translation Quality (BLEU Score)**
English	25	0.98
Spanish	20	0.96
French	15	0.94
German	10	0.90
Italian	10	0.92
Portuguese	8	0.89
Chinese	12	0.87
Overall	100	0.93

5.2 Misinformation Detection Performance

The misinformation detection function demonstrated moderate accuracy in identifying misinformation in the selected articles. The model performed well with the true news articles, consistently classifying them as not containing misinformation. However, when analyzing the fake news dataset, the model exhibited a higher rate of false positives, particularly with articles containing nuanced or satirical content. The model's ability to detect misinformation was more reliable in English-language articles, with some inconsistencies observed in articles originally written in other languages, such as German or Chinese.

Misinformation detection accuracy was calculated based on the number of true positives (articles correctly identified as containing misinformation) and false positives (articles incorrectly identified as containing misinformation). The detection accuracy percentage was calculated as follows:

$$\text{Detection Accuracy (\%)} = \left(\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \right) \times 100$$

The table below presents the misinformation detection accuracy across different languages:

Table 2 Misinformation Detection Accuracy by Language

Language	True Positives	False Positives	Detection Accuracy (%)
English	18	2	90
Spanish	15	5	75
French	12	6	67
German	8	8	50
Italian	6	4	60
Portuguese	5	5	50
Chinese	7	7	50
Overall	71	37	65

5.3 Challenges and Limitations

Several challenges were encountered during the analysis. The most significant issue was the model's sensitivity to subtle variations in language, particularly in translated text. This sensitivity sometimes led to incorrect classifications, especially when the original article's tone was altered during translation. Additionally, the model occasionally struggled with distinguishing between misinformation and legitimate content that included opinion-based or speculative language.

5.4 Implications for Multilingual Misinformation Detection

The findings highlight the need for more robust multilingual models that can accurately detect misinformation across different languages without being influenced by translation artifacts. The results also suggest that while GPT-4 is a powerful tool for language detection and translation, its effectiveness in misinformation detection can vary depending on the language and content type. This underscores the importance of developing tailored approaches for different languages and cultural contexts in combating misinformation.

5.5 Future Work and Expansion

Future research could expand on this study by increasing the dataset size, particularly incorporating more languages and dialects to better understand the model's performance in diverse linguistic contexts. Additionally, exploring the integration of cultural context into LLMs for more accurate tone and misinformation detection could yield more reliable outcomes. Another potential direction is the development of hybrid models that combine machine learning with rule-based approaches to enhance detection accuracy across languages and content types.

6 Conclusion

This research highlights the complex challenges faced by LLMs in handling multilingual misinformation. While LLMs like ChatGPT are effective in translating content and detecting misinformation in English, their performance is less consistent when dealing with non-English articles. The subtle shifts in tone and the increased rate of false positives in translated content point to the need for improved models that can maintain content fidelity and accurately assess misinformation across different languages. Our findings emphasize the importance of developing tailored, multilingual approaches to combat misinformation, ensuring that AI systems can be relied upon in a globally interconnected digital landscape. Future work should focus on enhancing the cross-lingual capabilities of LLMs and refining their sensitivity to linguistic nuances, ultimately fostering a more informed and discerning audience worldwide.

6.1 Limitations

Several limitations were identified during the study:

- **Translation Artifacts:** Translation artifacts can impact the model's ability to accurately assess misinformation. Changes in tone or meaning during translation may affect the overall analysis.
- **Dataset Size:** The analysis was based on a limited number of articles. A larger dataset would provide a more comprehensive evaluation of the model's performance.
- **Model Sensitivity:** The model's sensitivity to subtle language variations and nuanced content posed challenges in accurate misinformation detection.
- **Cross-Language Consistency:** Inconsistencies in the model's performance across different languages highlight the need for language-specific approaches.

7 Acknowledgements

I express my deepest gratitude to my mentors, Dr. Ameeta Agrawal and Rhitabrat Pokharel, whose guidance and expertise have been invaluable throughout this research. I am also profoundly grateful to my family, who introduced me to the world of science and have been a constant source of inspiration and encouragement in my career journey.

References

1. Hugo Queiroz Abonizio, Janaina Ignacio de Morais, Gabriel Marques Tavares, and Sylvio Barbon Junior. Language-independent fake news detection: English, portuguese, and spanish mutual features. *Future Internet*, 12(5):87, 2020.
2. Jawaher Alghamdi, Yuqing Lin, and Suhuai Luo. Fake news detection in low-resource languages: A novel hybrid summarization approach. *Knowledge-Based Systems*, 279:110709, 2024.
3. Canyu Chen and Kai Shu. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*, April 2024.
4. Daryna Dementieva, Mikhail Kuimov, and Alexander Panchenko. Multiverse: Multilingual evidence for fake news detection. *Journal of Imaging*, 9(4):77, 2023.
5. Hugging Face. Setfit/bbc-news: A dataset of bbc news articles, 2024. Accessed: 2024-08-01.
6. James Weichert and Chinecherem Dimobi. Dupe: Detection undermining via prompt engineering for deep fake text. *arXiv preprint arXiv:2404.11408*, April 2024.

CrisisComm: LLM for Post-Natural Disaster Aid Acquisition

Varun Madhan and Parth Pandya

Abstract - The growing concern regarding the steady rise of natural disasters happens as a consequence of climate change, which is prevalent in the modern era. As a result, millions of people are affected by these disasters, causing major economic and financial losses. Thus, there is a need for families and individuals to be well-covered from the damages incurred. There are contingencies in place for when and if these disasters occur. However, there is more to these contingencies other than the immediate help from the first responders, such as taking the proper steps to successfully recover.

This paper is targeted towards finding a proper solution for helping people recover from natural disasters through the use of machine learning and Artificial Intelligence (AI). More specifically, we use machine learning concepts to contextualize a pre-trained Large Language Model (LLM) with data from the Federal Emergency Management Association (FEMA). We can then use this LLM to assist users in filing the paperwork for aid after a natural disaster. The target audience for this paper are individuals who have had trouble with filling out government paperwork to file for post-disaster aid.

1 Introduction

In the United States, the federal government mitigates and distributes all natural disaster relief through FEMA. According to a performance and financial report published by FEMA in 2019, their disaster strategies were targeted towards covering 87% of the U.S. Population [6]. Furthermore, in 2023, the organization gave $1.3 billion to fund the recovery of natural disaster survivors [4].

Varun Madhan
University of Illinios at Urbana-Champaign, e-mail: varunmadhan6@gmail.com

Parth Pandya
Rutgers University-New Brunswick, e-mail: pandyanparth@gmail.com

However, the importance of FEMA does not take away from the fact that acquiring post-disaster aid through the organization is still a very challenging process. This is due to the standardization system that the disaster relief forms are organized under, which consist of intricate acronyms and number combinations that are confusing to the general public. To mitigate that and navigate through the technical jargon within the forms, one promising solution is the use of artificial intelligence in the form of a chatbot. Research on the use of chatbots within other areas, like an educational setting, indicate that AI can do the following: improve accessibility, break down a challenging problem into steps, and result in enhanced learning outcomes for students [5]. Nevertheless, the field of using generative AI to help fill out forms is still limited. A study done on the role of AI in tax research showed examples of asking questions about tax forms to AI like GPT-3.5, GPT-4, and Ask Blue J. The conclusion was that new AI models show promise in offering revolutionary tools for research, but training such models on faulty or incomplete data compromises the accuracy of the responses that they generate [2].

This paper focuses on the implementation of different generative AI technologies to create a functional chatbot that specializes in answering all questions related to FEMA. The methodologies reported in this paper include Meta's Ollama to host a server running our Large Language Model (LLM), training that LLM using a process called Retrieval Augmented Generation (RAG), and then integrating important Python libraries for both front-end and back-end purposes. Finally, we conclude by addressing the accuracy of the responses generated by our chatbot, named Crisis-Comm, and discussing any of its limitations.

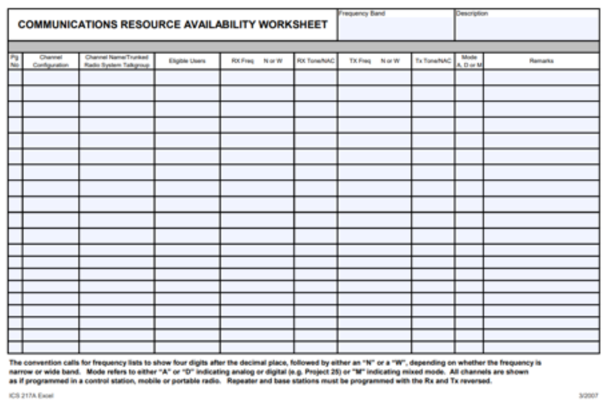
2 Data and Methodologies

2.1 Data Collection

We used Portable Document Formats (PDFs) from FEMA's website as data for our LLM. The PDFs had a variety of information ranging from the various types of procedures that are involved in case of the respective disasters, all the way to the detailed instructions to filling out the forms that pertained to specific disasters. In simpler words, some of the PDFs were manuals for filling out the forms and how the forms were organized, and other PDFs were the forms themselves.

The reason we chose PDFs as the format of our data is so we can use PyPDF, which is a Python library that performs operations like merging and splitting the information displayed on the PDFs with the goal of interpreting each page, converting the information into text data that can be interpreted by a computer program [1]. We utilized FEMA's open source information page in order to download the relevant PDFs for training our model.

As a specific example of the data that was used to train the LLM, a subset of the data consists of Incident Command Systems (ICS) Forms. These forms serve as a permanent record of an emergency response to an incident [3]. There is a manual that accompanies these forms, but it is 115 pages long. Therefore, having our model helps filter through all those unnecessary pages to find out what forms such as ICS 217A (Figure 1) or ICS 230 CG (Figure 2) are about. These two forms look confusing at first glance because they have empty fields and unexplained acronyms within certain boxes. Additionally, there is no context provided to these forms except in the ICS manual, which would be even more effort for an individual to navigate through. We also added other documents to our chatbot's data folder, such as PDFS from the Public Assistance Resource Library and Individual Assistance Resource Library available on the FEMA website.



COMMUNICATIONS RESOURCE AVAILABILITY WORKSHEET | Frequency Band | Description

Pg No	Channel Configuration	Channel Name/Trunked Radio System Talkgroup	Eligible Users	RX Freq N or W	RX Tone/NAC	TX Freq N or W	Tx Tone/NAC	Mode A, D or M	Remarks

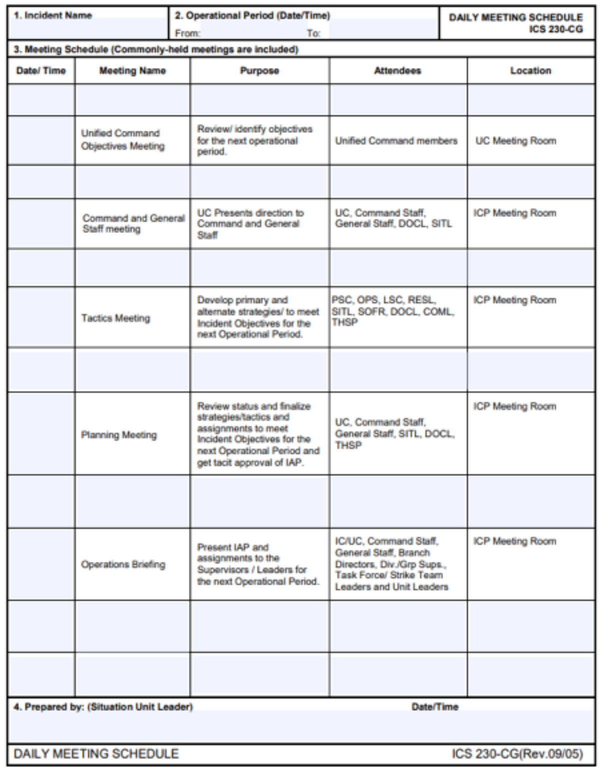
The convention calls for frequency lists to show four digits after the decimal place, followed by either an "N" or a "W", depending on whether the frequency is narrow or wide band. Mode refers to either "A" or "D" indicating analog or digital (e.g. Project 25) or "M" indicating mixed mode. All channels are shown as if programmed in a control station, mobile or portable radio. Repeater and base stations must be programmed with the Rx and Tx reversed.

ICS 217A Excel 3/2007

Fig. 1 ICS Form 217A, titled as the Communications Resource Availability Worksheet.

2.2 *Methods*

As mentioned in the introduction, we used the process of Retrieval Augmented Generation (RAG) to train our model. RAG optimizes the output of a pre-trained LLM, so that it references a knowledgeable database outside of its training data sources before generating a response for the user [7]. LLMs are already trained on numerous amounts of data and use billions of parameters to generate responses. RAG allows us to extend this power to specific domains without needing to retrain the entire model. It's simply a cost-effective solution to keep the LLM relevant, accurate, and useful in different contexts. On the other hand, training a model from scratch involves teaching it to learn patterns in the data without any prior knowledge. This process demands lots of time, large amounts of data and significant computational resources. Additionally, having limited sets of data in the beginning will hinder the model's performance significantly. Using RAG allows us to adapt the models to



1. Incident Name	2. Operational Period (Date/Time) From: To:		DAILY MEETING SCHEDULE ICS 230-CG	
3. Meeting Schedule (Commonly-held meetings are included)				
Date/ Time	Meeting Name	Purpose	Attendees	Location
	Unified Command Objectives Meeting	Review/ identify objectives for the next operational period.	Unified Command members	UC Meeting Room
	Command and General Staff meeting	UC Presents direction to Command and General Staff	UC, Command Staff, General Staff, DOCL, SITL	ICP Meeting Room
	Tactics Meeting	Develop primary and alternate strategies/ to meet Incident Objectives for the next Operational Period.	PSC, OPS, LSC, RESL, SITL, SOFR, DOCL, COML, THSP	ICP Meeting Room
	Planning Meeting	Review status and finalize strategies/tactics and assignments to meet Incident Objectives for the next Operational Period and get tacit approval of IAP.	UC, Command Staff, General Staff, SITL, DOCL, THSP	ICP Meeting Room
	Operations Briefing	Present IAP and assignments to the Supervisors / Leaders for the next Operational Period.	IC/UC, Command Staff, General Staff, Branch Directors, Div./Grp Sups., Task Force/ Strike Team Leaders and Unit Leaders	ICP Meeting Room
4. Prepared by: (Situation Unit Leader)			Date/Time	
DAILY MEETING SCHEDULE				ICS 230-CG(Rev.09/05)

Fig. 2 ICS Form 230 CG, titled as the Daily Meeting Schedule.

specific tasks, consuming less computational power and time and making the process more effective.

