8-18-2024

A Configurable Real-Time Event Detection Framework for Power Systems Using Swarm Intelligence Optimization

Umar Farooq
Portland State University

Midrar Adham
Portland State University

Mohammed Alsaid
Portland State University

Robert B. Bass
Portland State University

Citation Details

Farooq, U., Adham, M., Alsaid, M., & Bass, R. B. (2024). A Configurable Real-time Event Detection Framework for Power Systems using Swarm Intelligence Optimization. IEEE Access, 1–1. https://doi.org/10.1109/access.2024.3445312

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3445312

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xxx

A Configurable Real-time Event Detection Framework for Power Systems using Swarm Intelligence Optimization

UMAR FAROOQ[1,3], MIDRAR ADHAM[1], (Member, IEEE), MOHAMMED ALSAID[2], ROBERT B. BASS[1], (Member, IEEE),

[1]Department of Electrical & Computer Engineering, Portland State University, Portland, OR, USA
[2]Department of Computer Science, Portland State University, Portland, OR, USA
[3]National Grid ESO, Wokingham, UK

Corresponding author: Umar Farooq (ixumer@gmail.com)

ABSTRACT Power system balancing authorities are routinely affected by sudden frequency fluctuations. These frequency events can take the form of negligible frequency deviations or more severe emergencies that can precipitate cascading outages, depending on the severity of the disturbance and efficacy of remedial action schema. It is imperative to arrest such disturbances quickly by activating primary frequency control measures. This manuscript proposes a configurable event detection framework using optimization methods to tune a detection algorithm to detect events as specified by experts from a Balancing Authority. The utility of the detection framework is demonstrated using a regression-based frequency event detection algorithm with tunable parameters. Two swarm intelligence-based optimization algorithms, Grey Wolf Optimization and Particle Swarm Optimization, are applied to tune the parameters of the detection algorithm according to the definition of frequency events specified by experts.The performances of the GWO and PSO algorithms are analyzed, and the efficacy of the proposed system is demonstrated using an algorithm evaluation environment and a suite of evaluation metrics. The proposed event detection framework is capable of detecting events in real-time with high accuracy and speed using real-world, real-time phasor measurement unit data.

INDEX TERMS Frequency Event, Event Detection, Grey Wolf Optimization, Particle Swarm Optimization, Phasor Measurement Unit, Primary Frequency Response

Acronyms

AEE	Algorithm Evaluation Environment
BA	Balancing Authority
CNN	Convolution Neural Networks
DWT	Discrete Wavelet Transform
GWO	Grey Wolf Optimization
LSTM	Long Short-term Memory
NERC	North American Electric Reliability Corporation
PCA	Principal Component Analysis
PFR	Primary Frequency Response
PMU	Phasor Measurement Unit
PSO	Particle Swarm Optimization
ROCOF	Rate of Change of Frequency
RTAC	Real-time Automation Controller
SAE	Stack Auto-Encoders

I. INTRODUCTION

Two primary goals of system operators are to maintain stability of the power system and ensure continuity of supply to consumers in the event of disturbances. Modern power systems are more vulnerable to experiencing critical situations since they are operated close to steady-state stability limits to maximize use of capital [1]. Therefore, modern power systems require sophisticated monitoring and control schemes to identify and mitigate such disturbances at an early stage.

Cascading events in power systems have two stages. In the first stage, there is a slowly evolving process of consecutive events, which deteriorates system operating conditions. A transient action in the second stage results in cascading events and ultimately system collapse. A possible cascading event can be averted by taking proper control actions at an early stage [2]. The ubiquitous unpredictability prevailing in power systems, such as loss of generation and/or transmission line outage, creates hurdles for reliable operation. Stable operation of power systems is dictated by maintaining frequency within permissible limits as close as possible to nominal value of 50/60 Hz, around $\pm 0.2\%$. Frequency, therefore, is a key reliability aspect of power systems and failure to maintain it within predetermined limits may lead to disruption in consumer supply, outage of generators, and possibly cascading failures [3].

Disturbances in power systems can be initiated by numerous factors such as generator outages, transmission line tripping, lightning strikes, equipment failure, human error, and substandard maintenance [4]. The complexities and uncertainties associated with maintaining frequency stability and grid resilience of power systems have increased to an unprecedented level with the growing adoption of distributed and renewable power sources. The rapid decline in system rotational inertia due to replacement of synchronous generators with inverter-based generators has further aggravated the situation [5]. The inability to implement state-of-the-art techniques for modern problems will result in imposing conservative caps on allowable renewable energy capacity to preserve system security, hindering the transition to a more modern infrastructure [6].

The recent large-scale deployment of Phasor Measurement Units (PMUs) have enabled operators to have improved real-time situational awareness of the operational state of power systems. PMUs provide precise, time-synchronized local measurements of system frequency along with voltage and current phasors. With the advancement of synchrophasor technology, PMUs can now record data at a high rate, in the range 30-120 samples per second [3]. High resolution PMU data allow monitoring of sharp dynamic information throughout the network. For managing the health of critical infrastructure and maintaining stability of the power system, accurate detection of events plays a vital role in timely triggering remedial action schema and successful restoration of service. Requirements for triggering of frequency control assets after an event vary by jurisdiction. In Great Britain, Primary Frequency Response (PFR) needs to activated within two seconds of the triggering event, with full provision of the requisite power within ten seconds [7]. The Australian Energy Market Operator mandates a 5% increase in active power achieved within ten seconds of the frequency deviation from PFR deadband, which is ± 1.5 Hz around the nominal value [7].

The definition of an event is not absolute; it varies for different balancing areas depending upon critical stability limits. The North American Electric Reliability Corporation (NERC) published the Frequency Response Standard Background Document BAL-003 [8] wherein frequency events are extensively discussed, but NERC provides no standard definition as to what qualifies as a frequency event. This is not an oversight; large stable interconnects with enormous synchronous inertia are less sensitive to frequency deviations while smaller isolated powers systems, or in ones where system stability is already compromised, a Balancing Authority (BA) may be interested in arresting even minor frequency deviations to prevent potential system collapse. Therefore, detection algorithms must be configured to meet specific system requirements for each BA.

The contributions of this work address the customization of frequency event detection algorithms using optimization techniques. These contributions are discussed in two parts. In the first part, an Algorithm Evaluation Environment (AEE) is described that can be used to tune a frequency event detection algorithm to enable detection that matches the definition of frequency events as specified by power systems experts. The performance of the AEE is demonstrated using a linear regression-based event detection algorithm with five tunable parameters that can be adjusted to enable desired performance. In the second part, swarm intelligence optimization is used to optimally tune the parameters of the detection algorithm to meet the particular needs of balancing authorities. The tuning process is independent of the detection algorithm and works for any detection algorithm that has adjustable parameters. Once tuned, the algorithm is uploaded into an automation controller for real-time decision support using streaming PMU data to identity events