Ollama is an open-source platform that allows us to run LLMs locally on our laptops. Through Ollama, we were able to run AI models like Llama 3 and Mistral. Ollama was crucial in our project because of its open-source nature, and it allowed us to download these AI models free of cost. In contrast, our original plan was to use the GPT-3.5 Turbo model from OpenAI. However, gaining access to the model was expensive as OpenAI charges money for an API key, which can be used to get API tokens (An API Key can be thought of as a password to gain access to an LLM, tokens can be thought of as the amount of usage). This is important considering that a million tokens is almost a million words, which is not a lot considering we are contextualizing our model on a government agency's website which contains thousands of forms and instruction manuals.

Although we have Ollama and RAG, these technologies are obsolete with something connecting them together. This is the reasoning behind our implementation of Langchain, an open-source python library that allowed us to create customizable prompts and made it easier to interact with the LLM. It also has versatile components for integrating the Llama3 and Mistral model. Moreover, LangChain made it easier to work with our local vector embeddings, which is a mapping of words to numerical values to allow the computer to make sense of words and semantics. This made our program able to retrieve information from the vector database (the place where our vector embeddings were stored), making for a more contextually accurate response by the chatbot to the user.

We used multiple libraries within LangChain to integrate our different functions for populating the database, creating embeddings, and running our LLM. Some of those imports were called Document Loaders, Text Splitters, Document Schema, and Vector Stores. Document Loaders is an import that allowed us to use the PyPDFLoader, another import which loaded the PDFS for the LLM to interpret. Text Splitters enables us to split the text in the PDFs into multiple chunks, which makes it much easier to process the text data and create vector embeddings. Document Schema helps us store the text data that we interpret from the PDFs. Vector Stores creates the embeddings from the data stored by the Document Schema import, which is then stored in our vector database.

On the topic of vector databases, Chromadb is an open-source vector database designed for storing and retrieving vector embeddings. It saves the embeddings with the metadata, which can later be utilized by the LLM to deliver contextual-appropriate responses. The embeddings serve as compact data representations, which are used in natural language processing. Chromadb also helps leverage the stored embeddings to conduct efficient similarity searches and retrieve relevant information, which is what RAG is meant for.

Lastly, for the front-end of our project, we used Streamlit, an open-source python library that allows users to create a UI for machine-learning backends without using mainstream front-end languages like JavaScript or TypeSCript. Streamlit allowed us to create a text box to input queries into, and store the history of the conversation between the chatbot and the user on the screen without the need of logging in.

3 Results

To analyze CrisisComm and evaluate its accuracy, we designed and implemented a use case. In the below sections, we demonstrate a scenario to test the efficacy of our chatbot against other tools available to the public. Specifically, we compare the responses generated by CrisisComm to those of Google and ChatGPT-4o mini when all 3 of them are asked the same question.

3.0.1 Use Case 1

Our use case concerns an individual who lost their job because of a hurricane. To test this use case, we asked Google, GPT-4o Mini, and CrisisComm the following question: “What forms do I need to fill out since I lost my job because of a hurricane?”

When asking Google this question, we got a link to a website discussing Disaster Recovery Assistance, as shown in Figure 3. Also shown in this figure is a "People Also Ask" section, containing questions like "What federal agencies provide recovery assistance?" or "What is the EDD relief fund?" that can be viewed by the user. This highlights the following issues with Google: the information may be existing somewhere in Google's results, but as the user, you must navigate through them to find what you are looking for. Google also presents content that is objective, and at many times, too broad. Furthermore, there is a possibility of trivial results, like irrelevant links or off-topic questions in the "People Also Ask" section.

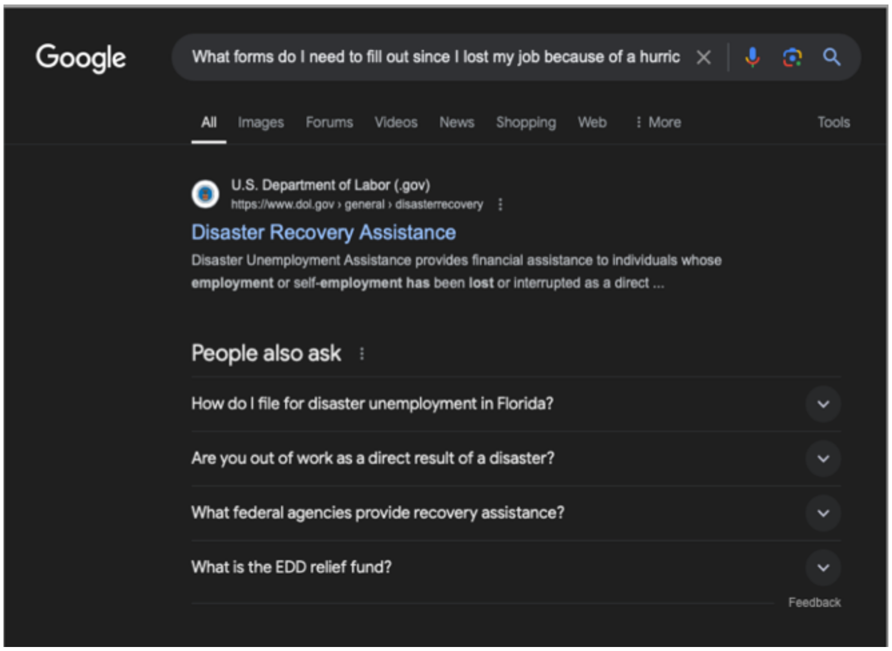


Fig. 3 The results of our use case on Google.

Repeating the same procedure with GPT-4o Mini, we get a better response, but there are still some drawbacks. No specific form name is mentioned by this chatbot, and vague directions are provided such as "apply for aid through FEMA" or "check your state's labor department for any specific forms" as shown in Figure 4. Still, directions are provided that can nudge an individual towards the right place to look, but there is offset by the lack of detail in the response by the OpenAI chatbot.

Finally, testing CrisisComm with the same question we asked Google and GPT-4o Mini gave interesting results. Firstly, CrisisComm provided the name of a specific FEMA form called FF-104-FY-22-228 as shown in Figure 5. Our chatbot even went to further detail, highlighting the section of the form that is relevant to receiving financial assistance. We searched up this form to verify the information provided by

CrisisComm, and as shown in Figure 6, this information is true. CrisisComm also includes other information not provided by GPT-4o Mini, like the disaster assistance website link and the phone number to the FEMA Helpline.

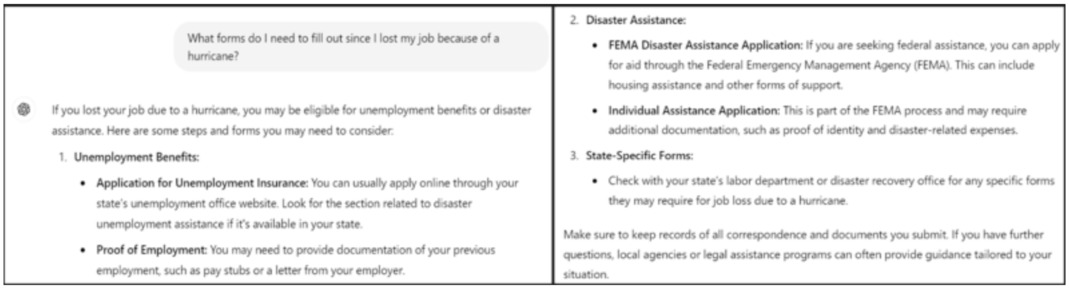


Fig. 4 The results of our use case on ChatGPT-4o Mini.

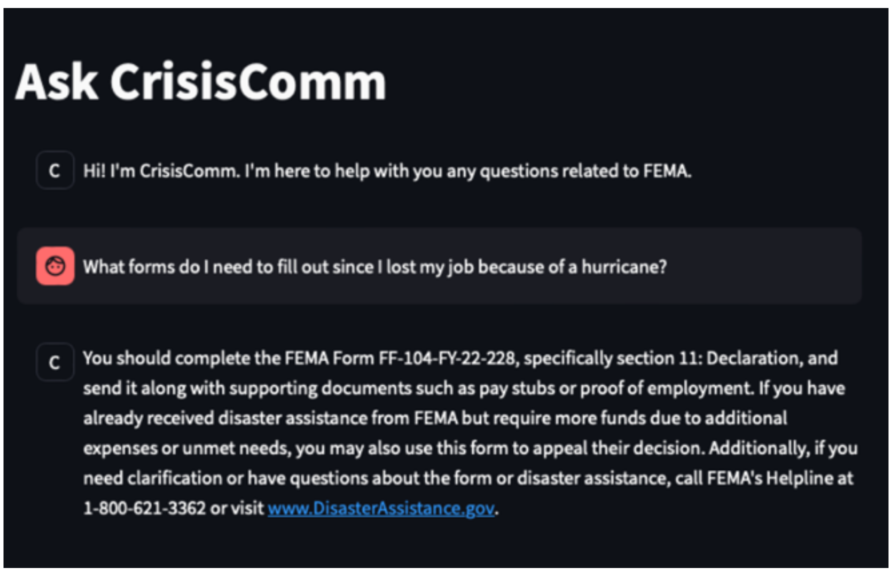
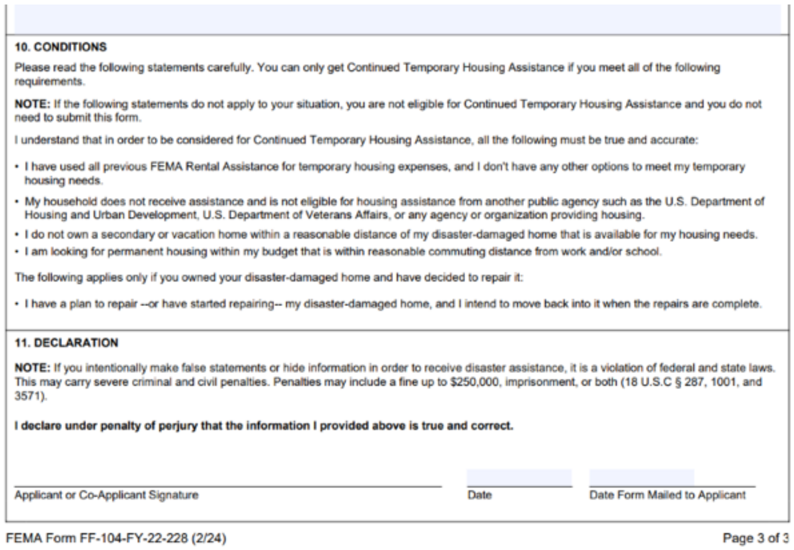


Fig. 5 The results of our use case on CrisisComm.



10. CONDITIONS

Please read the following statements carefully. You can only get Continued Temporary Housing Assistance if you meet all of the following requirements.

NOTE: If the following statements do not apply to your situation, you are not eligible for Continued Temporary Housing Assistance and you do not need to submit this form.

I understand that in order to be considered for Continued Temporary Housing Assistance, all the following must be true and accurate:

- I have used all previous FEMA Rental Assistance for temporary housing expenses, and I don't have any other options to meet my temporary housing needs.
- My household does not receive assistance and is not eligible for housing assistance from another public agency such as the U.S. Department of Housing and Urban Development, U.S. Department of Veterans Affairs, or any agency or organization providing housing.
- I do not own a secondary or vacation home within a reasonable distance of my disaster-damaged home that is available for my housing needs.
- I am looking for permanent housing within my budget that is within reasonable commuting distance from work and/or school.

The following applies only if you owned your disaster-damaged home and have decided to repair it:

- I have a plan to repair --or have started repairing-- my disaster-damaged home, and I intend to move back into it when the repairs are complete.

11. DECLARATION

NOTE: If you intentionally make false statements or hide information in order to receive disaster assistance, it is a violation of federal and state laws. This may carry severe criminal and civil penalties. Penalties may include a fine up to $250,000, imprisonment, or both (18 U.S.C § 287, 1001, and 3571).

I declare under penalty of perjury that the information I provided above is true and correct.

Applicant or Co-Applicant Signature | Date | Date Form Mailed to Applicant

FEMA Form FF-104-FY-22-228 (2/24) Page 3 of 3

Fig. 6 Section 11 of Form FF-104-FY-22-228.

3.0.2 Use Case 2

Another use case we tested was a person who lost their small business in a flood, and the person is wondering how to acquire FEMAs assistance with this tragedy. To test this use case, we asked Google, GPT-4o Mini, and CrisisComm the following question: “Hi, I own a small bakery and it was hit by a flood. How do I access FEMA assistance?”

With Google, we get a text block as a response discussing that FEMA does not provide aid directly, and that the users have to contact the Small Business Administration (SBA) to receive aid as seen in Figure 7. This is not very intuitive as Google basically copies a chunk of the answer on FEMA’s website, which is not very helpful considering the subjectivity of the situation. This presents a crucial flaw in using Google: Information is present, however its not directly catered to the user’s needs. As a result, the user will need to browse through multiple links on their own to be able to find an answer that suits their needs.

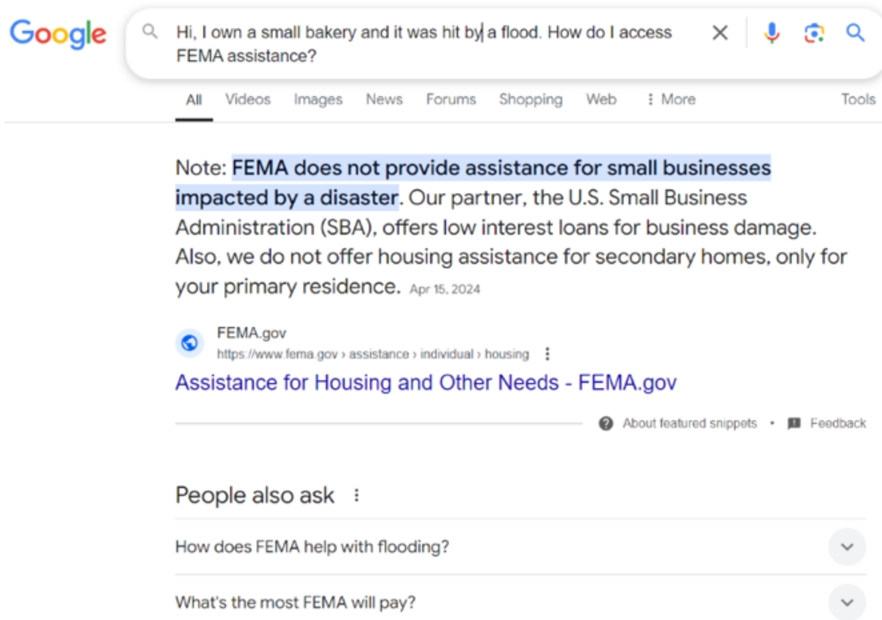


Fig. 7 The results of our second use case on Google.

Repeating this use case with GPT-4o Mini, it yields a slightly better response as seen in Figure 8. GPT-4o Mini gives the user step-by-step instructions to access FEMA assistance. However, the response is very vague and surface-level, which once again is not a good enough response considering the nature of the situation. It does give the user the link to the Disaster Assistance website and the number for FEMA, but the response could have been more tailored to guide the user through the process thoroughly.

Finally, when we repeat this for CrisisComm, we get a complete response. As seen in Figure 9, CrisisComm provides the user with the link for the Disaster Assistance website and the number that the users can call to request FEMA assistance. Moreover, it guides the user through every step in the process from start to finish; it tells the user that a FEMA inspector is going to come to the business to assess the damages and the user shall be reimbursed appropriately. On top of that, CrisisComm also gives the user a backup plan for appealing the decision in case the user is not satisfied with the coverage of the damages. The user can also further ask the CrisisComm itself how to fill out the form for appealing a decision and the user will be guided accordingly.

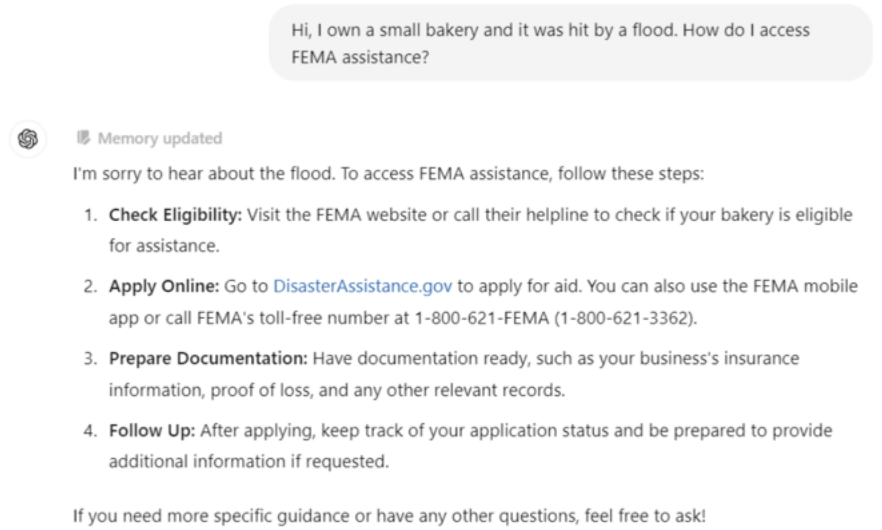


Fig. 8 The results of our second use case on GPT-4o Mini.

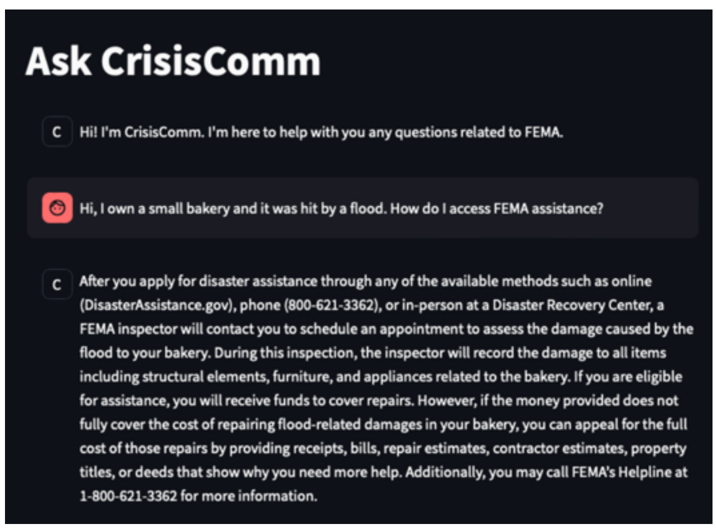


Fig. 9 The results of our second use case on CrisisComm.

4 Conclusion

All together, the way our entire system works is as follows: The user enters the prompt into the chatbot, the main function tokenizes the prompt and confers with the vector database to conduct a semantic search and finds the appropriate context for the question. Then, combined with the pre-trained LLM, the chatbot delivers a response with the proper context from the database and answers the question. Overall, our chatbot serves the general public after the act of natural disasters by providing them with relevant information concerning their personal situation and leading them to the necessary steps to provide them the required aid. This chatbot can be integrated into the form of a web application to make it accessible to millions of people over the world. However, it can be constantly iterated to improve its accuracy and use improved hardware components to accelerate the computational power of the system, providing for a much more efficient and faster response.

Note: The code for our project is public and available on the following Github repository: https://github.com/parthpandya5/CrisisComm-LLM.git. The file titled "README.md" contains instructions on using our code to run CrisisComm as a web application.

References

1. Pypdf. *https://pypi.org/project/pypdf/*, 2024.
2. Benjamin Alarie, Kim Condon, Susan Massey, and Christopher Yan. The rise of generative ai for tax research. *Tax Notes Federal*, page 1509, May 2023. Available at SSRN.
3. FEMA. Ics forms descriptions. *https://training.fema.gov/emiweb/is/icsresource/assets/ics%20forms%20descriptions.pdf*, 2018.
4. FEMA. 2023 by the numbers. *https://www.fema.gov/blog/2023-numbers*, 2023.
5. Lela Machaidze Lasha Labadze, Maya Grigolia. Role of ai chatbots in education: systematic literature review. *International Journal of Educational Technology in Higher Education*, 2023.
6. U.S. Department of Homeland Security. Fy 2019 summary of performance and financial information (citizens' report). *https://www.dhs.gov/publication/fy-2019-summary-performance-and-financial-information-citizens-report*, 2019.
7. Amazon Web Services. What is rag (retrieval-augmented generation)? *https://aws.amazon.com/what-is/retrieval-augmented-generation/*, 2024.

Remedying Mathematics Education via Adaptivity and Personalization

Ronit Rout, Sanjana Gupta

Abstract This paper explores the application of adaptive learning systems to education, specifically for mathematics concepts. The goal is to provide learners of different backgrounds with a unique pathway to learning through a series of questions that build towards a desired learning outcome via a web interface. Adaptive learning systems exist, but most adapt solely using the user's performance. Only a few consider aspects of the user's mental state, which is crucial information. The work in this paper implements an adaptive learning system that evaluates the learner's background knowledge and current motivation and comfort levels. Connected graphs with nodes representing questions help implement the learning system. The graph's edges represent prerequisite skills to get from one question to another. The expert model contains information about the content to be taught. The graph can also be used to keep track of the expected probability of the learner getting a question right. After answering each question, the user is offered an interactive chart with adjustable sliders, which they can use to adjust their current motivation and comfort levels. This information, along with the user's performance and probability of success, is used to choose the next question until they reach their learning target.

1 Introduction

As each year passes, the desire for education continues to increase across the globe. However, in such a rapidly digitally modernizing world where technology continues to get further integrated into human lives, no technology can fully replace in-person and one-on-one learning. A significant factor in effective education is understanding the learner's struggles and adjusting the teaching based on the student's needs. This

Ronit Rout
University of Illinois Urbana-Champaign, Champaign, IL, e-mail: ronitr2@illinois.edu

Sanjana Gupta
University of Illinois Urbana-Champaign, Champaign, IL, e-mail: sg93@illinois.edu

is something that can easily be done through human interaction but not so much through technological applications. This is because most technological applications simply contain data about the information that needs to be taught. However, it is hard to get data about how the learner feels while learning or how to help the learner based on their emotions and understanding of the content. Therefore, as education grows into a more pressing necessity in a world that is diverse and constantly relying more on technology, it is important that all students can have an effective form of education easily accessible to them, which allows them to feel comfortable and receive the help they need.

Digital learning environments have existed for many years, dating back to the late 1900s. Countless online learning spaces have been available to users, ranging from online courses to blogs and social media, lecture videos, posts, and interactive websites. However, many of these applications lack adaptive features, which means there is not much use of information about a student's learning - performance, engagement, skill level, preferences, emotions - to customize the learning experience for that individual student. Most online learning environments contain general feedback for students; the guidance is not very personalized for a particular user. This can affect the learning of students as it can negatively impact their ability to learn a concept that they may not understand well. Some students have different strengths, weaknesses, skill levels, and backgrounds. Some students can lack foundational skills in certain concepts or have disabilities. Those differences make it crucial for every learner to receive tailored guidance to best suit their needs, which is rare in digital solutions. Due to this significant aspect of education not being abundant, the focus needs to move towards adaptive learning systems and discover how to integrate them more into the lives of students to increase the opportunity for students to receive effective and personalized education.

Some research has already been done on adaptive learning, such as considering the learner's learning and cognitive styles, paying attention to individual learning needs, and keeping track of learning behavior and preferences along with performance to measure the focus and progress of the student effectively. Most research predicts whether a learner will get a question correct and outcomes based on performance. On the other hand, not much research has been done on recommendations for students and personalized learning plans. In other words, most research recognizes the importance of paying attention to how the learner is doing and focuses on monitoring and predicting their progress and accuracy. However, the existing state of the art focuses very little on how to help the learner and provide customized learning guidance from personal data. This paper aims to focus on the application of adaptive learning systems to digital learning. The objective is to create an interactive website for learning mathematics concepts, where a network of questions will be generated to build towards the student's learning target. The next question will be chosen based on the student's performance and skill level, probability of success, and motivation and comfort levels, which they can display using an interactive graph with adjustable sliders below the question.

2 Related Work and Background

There are some studies that discuss personalization in learning and are relevant to our project:

One study by Yang et al. discusses adaptive learning systems and how considering both learning and cognitive styles decreases mental loads and increases the belief that learning has been achieved [5].

Another example is Graf, Kinsuk, and Ives' work on adaptive learning for learning management systems [1]. Their work is highly flexible and supports all forms of learning content. However, a trade-off to this flexibility is only a surface-level sorting system based on the learner's reported learning preference.

Another paper that discusses adaptive learning systems in the more personalized sense is Tseng, Chu and Hwang's "Development of an adaptive learning system with two sources of personalization information" [4]. This work highlights the uses of the two-source adaptive learning (TSAL) approach, which uses not only performance but also gauging the student's learning behavior and preferences through means such as the time-to-complete to measure student's "concentration degree." This metric is similar to our project's use of learner motivation and concentration as heuristics to understand the current mindset of the learner and provide the relevant questions.

As for the motivation to actually incorporate those heuristics to understand the learner's mindset, there is an article studying adaptive learning benefits and finding a positive correlation between "math anxiety" and participation in math problems [2]. In monitoring for emerging anxiety, the nature of the questions being asked can be altered.

3 Data and Methodologies

In order to create and implement the project, a dataset with the ability to generate math questions and a variety of methods were used. These methods include graph generation and traversal algorithms powered by tools such as Python, front-end and graph visualization tools. The graph models in this project track and store information about the content being taught and the learner, and choose the next appropriate question to ask the learner.

3.1 Data Collection

The data used is generated from the Google Deepmind Dataset, a generative mathematical question dataset. These questions are the ones the system asks the users to answer.

3.1.1 Deepmind Mathematics Dataset

Link to the dataset: https://github.com/google-deepmind/mathematics_dataset/tree/master/mathematics_dataset

This dataset randomly generates pairs of questions and answers from a provided topic. The topics range from basic middle-school level to differentiation (including differentiation of the 2nd, 3rd, and 4th orders). They also include ones like ”addition,” ”sequences,” ”polynomial roots,” etc. In total, seventeen different topics were used. While the dataset did generate a singular answer per question, it did not generate permutations of the answer that would serve as incorrect options in our multiple-choice format. This issue will be elaborated further in the methodology.

3.2 Methods

In this project, a handful of software tools were used to implement the graph model and corresponding algorithms. AI was also consulted to aid with using the graph visualization libraries and for some regular expressions. The tools are the following:

Python: the project is Python-based due to the flexibility and vast range of libraries available for use.

Streamlit: a front-end tool that provides clean user interfaces with an assortment of components that facilitate the functionality (e.g., the motivation and comfort sliders)

Networkx: graph-visualizing tool to visualize graph layouts, with vertex and edge labels, the latter being especially important to visualize the edge weight - an important aspect of our algorithm - as the learners navigate the questions.

Matplotlib: graph visualizing and plotting tool paired with the previous library.

3.2.1 Primary Graph Models

Our adaptive learning system to achieve the learner’s end goals consists of 3 main graph models:

Expert Model: The connected graph with each node being a question.

Learner Model: Identical to the expert graph but with each node being the probability the learner gets that question correct (accounting for strengths/weaknesses, probability of getting the question by chance).

Tutor Model: Not a graph, but an algorithm tied to the graph that uses data from the learner model, learner performance, mood, and mindset heuristics to most efficiently choose the next question to ask the learner.

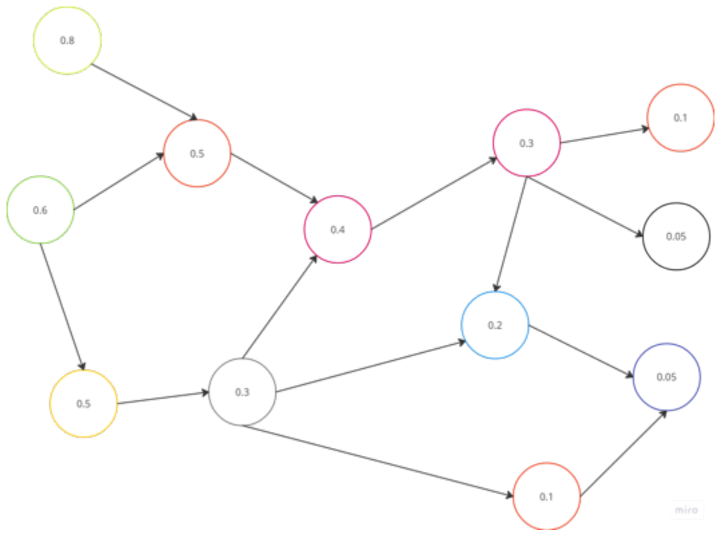


Fig. 1 Learner Graph: has the same structure as its corresponding expert graph but with labeled probabilities

3.2.2 Expert Graph Generation

The expert graph, including each of the questions, is automatically generated through an algorithm:

1. The expert graph is generated from a structure called a skill tree (see Figure 2). This skill tree is a graph of the various topics containing the connections between topics, dictating their pre-requisite skills. Its hierarchy of topics makes questions asked to the user conceptually organized. The actual expert graph (see Figure 3) itself is iteratively created from this tree. When generated, it consists of hundreds of nodes.

2. For each answer to a question in the expert graph, three other similar options are created by randomly altering the numbers in the answer choice, creating plausible but incorrect solutions to the problem. This aspect of the generation of the expert graph serves to fill in the gap left by the dataset used.

3.2.3 Learner Adaptivity

Adaptivity in our learning system comes from two key elements of the system itself:

1. The learner graph modifies the predicted probability of success of the learner based on factors such as skill strengths, weaknesses, recent motivation, and comfort levels. The probability considers the learner getting the question correct by random chance and is modified according to item response theory (IRT). A study on this

Fig. 2 Skill Tree: a graph with the various topics interconnected

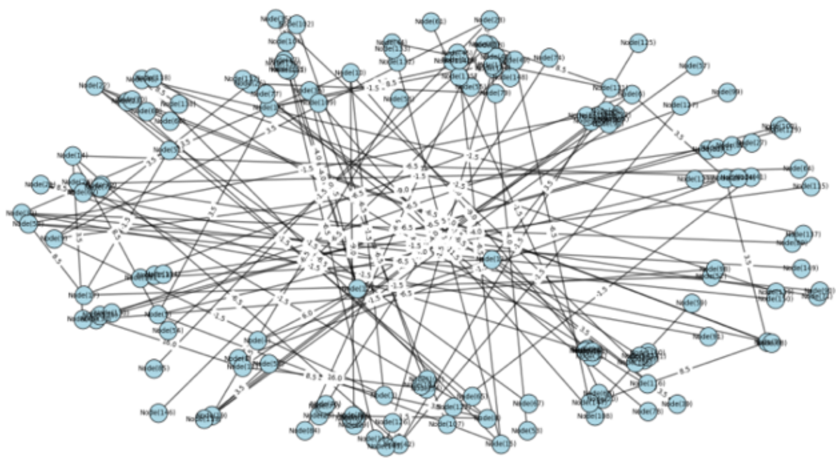


Fig. 3 An example of a randomly generated expert graph

theory uses psychological analysis to demonstrate that it is an effective method of connecting an individual's position on a continuum and the probability of response to a certain place on that continuum [3]. In our case, the continuum is proficiency with a certain skill. According to IRT, the probability graph follows a Sigmoid curve.

2. The tutor model algorithm, as the name suggests, acts like an in-person tutor would, adjusting the material shown to the student based on factors like performance and perceived mood. The tutor's end goal is to reach the learner's desired goals and master a certain topic. The tutor model is also generative in the sense that edge weights, a.k.a 'viability score' of the graph, are changed with every node visited to reflect the viability of traversing adjacent nodes. This update helps guide the algorithm.

To this end, the tutor model uses graph traversal algorithms similar to A-star. The A-star heuristic would be the current performance (i.e. weighted distance away from the goal skills) using the viability scores of the graph.

The end product of combining these modes of adaptivity is a path to achieving the learner's desired goals that thoroughly incorporate a combination of personalization and efficiency.

4 Results

So far, the web interface is complete, with users able to input both the skills they have a background in (see Figure 4) as well as their desired goal (see Figure 5). After that, the web interface asks the user a series of questions to eventually allow the learner to achieve their inputted goal. While answering questions, they are able to make their choice and update their motivation and comfort levels through sliders (see Figure 6).

Currently, the program is able to generate a fully connected expert graph of 200 questions in around 2 minutes, leaving a lot of room for improvement in the efficiency and runtime of the program. To provide a productive and worthwhile learning tool for students, more technical tests will be conducted to view the functionality of the learning system, which is believed to give a starting point for determining how effective it is for learners before releasing it to students worldwide.

The preliminary tests of running the application and answering the generated questions do point towards the fact that these are effective algorithms that accurately consider learner performance and mental state in reaching learning targets. However, at this point in the project, more work is required for more concrete results, and feedback from other subjects is required to develop a more accurate understanding of how effective the learning system is.

5 Conclusion

The project is still in the process of development and has not yet been tested by users, but the implementation of the learning system strives to make effective education more accessible to all types of students. Since the learning system has not been tested yet, it is unclear whether the learning experience is positive. It is also unknown how far users can go with the learning system. The testing will hopefully provide insight into how effective the project actually is and whether it will help learners receive customized education that benefits them and makes them stronger with the concepts they do not understand. The next steps are to test the application and algorithms for

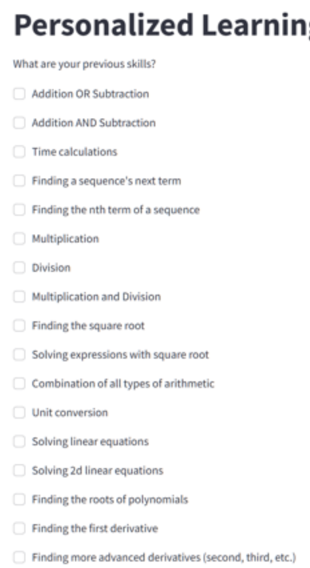


Fig. 4 Menu for learners to choose their background skills

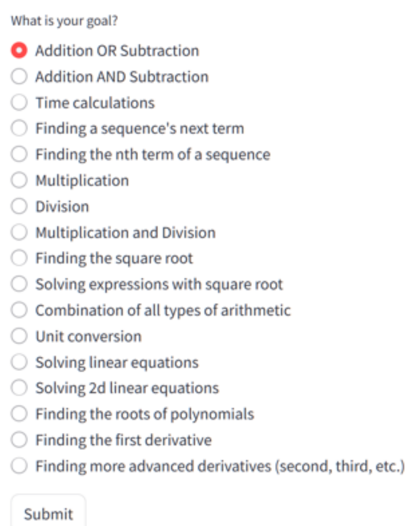


Fig. 5 Menu for learners to choose their goal skill

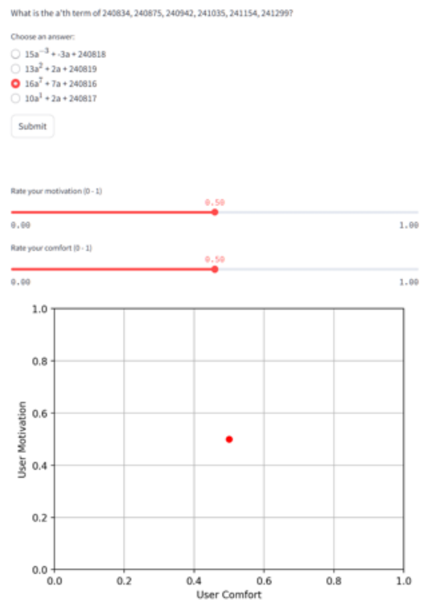


Fig. 6 Layout for each question asked

effectiveness and functionality via other subjects and continue to potentially enhance the interface along the way throughout the testing process.

Acknowledgements This work was supported by the altREU program at Portland State University. Thanks are extended to Professor Christof Teuscher for his continued support, as well as the following professors also from Portland State University: Bart Massey, Suresh Singh, and Wayne Wakeland.

References

1. Sabine Graf, Kinshuk, and Cindy Ives. A flexible mechanism for providing adaptivity based on learning styles in learning management systems. pages 30–34, 2010.
2. Min Liu, Emily McKelroy, Stephanie B. Corliss, and Jamison Carrigan. Investigating the effect of an adaptive learning intervention on students' learning. *Educational Technology Research and Development*, 65(6):1605–1625, 2017.
3. Steven P. Reise, Andrew T. Ainsworth, and Mark G. Haviland. Item response theory: Fundamentals, applications, and promise in psychological research. *Current Directions in Psychological Science*, 14(2):95–101, 2005.
4. Judy C.R. Tseng, Hui-Chun Chu, Gwo-Jen Hwang, and Chin-Chung Tsai. Development of an adaptive learning system with two sources of personalization information. *Computers & Education*, 51(2):776–786, 2008.
5. Tzu-Chi Yang, Gwo-Jen Hwang, and Stephen Jen-Hwa Yang. Development of an adaptive learning system with multiple perspectives based on students' learning styles and cognitive styles. *Journal of Educational Technology & Society*, 16(4):185–200, 2013.

Orphan Drug Trials: Forecasting Phase Three Clinical Trial Success

Shani Inbari

Abstract The process of bringing a drug from discovery to market is highly costly and can take up to several decades. Despite significant investment, the proportion of drugs that survive this process is relatively low compared to the quantity discarded. At the same time, the need for treatment remains as high as ever, particularly for orphan drugs – medications that target rare diseases affecting small patient populations. To enhance the efficiency of the drug delivery process, the application of machine learning algorithms for trial outcome prediction has emerged as a powerful tool to combat this disparity. In this project, a random forest model was used to predict historic phase III orphan drug trial outcomes based on data from clinicaltrials.gov, relying on variables related mainly to trial design, enrollment, and previous sponsor success. These findings inform researchers on the vitality of quality trial preparation and contribute to improving the efficiency of the drug development process.

1 Related Works and Background

Clinical trials are often the most expensive stage of the drug development process, costing about $2.6 billion [3] USD for a successful trial. Successful clinical trials can last 10-15 years or even longer before a drug or therapy enters the licensing stage – the regulatory approval process companies must undertake to obtain rights to manufacture and produce their product. Despite the significant investment of time and resources, the efficiency of this process remains low – the success rate of a new drug from discovery to registration is 7.9% [12].

Historically, this challenge was even more pronounced for orphan drugs, a class of pharmaceuticals developed to treat rare diseases affecting fewer than 200,000 people in the United States [11]. As only a small patient population is impacted by each rare disease, few pharmaceutical companies were willing to spend billions of

Shani Inbari
The University of North Carolina at Chapel Hill, Chapel Hill, NC, e-mail: sinbari@ad.unc.edu

dollars in drug development for little return profit. To address this issue and make rare disease treatment more accessible, Congress passed the US Orphan Drug Act in 1983. This act provides various incentives to the sponsoring company, including tax credits for qualified clinical trials, exemptions from user fees, and up to seven years of market exclusivity upon drug approval [6]. The benefits of this legislation are evident not only for the orphan disease population but also for the entire drug development process. Monocl Strategy and Communication reports that phase III drug approval success rates have increased over the last decade, potentially due to the incentives provided for orphan drug approvals [8].

Given the high stakes associated with drug development, it is critical to streamline this process to deliver medicine to patients more quickly and improve resource allocation for investors and shareholders. To address these needs, machine learning algorithms (MLAs) are becoming increasingly popular in FDA approval predictions and in the design of clinical trial protocols. MLAs, which are algorithms in the field of artificial intelligence (AI), learn the relationships and associations in datasets and leverage these relations for predictive modeling and classification [1]. Predictive models can analyze vast amounts of data to forecast trial outcomes, and by accurately predicting the success of ongoing trials, the drug development process has the potential for acceleration. This ensures patients receive therapies more efficiently while companies can reduce costs and mitigate financial risks. Several statistical analyses and trend identifications have been conducted to pinpoint the key factors influencing trial success, such as those by Kim et al. [9] and Fogel [5]. However, there is still potential for growth in applying MLAs to predict trial outcomes. The most comprehensive study to date using machine learning for drug approvals was conducted by Lo et al. [10], where they predicted drug approvals across many different disease groups using both drug and trial information.

This project aims to predict the trial outcomes of historic Phase III clinical trials that utilize approved orphan drugs. Using a random forest model, which aggregates the result of multiple decision trees to reach a single prediction, the concluded orphan drug trials will be classified as either completed or terminated and will be compared with their actual trial status to determine the accuracy, precision, recall, ROC AUC, and F1 scores of the model. These metrics will reflect the model's success in accurately predicting trial outcomes, providing valuable insights into its reliability and effectiveness in making assessments from trends in the dataset. Additionally, this project exclusively examines public-source information gathered from clinicaltrials.gov and seeks to analyze if accurate predictions can still be made despite the less detailed nature of public data. By leveraging advanced machine learning techniques, this project aims to provide insights into trial outcomes for historic orphan drug trials and ultimately contribute to improved efficiency in the drug development process.

2 Data Collection and Methodologies

The FDA can grant orphan drug designation to any biological product or drug used to prevent, diagnose, or treat a rare condition or disease. The sponsor (a person, company, or organization that oversees a trial and analyzes the results) must file an Orphan Drug Designation Request Form and provide a plausible hypothesis demonstrating that their drug or biological intervention offers clinical superiority over existing treatments for a specific indication (a medical condition) [7]. Due to these precise requirements, each trial's data used for this study was only collected if the sponsor, indication, and drug were correctly aligned to ensure that the orphan drug designation was valid for each specific trial. In this study, a "designated orphan drug" refers to a drug that has been granted FDA orphan designation and is not yet market-approved, and an "approved orphan drug" refers to a drug that has been granted FDA orphan drug designation and market approval.

2.1 Data Collection

Firstly, a list of all designated, approved, and withdrawn orphan drugs was gathered from the FDA Orphan Drug Designations and Approvals Database. This data provided information on the drug, condition, sponsor information, designation status, designation date, approved labeled indication (when applicable), and market exclusivity period (when applicable). Approved drugs were isolated for further processing. Drugs on the dataset were randomly selected for further search on clinicaltrials.gov. For each search, the drug, indication, and sponsor were all inputted into the search query, and their available trial information was extracted manually.

2.1.1 Data Filtration

Once trial information was found on approved orphan drugs, these trials were filtered to ensure consistency. All trials with the following details were removed: Trials with withdrawn/suspended/other trial status, any phase besides phase III, any trial that began conduction before January 1, 2000, any funder type that was not industry, or any trial with missing information. This accumulated to 78 finished trials that used approved orphan drugs. Of the 78 concluded trials, 50 have their final trial status as completed, while 28 have their final trial status as terminated.

Upon acquisition, the following trial information was available: NCT Number, study title, study URL, acronym, study status, summaries of the trial and outcome measures, condition, intervention, demographics, results, sponsor, collaborator, phase, enrollment, funder type, study design, documents, locations, and key dates. Information was isolated from this data to become the variables of interest based on previous findings from Lo et al. [10] and Kim et al [9] . These variables are referenced in Table 1. All information related to the sponsor's track record was gathered by searching the sponsor in clinicaltrials.gov and gathering information on their terminated and completed trials since 2000. The number of approved orphan drugs of the sponsor was found by searching the sponsor's name through the list of the orphan drug information, gathered from the FDA Orphan Drug Designations and Approvals Database.

Trial Variable	**Definition**
Final Trial Status	Final trial outcome of the study. In this project, trial status can only go to completed or terminated.
Enrollment	The number of patients listed as participants in the study.
Duration	The clinical trial length from the start date to the completion date.
Presence of a collaborator	The presence of an organization besides the sponsor providing support for a study, which can include funding, design, implementation, data analysis, and reporting.
Number of total concluded trials since January 1st, 2000	The number of completed and terminated trials the sponsor has undertaken since January 1st, 2000, of any phase or trial type (according to clinicaltrials.gov).
Number of total successful trials since January 1st, 2000	The number of completed trials the sponsor has undertaken since January 1st, 2000, of any phase or trial type (according to clinicaltrials.gov).
Success ratio of sponsor	The number of completed trials by the sponsor since January 1st, 2000 divided by the number of completed and terminated trials by the sponsor since January 1st, 2000.
Number of approved orphan drugs by the sponsor	The number of orphan drugs the sponsor has reached approved orphan drug status for according to the FDA Orphan Drug Designations and Approvals Database.
Study Design: Allocation	The method used to assign participants to different arms of the clinical trial.
Study Design: Masking	The design strategy selected to prevent researchers and patients from knowing which patients have been assigned to which intervention.
Study Design: Interventional Model	The strategy of assigning participants to interventions (drugs or treatments).
Study Design: Primary Purpose	The main reason the study is being performed.

Table 1 A depiction of the trial variables used for the random forest model and their descriptions.

2.2 Model Assembly

Using the data collected, a random forest model was generated from the scikit-learn and imbalance-learn libraries. The model's goal was to predict the class label of finished clinical trial data, which in this case is the trial status, and assign it to either completed or terminated based on the variables provided. An assessment of the model's performance quality was conducted by determining the metrics referenced in Table 2.

Metric	**Definition**	**Equation**
Accuracy	The ratio of correctly predicted samples (both true positives and true negatives) to the total number of predictions made by the model.	$\frac{\text{Correct Predictions Made}}{\text{Total Predictions Made}}$
Precision	The ratio of true positive predictions to the total number of positive predictions made by the model (true and false positives).	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
Recall	The ratio of true positive predictions to the total number of actual positive samples in the dataset (true positives and false negatives). Also known as the sensitivity of the model.	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
F1	The weighted harmonic mean between precision and recall.	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
ROC AUC	The Area Under the Receiver Operating Characteristic Curve (ROC AUC) is a measurement that plots the true positive rate against the false positive rate. This is an aggregated performance measure that reflects all classification thresholds.	

Table 2 A depiction of the metrics used to measure model performance, along with their descriptions and equations.

2.2.1 Encoding Categorical Data

One hot encoding was used to assign categorical data numeric values. In one hot encoding, each unique category is assigned a column, and each option within the category is mapped to its respective binary value (0 or 1). The 'get_dummies' function from the pandas library was utilized for this task. The following binary labels were assigned to their corresponding categories:

- Presence of a collaborator – 1
- Lack of a collaborator – 0
- Terminated trial status – 0
- Completed trial status – 1

Additionally, the trial design category contained four elements – allocation, interventional model, masking, and primary purpose — which were separated into individual columns. Each unique category within these elements was assigned a binary value, allowing for a clear representation of each category in the dataset.

2.2.2 Random Forest Model

A random forest model is an ensemble learning technique that improves predictive performance by combining the output of multiple models, in contrast to performing on one model's results alone. In a random forest, numerous decision trees are used, and their predictions are aggregated to reach a final outcome for classification or regression problems. In this project, the model classifies the trial as completed or terminated. A random forest classifier relies on the bagging (bootstrap aggregation) principle, where a subset of the data is sampled with replacement to create multiple datasets, and each decision tree is trained on one of the subsets. A final decision is reached based on combining the results of the trees through majority voting, and the data is aggregated into a single, consolidated output [4].

2.2.3 k-Fold Cross Validation

Cross-validation is a resampling technique that uses different proportions of the dataset to train and test a model. Unlike a simple train-test split, where the model is trained on a larger portion of the data (the train set) and evaluated on a smaller, separate portion (the test set), cross-validation involves dividing the dataset into multiple subsets, known as folds. Thus, the model is trained and tested numerous times, allowing for a more robust assessment of its performance and results in a less biased estimate of model skill compared to a simple train-test split.

This procedure begins with shuffling the dataset. Next, the number of folds (k value) is selected; in this case, five. One of the folds is used as the test set, and the remainder is used as the train set. Each fold takes turns being the test set, while the remaining folds are used as the train set. The model is trained on the train set and evaluated on the test set for each iteration. This procedure is repeated as specified (n times). The results from all iterations are aggregated to produce a final performance estimate of the model, theoretically improving metrics such as accuracy, recall, precision, F1, and ROC AUC scores. This technique provides a more reliable assessment of how well the model is likely to perform on unseen data [2].

2.2.4 Hyperparameter Tuning and Feature Importance

Four different techniques were utilized to account for the class imbalance between the completed and terminated trials. Firstly, the RandomForestClassifier method from scikit-learn was used with no class weight adjustment. Then, the RandomForestClassifier method was used with the 'balanced' weight, which assigns inverse weighting from the training to the testing dataset, and the 'balanced_subsample' weight, which bases the class distribution off each bootstrap sample rather than the entire training set. Finally, the BalancedRandomForestClassifier from the imbalance-learn library was used, which uses random under-sampling (data resampling on the bootstrap sample to change the class distribution explicitly) to balance the classes.

The RandomForestClassifier model with the 'balanced_subsample' weight was selected for additional hyperparameter tuning. The 'GridSearchCV' technique from the scikit-learn library was implemented to locate hyperparameters that yield optimal model performance on the ROC AUC metric. Additionally, feature importance examination was conducted by using the fitted attribute 'feature_importances_' to determine what weight each variable had on the model's decision to classify a trial as completed or terminated.

3 Results

The comparison of each class weight's influence on the metrics of the random forest model is referenced in Figure 1. From this table, it was determined that the results were very similar among all models, and no significant advantage between models could be concluded. The average score of each metric considering all models is as follows:

- 84.3% ROC AUC
- 84.1% precision
- 81.5% recall
- 75.6% accuracy
- 81.4% F1

As each model reached an average score of around 80%, regardless of the metric, it can be inferred that the models performed consistently well in correctly identifying class instances and balancing both precision and recall. For simplicity, the

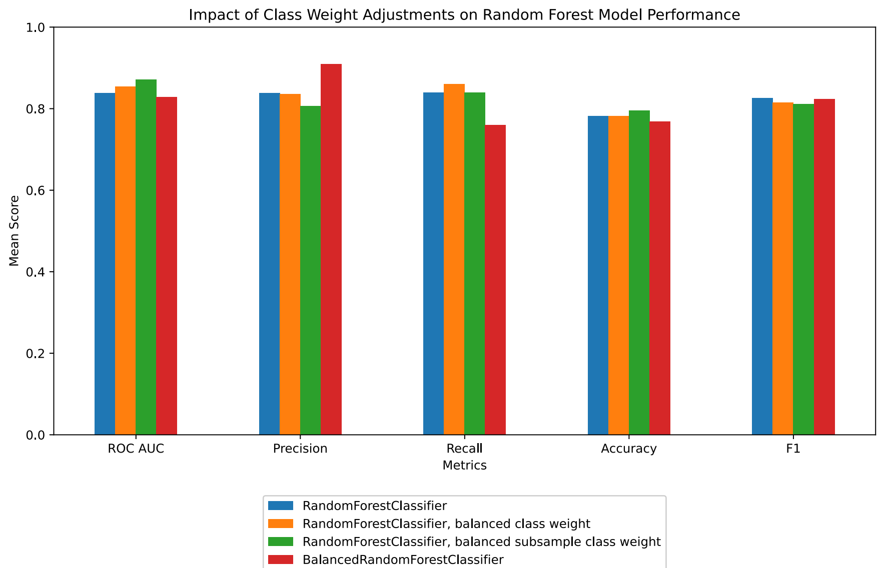


Fig. 1 Comparison of different class weight adjustments and sampling methods on the random forest model performance.

RandomForestClassifier with 'balanced_subsample' class weight was selected for further hyperparameter tuning.

Using GridSearchCV with five folds and only one repeat, it was concluded that the most optimal hyperparameters were as follows: maximum tree depth of two, 'log2' for maximum features to consider, one minimum sample per leaf, two minimum samples per split, and 41 estimators (trees). A comparison of the metrics before and after hyperparameter tuning for the RandomForestClassifier with the 'balanced_subsample' class weight is shown in Figure 2. Additionally, an analysis of the trial features that most influenced the model's classification determination was conducted using the 'features_importances_' method in scikit-learn. This analysis revealed to what extent each trial attribute influenced the completion or termination classification, as depicted in Figure 3.

4 Discussion

This analysis of clinical trial outcome prediction can conclude two significant findings. Firstly, regardless of the metric or class weight adjustment, each model scored about 80%, revealing the success of the model in sensing the underlying patterns in the data and determining trial completion based on our variables of interest. Additionally, the average AUC score was 0.85, which is consistent with the average AUC

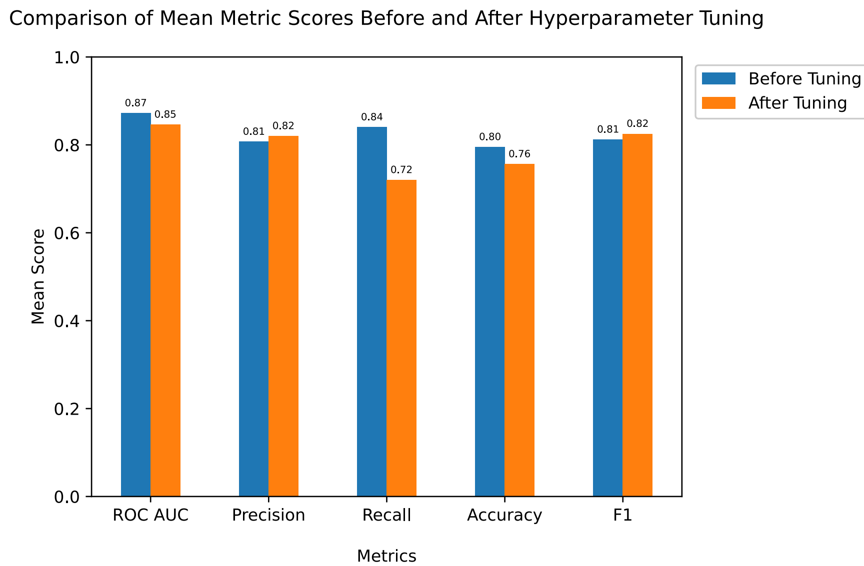


Fig. 2 Comparison of model performance with the balanced subsample class weight adjustment, before and after hyperparameter tuning.

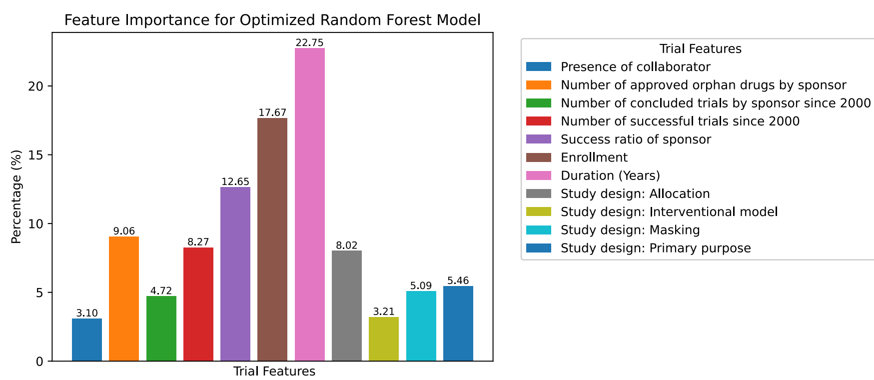


Fig. 3 Analysis of the trial attributes that impact the model's classification of trial status. This graph illustrates the results from the model optimized with the best hyperparameters, as identified by GridSearchCV.

score found by Lo et al. [10] for rare diseases, which was 0.819. This furthers the notion that predictive analysis on phase III orphan drug trials can be conducted on trial information alone. This reflects a critical point about clinical trial structure and preparation. This study focused exclusively on trial attributes — without considering the drug design, medical impact, or primary and secondary outcomes — and found that these attributes can be reflective of a study's success enough to make predictions on the final trial status. These findings emphasize the importance of meticulous trial structure, preparation, and design to optimize the chance of a trial's success. Regulatory and clinical professionals, pharmaceutical companies, and marketing groups involved in trial preparation should prioritize meticulous planning to increase the likelihood of a favorable trial outcome. Additionally, by assessing the quality of a trial design, stakeholders and industry leaders can better strategize and allocate resources to ensure higher chances of return on investment.

Of all the variables used to make these outcome predictions, duration stood out as the attribute the model most closely relied on for predictive assessment, followed by enrollment and the sponsor's success ratio. This finding stresses the importance of maintaining efficient trial speed and adequate enrollment numbers. Additionally, shorter trials often correlate with lower investment risk, making them an appealing option for stakeholders who are cautious about investing in new drugs. By minimizing trial duration, sponsors accelerate their time-to-market and establish market exclusivity before other competitors. A trend of efficient trials conducted by a sponsor sends a positive signal to investors looking at reducing risk profiles. It also can attract support from stakeholders eager to see lower financial exposure and quicker return on investment.

Additionally, adequate enrollment can be particularly challenging for orphan drug trials where the patient population is limited. Therefore, larger participant groups are a positive indicator that the study will gain substantial quantities of data, which is crucial for achieving statistically significant results and expanding the implications of the findings. This also enables a study to reach its endpoints more efficiently, thus also contributing to a shorter trial duration and quicker time-to-market. For investors and stakeholders, high trial enrollment reflects a well-designed study, an attractive feature from both a financial and scientific perspective.

4.0.1 Limitations and Future Directions

There are several ways in which this study could be improved in future works. Firstly, the dataset was limited due to challenges in the data collection process; future studies should include significantly more data to reach accurate prediction levels. Given the limited dataset in this project, all conclusions would need to be consistent with a larger dataset to ensure reliability. These preliminary findings should be interpreted cautiously until further research on a larger scale can validate these findings. Additionally, future studies should incorporate statistical testing to determine the differences between models and explore other machine learning techniques to improve predictive analysis. Future studies should also analyze active orphan drug trials to offer real-time insights to sponsors, helping them determine whether a trial is likely to reach completion or face termination—an objective this project was unable to accomplish due to time constraints. This project focused on orphan drugs because of the unique financial incentives provided to the sponsor, unlike other types of medications; however, due to time constraints, a thorough analysis of the economic impact was not conducted. Further analyses should integrate external financial variables – such as the stock price of sponsoring companies, changes in the GDP, market competition, etc – with clinical data. This would offer a more comprehensive understanding of orphan drug trials and provide deeper insights into the factors influencing their success.

5 Conclusion

The results of this study demonstrate the capability of using a random forest model to predict the trial outcomes of historic phase III clinical trials using orphan drugs, based on limited data pertaining to trial attributes and design. The model's ability to predict the result of the trial and achieve approximately an 80% accuracy, precision, recall, F1, and ROC AUC score reflects the model's ability to identify trends in the data and assess the success of a trial independent of medical analysis of the drug or treatment. This highlights the critical role that quality trial preparation and execution play in the success of an orphan drug trial, especially by sticking to a tight schedule to limit duration, thus saving money and expediting the time to market. In future studies, much larger datasets must be used to analyze if this could have any real-world applications, but if successful, the outcomes of this project could assist in creating more informed decision-making in the pharmaceutical industry, leading to enhanced planning, optimized resource allocation, and increased likelihood of trial completion.

Acknowledgements This work was supported by the altREU program at Portland State University. Special thanks goes to Dr. Christof Teuscher for his support during this project, in addition to the guidance from Dr. Bart Massey, Dr. John Lipor, and Yogeshewar Landge from Portland State University. This work acknowledges the use of AI in the construction of this project.

References

1. BA Badwan, G Liaropoulos, E Kyrodimos, D Skaltsas, A Tsirigos, and VG Gorgoulis. Machine learning approaches to predict drug efficacy and toxicity in oncology. *Cell reports methods*, 2023.
2. J Brownlee. A gentle introduction to k-fold cross-validation. https://machinelearningmastery.com/k-fold-cross-validation/, 2023.
3. JA DiMasi, JC Hermann, K Twyman, RK Kondru, S Stergiopoulos, KA Getz, and W Rackoff. A tool for predicting regulatory approval after phase ii testing of new oncology compounds. *Clinical pharmacology and therapeutics*, 98(5):506–513, 2015.
4. Sruthi ER. Understanding random forest algorithm with examples.
5. David Fogel. Factors associated with clinical trials that fail and opportunities for improving the likelihood of success: A review. *Contemporary Clinical Trials Communications*, 2018.
6. U.S. Food and Drug Administration. Designating an orphan drug or biologic. https://www.fda.gov/industry/medical-products-rare-diseases-and-conditions/designating-orphan-product-drugs-and-biological-products.

7. U.S. Food and Drug Administration. Frequently asked questions. https://www.fda.gov/industry/designating-orphan-product-drugs-and-biological-products/frequently-asked-questions-faq-about-designating-orphan-product, 2023.
8. M Jewett. Biostock's article series on drug development: The success of orphan drug designation. https://www.biostock.se/2019/10/biostocks-article-series-on-drug-development-the-success-of-orphan-drug-designation/, 2019.
9. E Kim, J Yang, S Park, and K Shin. Factors affecting success of new drug clinical trials. *Therapeutic innovation regulatory science*, 2023.
10. AW Lo, KW Siah, and CH Wong. Machine learning with statistical imputation for predicting drug approvals. *Harvard Data Science Review*, 2019.
11. National Conference of State Legislatures. Brief rare and orphan diseases. https://www.ncsl.org/health/rare-and-orphan-diseases, 2023.
12. Biotechnology Innovation Organization. Clinical development success rates and contributing factors 2011- 2020. https://go.bio.org/rs/490-EHZ-999/images/ClinicalDevelopmentSuccessRates2011$_2$020.pdf, 2020.

Applying Deep Learning Techniques to Diagnose Pneumonia Using Lung X-ray Images

All members equally contributed to this study
Yash Deswal, Riyaan Jain, and Kashan Ali

1 Abstract

Pneumonia is a major cause of death globally and presents significant challenges in medical diagnostics [1]. Advancements in artificial intelligence, specifically computer vision using deep learning, can offer assistance to medical professionals in diagnosing pneumonia efficiently and with increased accuracy. Analysis of the lungs using chest X-rays is essential to diagnose respiratory conditions like pneumonia. In this study, we introduce an open-source integration of a convolutional neural network (CNN) trained on an extensive, publicly available dataset of segmented lung X-ray images containing 35,000 images. This framework provides a diagnosis, a certainty metric, and a heatmap that highlights regions of interest that could aid health professionals in their diagnosis. This tool shows promise to be a reliable means to enhance diagnostic accuracy and efficiency in the healthcare industry.

Yash Deswal,
University of Illinois at Urbana-Champaign, e-mail: ydeswal2@illinois.edu

Riyaan Jain,
University of Illinois at Urbana-Champaign, e-mail: riyaanj2@illinois.edu

Kashan Ali,
University of Illinois at Urbana-Champaign, e-mail: kali26@illinois.edu

2 Introduction

By integrating machine learning in medical diagnostics, we have the potential to create a future where interpretations on radiograph images are accurate and reliable. Analyzing chest X-ray images with artificial intelligence has the power to improve treatment outcomes by providing medical professionals with complementary tools in their analysis of these images. By contributing to this field of research, we hope to propel medical diagnostics towards reliability and autonomy.

A pivotal role in diagnosis is played by radiology, which is a non-invasive procedure that provides a visual insight into the patient's internal state [2]. Among the most common radiology tests are chest X-rays, providing important visual information about the functionality of vital organs such as the heart and lungs. Chest X-rays are essential to the diagnosis of respiratory diseases such as lung cancer, pneumonia and tuberculosis [3][4].

Radiology reports are in high demand, however, there is a relative unavailability of radiologist specialists. The lack of specialists can lead to the risk of inaccurate diagnosis due to evaluations being given by non-specialists [5].

Deep learning, a subset of machine learning, has significant potential to resolve many of these issues, particularly through the use of convolutional neural networks. The application of deep learning in medical imaging is a widely researched area, with a strong emphasis on the uses of computer vision. Significant strides have been made, such as the introduction of U-net—a neural network that greatly improves biomedical imaging segmentation—and generative adversarial networks (GANs) for anomaly detection [10][11].

Studies like those by Racic et al. and Pant et al. tend to either excel in accuracy but lack a large dataset, or lack transparency in datasets and code. While such studies are impressive and critical to our work, they are often irreproducible or lack generalizability due to small datasets and overfitting. Our goal is to achieve the best possible accuracy on a large, free, and public dataset using a pre-trained CNN, while also providing insight into the model's decision making.

3 Related Work and Background

Research on creating sophisticated algorithms for medical diagnosis is an up-and-coming field that can be highly beneficial to health professionals worldwide. Providing large datasets of chest X-rays to sophisticated machine learning models presents the potential for AI to understand the subtleties of radiographic images, efficiently diagnosing pneumonia. This research field is growing with the ever-expanding possibilities of deep learning models. Large amounts of critical care information, including lung X-ray images, were provided via the MIMIC-IV database shared on PhysioNet. This information is imperative to creating reliable deep learning diagnostic models. The MIMIC-CXR dataset, also found on PhysioNet, provides a large collection of clinical reports that correspond to chest X-rays; this is very necessary for both training and testing deep models. The MIMIC-CXR-JPG dataset contains 243,324 frontal chest X-ray images containing images of lung X-rays segmented by the SA-UNet 13 architecture.

Our convolutional neural network (CNN) architecture was trained on labeled segmented lung X-ray images with different abnormalities from the CXLSeg dataset, which contains segmented chest X-ray radiographs based on the MIMIC-CXR dataset. The results by Yifan Zhang [5], published by Northwestern University, present the performance of CNNs in recognizing pneumonia specifically in the MIMIC-CXR-JPG dataset and thus set a benchmark for our model.

Recent deep learning developments, such as those by Zhang et al. (2020), where it is shown that CNNs can be very effective at differentiating between pneumonia and healthy lung X-rays, also help inform our methodology.

We have used Gradient-weighted Class Activation Mapping, a technique described by Selvaraju et al. (2017), to provide interpretability in the convolutional neural network's decision-making. Grad-CAM increases the transparency of our model's inner workings by generating a heatmap that points out areas in images that contribute most to the prediction made by the CNN. Caruana et al. (2015) consider model interpretability critical in clinical settings and promote techniques that provide insight into the reasoning behind AI predictions.

Comparative research into deep learning architectures by Rajpurkar et al. (2018) has been very instrumental in helping us fine-tune our CNN architecture for pneumonia diagnosis. Finally, studies like those by Lakhani and Sundaram (2017) on the challenges and progress in machine learning-aided diagnosis of pneumonia underpin the necessity for better diagnostic tools, highlighting the applicability and potential benefits of an AI-based approach. Taken together, these references help in developing and refining our diagnostic tool and ensure its utility to the medical practitioner.

4 Data and Methodologies

4.1 Dataset and Preprocessing

For this research, we used the Chest X-ray with Lung Segmentation (CXLSeg) dataset (see Fig 1 for a sample batch). The CXLSeg dataset is a large, freely accessible collection of segmented chest radiograph images created to bolster medical image segmentation research. It was derived from the MIMIC-CXR dataset and includes 243,324 frontal-view chest X-ray images. Each of these X-ray images was preprocessed with AI-generated segmentation masks that highlight the lung regions. These masks are essential for fine-tuning AI-based diagnostic systems by enabling models to focus on the most relevant parts of an image, thereby improving the accuracy of medical image analysis. The CXLSeg dataset is applicable to a broad spectrum of computer vision tasks, including medical image classification, semantic segmentation, and the creation of medical reports. Specifically, models trained with CXLSeg have proven to be quite effective. For example, the SA-UNet model achieved an IoU score of 91.97% and a Dice similarity coefficient of 96.80% for lung segmentation [12].

As the dataset was large, we decided to split it into three sections, each containing 81,110 chest X-ray images. This allowed us to shorten the amount of time needed for the preprocessing. The biggest issue with our current dataset was the fact that lung segmentation was carried out by an AI, which resulted in about 10% of the images being of poor quality with incorrect segmentations. This forced us to manually remove the X-ray images with low quality (see Fig 1 for Good and Poor Quality Segmentation) and incorrect segmentation, leaving us with 220,203 images. Furthermore, we also needed to rotate the images as they all came oriented differently.

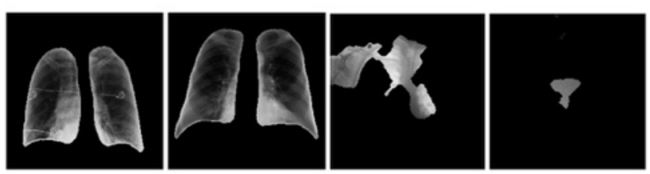


Fig. 1 Chest X-ray Images from the Dataset. This shows the masking of the lungs which was given in the dataset. The two images on the left show good segmentation, while the two images on the right show poor segmentations.

4.2 Labeling

Additionally, the original CXLSeg dataset did not differentiate between pneumonia and non-pneumonia patients. It contained X-ray images from over 10 different respiratory diseases such as COVID-19 and Pneumothorax. The dataset included X-ray images that were not labeled, which made it so we had to label the images accordingly. The dataset contained a metadata file that included values of [-1, 0, 1] under each corresponding column for each disease. Each of these integer values had its own meaning when it came to the diagnosis. A -1 indicated that the findings were uncertain or not readable, a 0 indicated there was no instance of the disease in the lung X-ray, and a 1 indicated there was an instance of the disease in the lung X-rays.

From this, we were able to create a Python program that looked for only instances of a 1.0 under the "Pneumonia" and "No Finding" columns and placed those X-rays in their respective folders, either pneumonia or normal. As our framework only dealt with pneumonia and non-pneumonia patients it was unnecessary for us to look at other diseases or X-rays related to other diseases. With this, we not only condensed the full dataset down to about 165,000 X-ray images but we made sure that our dataset was properly cleaned and only included relevant information.

We needed to split our dataset into testing, validation, and training folders. This was so that rather than the AI using the full dataset and getting overlaps between X-ray images, we could ensure that 80% of our dataset was being used for training and was unique compared to the 10% used for validation and 10% for testing. This percentage was recommended from the studies that used MIMIC. This was simply done using Python code that went through our full dataset and randomly split the data into its corresponding folders.

4.3 Model development

Our research for finding the most effective and accurate AI model for pneumonia classifications with the CXLSeg dataset was not as fruitful as we hoped. There had been research done using so many models with similar accuracy rates that we felt deciding on one particular model would limit our ability to potentially improve results. We aimed to try most of the models and test them – models such as ResNet50, ResNet34, Vgg16, and Nicholas Muchinguri's fine-tuned model for pneumonia [15] – without missing anything that might have seemed to hold some potential. In so doing, we were trying to find a model that provided the best trade-off between model accuracy, computational efficiency, and good generalizability across different datasets. This rigorous approach further enabled us to gain a finer understanding of the advantages and disadvantages of each type, which will help in coming up with some solid, well-informed solutions for pneumonia classification.

We chose to adopt a transfer learning approach, where these models were trained on our CXLSeg dataset after being pre-trained on large datasets such as ImageNet. Transfer learning speeds up the training phase and enhances performance by letting the models exploit features already learned, possibly useful for the task at hand.

First, we decided to implement our VGG16 model using TensorFlow Keras on Khan's Kaggle Dataset, which was much smaller in size. We used this model due to its comprehensive set of tools and high-level interface, that makes training and prototyping rather easy. In the process of development, though, we did face convergence problems of the models and the difficulty in tuning hyperparameters with a larger dataset, which turned out to bring very underwhelming performance with long training times. While with a smaller dataset, we were able to get much higher accuracy this was not the case when we ran the framework with the much larger CXLSeg dataset.

We worked on a reduced dataset of 35,000 images because of our limited computing resources. This was done on Kaggle, as it is a free cloud server to run machine learning models. In this way, we could run several iterations which wouldn't put too much stress on computing. Further, for testing the hyperparameters like learning rates and early stopping conditions, we also created a small subset dataset with 720 images for each folder, leading to a total of 2,160 X-rays. Early stopping was very useful in avoiding overfitting because it allowed us to terminate training at exactly the right time–that is when the model's performance on the validation set began to stabilize.

We decided to use ResNet34 for our smaller dataset, knowing that the design of the model is lighter and that it's easier to test with it, along with its smaller number of layers. Knowing that we had a larger dataset, we wanted to go with ResNet50, hoping that a deeper design would capture more complex features and result in better performance. Using ResNet34 with this smaller dataset, the model failed to generalize as much. With ResNet50, however, while having increased performance, also came issues associated with increased training time and thus computational load, which made hyperparameter tuning very hard and stable convergence even harder. These challenges underscored the importance of careful balancing between model complexity, dataset size, and computational resources, and drove us back to revisit the approach with further optimization strategies.

Knowing these weaknesses, we decided to look at other frameworks and changed to Python with PyTorch for Fast.ai. Using fastai gave us more flexibility and efficiency, enabling us to experiment on different models and methodologies. We used the fastai on Python module with PyTorch to accelerate the process since it had a much more user-friendly implementation process, particularly in regard to charting training progress and learning rates. Also, given our computing limitations, fastai was an efficient way to experiment with different models and hyperparameters.

Explainability was an important consideration in our effort to develop a reliable diagnostic tool for medical practitioners. We did this by incorporating Grad-CAM, or gradient-weighted class activation mapping, into the framework. Grad-CAM creates a heatmap that indicates which areas of the picture most impacted the model's conclusion. Grad-CAM works by using the gradients of the output class with respect to feature maps of a convolutional layer. It can also be used in identifying overfit models or otherwise not generalizing, focusing on the wrong parts of an image. This improves the utility of the tool to a medical professional who must understand the logic behind the AI outputs in addition to adding another layer of validation to the predictions of the AI.

Knowing the ”valley” and ” minimum point on steepest drop (Slice(min, steep))” values on the learning rate curve helps determine the learning rate. Minimum point on steepest drop is the steepest decrease in loss; hence, it's the most efficient learning rate. The valley marks the point where the loss is reduced without the danger of overfitting. We are able to optimize our model for improved performance by understanding these values.

5 Results

A total of 35,000 chest X-ray images were used to train, test, and validate the ResNet50 model. However, due to the vastness of our dataset and the lack of computation power, we used a smaller dataset containing 2,160 images (randomly selected from the full dataset) and split those into 720 images for each of the train, test and validation sets. This is in order to help us to find the optimal conditions for high accuracy on the model by fine-tuning and optimizing hyperparameters accordingly. We used ResNet34 for the smaller dataset.

Though attempts were made to incorporate Nicholas Muchinguri's fine-tuned model specifically for pneumonia detection, it proved to be too resource intensive, taking considerably longer to run than ResNet. VGG16 was also explored on our discontinued implementation that used TensorFlow, however, accuracy was inferior on the CXLSeg dataset, though was slightly better on Khan's dataset found on Kaggle, boasting 97% accuracy on compared to 96% found on ResNet50. The following section will discuss the results of running ResNet34 on the smaller model.

To begin with, we ran the model on the smaller dataset to find the optimal learning rate without any fine-tuning. We obtained the plot in Figure 2 for learning rate against loss and trained the model using both the learning rates of the valley value and the learning rate where the validation loss decreases most sharply and is at its minimum. After training the model for 4 epochs, we compared the accuracy and loss values obtained from training the model using the different learning rates, displayed in Table 1 and Table 2.

When initially training the smaller dataset, when the learning rate is set to a narrow region around the minimum loss on the steepest drop (Table 3), the final accuracy is 4.31% higher than when the learning rate is set to the valley value, however, the validation and training losses are also higher. The longest time to run an epoch using the learning rate of the minimum loss on the steepest drop was 9.05 minutes, as compared to 3.52 minutes for the valley learning rate. Without fine-tuning, we achieved an accuracy on the test dataset of 52.92% when training the model with the smaller dataset. When testing the model with an image from the validation set of a pneumonia patient, it correctly predicted the presence of pneumonia with 71.63% certainty.

Moving forward with the valley learning rate for all further runs, we trained the model on the smaller dataset and added fine-tuning. The initial run of fine-tuning spanned 20 epochs, but with the addition of early stopping with a patience of 3 epochs, the fine-tuning stopped after the 7th epoch, indicating no improvement since the 4th epoch in validation loss, presented in Table 3.

After optimizing hyperparameters by unfreezing the model and finding a new learning rate, the valley learning rate value decreased from slightly less than 0.001

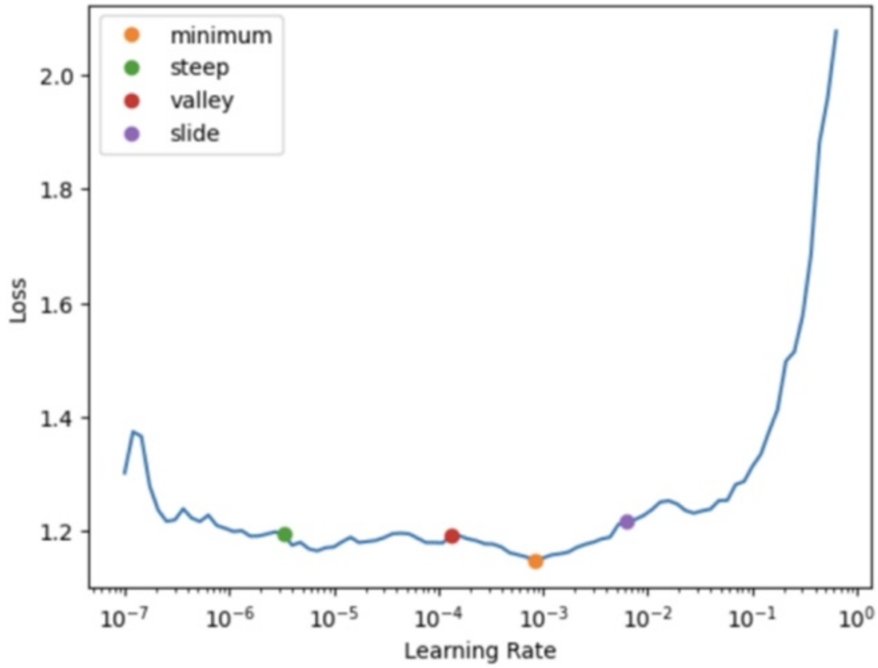


Fig. 2 Learning Rate vs Loss plot. This plot highlights key learning rate values with coloured dots, with a corresponding key on the top left.

epoch	train_loss	valid_loss	Accuracy	Time
0	1.101559	0.883700	0.525000	03:52
1	1.154545	0.823594	0.526389	03:45
2	1.134318	0.845187	0.511111	03:49
3	1.108461	0.881691	0.523611	03:50

Table 1: Loss and accuracy results from training the model on the smaller dataset using the valley learning rate value.

epoch	train_loss	valid_loss	accuracy	time
0	1.099027	0.860371	0.562500	08:27
1	1.152470	0.911116	0.523611	09:05
2	1.119738	0.968770	0.550000	08:52
3	1.126968	0.850834	0.566667	08:33

Table 2: Loss and accuracy results from training the model on the smaller dataset using the narrow region around the minimum loss on the steepest drop as the learning rate.

epoch	train_loss	valid_loss	accuracy	precision_score	recall_score	f1_score	time
0	1.069172	0.871970	0.573611	0.575931	0.558333	0.566996	02:37
1	1.043682	0.871399	0.587500	0.581818	0.622222	0.601342	03:13
2	1.058284	0.891779	0.595833	0.583133	0.672222	0.624516	04:16
3	0.991160	0.844746	0.595833	0.591512	0.619444	0.605156	04:16
4	0.988348	0.894822	0.544444	0.541885	0.575000	0.557951	03:23
5	0.967968	0.904318	0.590278	0.577197	0.675000	0.622279	04:16
6	0.943897	0.923989	0.584722	0.570115	0.688889	0.623899	04:21

Table 3: Loss, accuracy, precision, recall and F1 scores presented per epoch when fine-tuning the model on the smaller dataset using the valley learning rate.

to between 0.0001 and 0.00001. The model is then run to fine-tune for 10 epochs, but stopped at epoch 5 due to early stopping, indicating no improvement since epoch 2, evident in Table 4.

epoch	train_loss	valid_loss	accuracy	precision_score	recall_score	f1_score	time
0	0.847084	0.913624	0.570833	0.559441	0.666667	0.608365	03:50
1	0.834768	0.890128	0.572222	0.564039	0.636111	0.597911	04:05
2	0.832446	0.866220	0.591667	0.600610	0.547222	0.572674	03:31
3	0.866056	0.910678	0.583333	0.571090	0.669444	0.616368	03:32
4	0.863313	0.884044	0.590278	0.583120	0.633333	0.607190	03:42
5	0.846623	0.894150	0.577778	0.565728	0.669444	0.613232	03:32

Table 4: Loss, accuracy, precision, recall and F1 scores presented per epoch when fine-tuning over 10 epochs after optimizing hyperparameters for the smaller dataset.

Figure 3 shows that there is a general negative trend in loss. After fine-tuning and training with the small dataset, the total accuracy on the test portion of the dataset is 59.03%, a 6.11% higher accuracy than training without fine-tuning. The same pneumonia X-ray image was used to test the model as when no fine-tuning was conducted, and the model correctly predicted pneumonia with certainty of 83.24%. The Grad-CAM was also run and produced the heatmap in Figure 4, highlighting the regions of interest in the image that the CNN used most to determine its decision. The following section discusses the full, 35,000 image dataset, which was split into 80:10:10 (%) for train, test, and validation respectively, as suggested by the Chest X-ray with Lung Segmentation dataset. When running our code on the full dataset, we discovered that each epoch took more than 2 hours to complete (Table 5). The initial training run, since it was 4 epochs, took around 9 hours to run. After running the cell to find the learning rates, shown in Figure 5, and conduct initial training, our session timed out due to Kaggle's 12 hour session limit restriction. Due to the

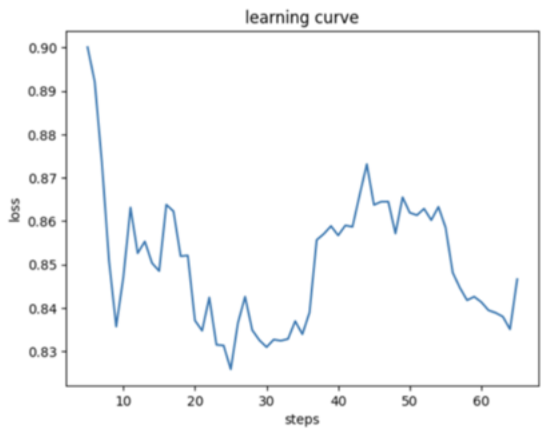


Figure 3: Learning curve plot with steps on the x-axis and loss on the y-axis.

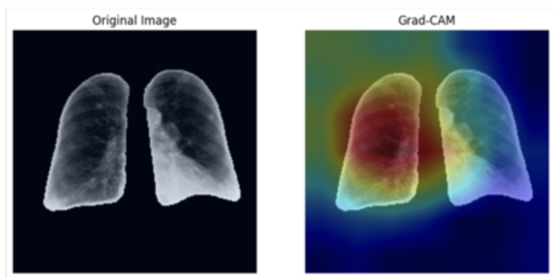


Figure 4: Heatmap produced by Grad-CAM, highlighting regions of interest to the CNN when trained on the smaller dataset.

Kaggle timeout restrictions, we had to reduce the number of epochs for initial training down to 2 (see Table 6), and removed all 20 epochs of initial fine-tuning. The hyperparameter optimization cell was run accidentally and could not be canceled without disrupting the session, and we were only able to fine-tune for 1 epoch after that. The learning curve displayed a stronger negative correlation on the full dataset than the smaller dataset, as seen in Figure 6. Figure 7 shows the confusion matrix of the run on the full dataset. A 56.96% test accuracy was reported. No test image was able to process due to a human error in file path, and once the error was spotted the session had already timed out.

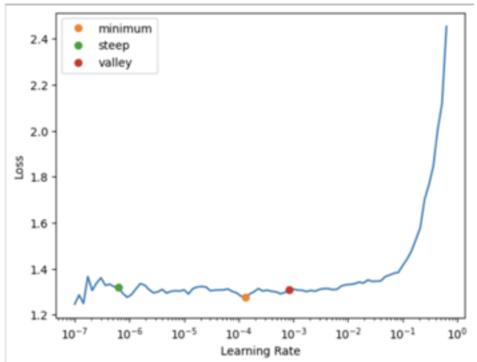


Figure 5: Learning rate vs loss for full dataset.

epoch	train_loss	valid_loss	accuracy	precision_score	recall_score	f1_score	time
0	0.813468	0.691017	0.563733	0.558550	0.608000	0.582227	2:05:08
1	0.688426	0.679791	0.549067	0.593117	0.312533	0.409361	2:16:55
2	0.677269	0.667919	0.585333	0.591429	0.552000	0.571034	2:17:46
3	0.661245	0.667244	0.572267	0.595355	0.451200	0.513350	2:20:23

Table 5: Metrics when conducting initial training on the full dataset for 4 epochs.

epoch	train_loss	valid_loss	accuracy	precision_score	recall_score	f1_score	time
0	1.109331	0.795858	0.550667	0.568741	0.419200	0.482653	2:08:12
1	0.966887	0.749632	0.564267	0.589858	0.421867	0.491915	2:11:44

Table 6: Metrics for initial training on the full dataset for 2 epochs.

epoch	train_loss	valid_loss	accuracy	precision_score	recall_score	f1_score	time
0	0.902063	0.706253	0.574667	0.582742	0.525867	0.552846	3:12:02

Table 7: Metrics when fine-tuning for 1 epoch on full dataset.

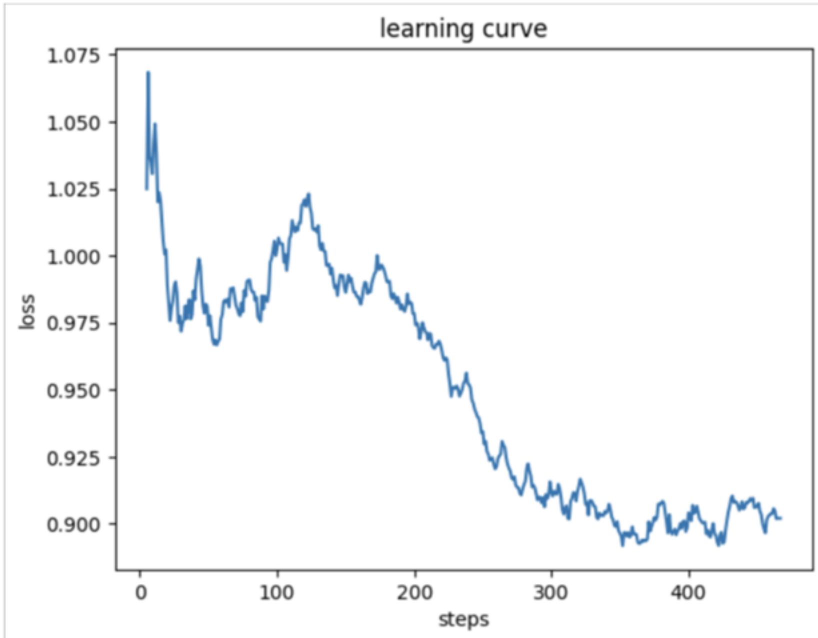


Figure 6: Learning curve: steps vs loss plot on the full dataset.

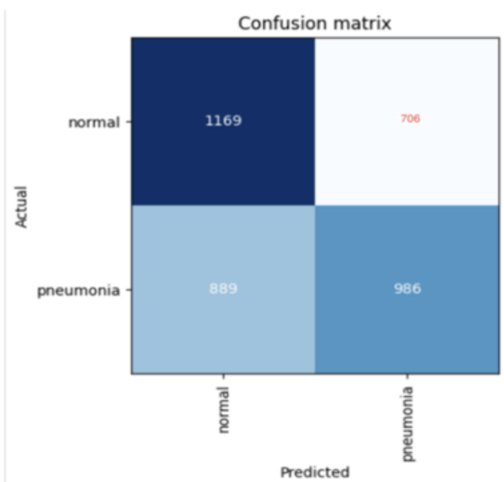


Figure 7: Confusion matrix of final result

6 Conclusion

Our research aims to build a framework to use deep learning models to efficiently diagnose pneumonia. Khan's dataset on Kaggle was considerably smaller in size with very clear and consistent images compared to CXLSeg, which led to us achieving much higher training accuracies than what we ended up with for CXLSeg. However, our values indicated strong overfitting and a lack of generalizability, underperforming on unseen data. Khan's dataset was also free of any foreign artifacts, such as pacemakers, a common occurrence found in the CXLSeg images. Our testing with the smaller dataset (2,160 X-rays) allowed us to find the optimal learning rate, which we found, accuracy-wise, to be a narrow region around the minimum point on the steepest drop of the learning rate vs. loss plot. We also experimented with the valley value for the learning rate, and while we found lower accuracy, the lower loss and significantly shorter time per epoch were essential in our ability to run the model on our full dataset.

Fine-tuning the model significantly improved the accuracy of our results with the smaller dataset. Implementing early stopping was also pivotal to the success of this, as not only did it save time, but it prevented overfitting by monitoring the validation loss and making sure the model continues to prioritise generalizability. Table 3 shows an uncertain precision score progression, meaning the model is not getting any better or worse at correctly identifying positive instances among all predicted positives. The recall score shows a steady increase, suggesting the model becomes better at identifying true positives. The F1 score, being a harmonic mean of precision and recall, increases through the epochs, implying a good trade-off between them. After hyperparameter optimization, the learning rate decreased, allowing more gradual updates to the weights and helping prevent overfitting by learning more subtleties in the images. The recall, precision, and F1 scores all hold up relative to pre-optimization. The continued negative trend in loss is also great at proving the lack of overfitting and generalization of the model. The test accuracy with fine-tuning was significantly better, proving the importance of fine-tuning. The learning curve's general negative trend is also a good sign of the model getting better for unseen data, and the heatmap produced by the Grad-CAM correctly focuses on the fluid and pus buildup in the left lung.

The full dataset that we used for our model was very large and contained lung segmentation done by the SA-Unet model. Unfortunately, this led to a significant number of images being segmented incorrectly and some of them even being completely blank. Though we cleaned as much of the dataset as we could manually, due to the sheer number of images, a lot of incomplete segmentations remained in the data, affecting the model's accuracy. The 12 hour Kaggle session timeout restriction was also a major hurdle we never fully overcame. We were unable to apply most of the changes we implemented that we learnt from training on the small dataset onto our full dataset. Due to a lack of powerful computers between either of us, we were unable to find an alternative to this. During our initial training run with the full dataset, where we ran 4 epochs, the accuracy was already very promising, and the decrease in loss was sharp (Table 5). The precision increased, the recall decreased, and the F1 also generally decreased, but generally fluctuated. With these initial training results, we believe that, if we were able to fine-tune and run the exact code as we did for the smaller dataset instead of having to decrease the number of epochs, our accuracy would've been comparable to that of Zhang (2024), where he achieved around 70% accuracy using various methods like downsampling and upsampling on the MIMIC-CXR-JPG dataset.

Instead, we were limited to 2 epochs to train, and 1 epoch for fine-tuning, which was very underwhelming but still produced a very strong negative learning curve, showing great promise in the ability of CNNs to benefit from and generalize well with big datasets of chest X-ray images. An accuracy of 56.96% test accuracy was surprisingly good for such few epochs to fine-tune the model, greatly outperforming the smaller dataset when it was not fine-tuned. The confusion matrix indicates that the model was strong at detecting true positives for 'normal' cases, which indicated non-pneumonia, but still struggles with determining whether images of lungs impacted by pneumonia are infected or not. By integrating Grad-CAM to produce heatmaps of the regions of interest to the CNN, we provided interpretability to our results in order to maximize our tool's applicability in the medical workspace.

Pneumonia can be classified into four categories: bacterial, viral, fungal and aspiration. Future work in this research field can help integrate the evaluation of these specific types of pneumonia in order to excel clinical diagnostics even further. Classification of other respiratory diseases such as tuberculosis and COPD could also be integrated into the same framework in the future, helping expand the usability of frameworks like ours. Through this study, we aim to contribute to the progression of artificial intelligence in medical diagnostics, which has the potential to produce accurate, efficient, and affordable interpretations of radiograph images, bringing a positive change to the medical industry.

Acknowledgements
We would like to thank Dr. Bart Massey at Portland State University for his constant support of our research.

The work was supported by the altREU program at Portland State University.

References

[1] World Health Organization. (n.d.). Pneumonia. World Health Organization. https://www.who.int/health-topics/pneumoniatab=tab1

[2] Abhisheka, B., Biswas, S. K., Purkayastha, B., Das, D., Escargueil, A. (2023). Recent trend in medical imaging modalities and their applications in disease diagnosis: A review. Multimedia Tools and Applications.

[3] Candemir, S., Antani, S. (2019). A review on lung boundary detection in chest X-rays. International Journal of Computer Assisted Radiology and Surgery, 14(4), 563–576.

[4] Cotes, J. E., Chinn, D. J., Miller, M. R. (2009). Lung function: Physiology, measurement and application in medicine. John Wiley Sons.

[5] Zhang, Y. (2024). Advancements in chest radiography pneumonia classification through fine-tuning using MIMIC-CXR-JPG dataset. https://www.mccormick.northwestern.edu/computer-science/documents/yifan-zhang-tr.pdf

[6] Johnson, A., Lungren, M., Peng, Y., Lu, Z., Mark, R., Berkowitz, S., Horng, S. (2024). MIMIC-CXR-JPG - chest radiographs with structured labels (version 2.1.0). PhysioNet. https://doi.org/10.13026/jsn5-t979

[7] Račić, L., Popović, T., Čakić, S., Šandi, S. (2021). Pneumonia detection using deep learning based on convolutional neural network. In 2021 25th International Conference on Information Technology (IT) (pp. 1-4). IEEE. https://doi.org/10.1109/IT51528.2021.9390137

[8] Pant, A., Jain, A., Nayak, K. C., Gandhi, D., Prasad, B. G. (2020). Pneumonia detection: An efficient approach using deep learning. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225543

[9] Verma, D., Bose, C., Tufchi, N., Pant, K., Tripathi, V., Thapliyal, A. (2020). An efficient framework for identification of tuberculosis and pneumonia in chest X-ray images using neural network. Procedia Computer Science, 171, 217-224. https://doi.org/10.1016/j.procs.2020.04.023

[10] Skandarani, Y., Jodoin, P. M., Lalande, A. (2023). GANs for Medical Image Synthesis: An Empirical Study. Journal of imaging, 9(3), 69. https://doi.org/10.3390/jimaging9030069

[11] Zhang, C., Deng, X., Ling, S. H. (2024). Next-gen medical imaging: U-Net evolution and the rise of transformers. Sensors, 24(14), 4668. https://doi.org/10.3390/s24144668

[12] Changlu Guo, M´arton Szemenyei, Yugen Yi, Wenle Wang, Buer Chen, and Changqi Fan. Sa-unet: Spatial attention u-net for retinal vessel segmentation, 2020. doi: https://doi.org/10.48550/ ARXIV.2004.03696.

[13] Johnson AE, Pollard TJ, Berkowitz S, Greenbaum NR, Lungren MP, Deng CY, Mark RG, Horng S. MIMIC-CXR: A large publicly available database of labeled chest radiographs. arXiv preprint arXiv:1901.07042. 2019 Jan 21.

[14] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. E215–e220.

[15] Mucinya, N. (2023). vit-finetuned-chest-xray-pneumonia [Model]. Hugging Face. https://huggingface.co/nickmuchi/vit-finetuned-chest-xray-pneumonia.

[16] Khan, A. (2023). Pneumonia detection using CNN (96% accuracy) [Kaggle Notebook]. Kaggle.https://www.kaggle.com/code/arbazkhan971/pneumonia-detection-using-cnn-96-accuracy

[17] L. P. B. (n.d.). Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*. https://pubmed.ncbi.nlm.nih.gov/28436741/

[18] Caruana, E. J., Roman, M., Hernández-Sánchez, J., Solli, P. (2015, November). Longitudinal studies. *Journal of Thoracic Disease*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4669300/

[19] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2019, December 3). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *arXiv.org*. https://arxiv.org/abs/1610.02391

[20] Zhu, N., Zhang, D., Wang, W., Li, X., Yang, B., Song, J., Zhao, X., Huang, B., Shi, W., Lu, R., Niu, P., Zhan, F., Ma, X., Wang, D., Xu, W., Wu, G., Gao, G. F., Tan, W. (n.d.). A novel coronavirus from patients with pneumonia in China, 2019. *The New England Journal of Medicine*. https://pubmed.ncbi.nlm.nih.gov/31978945/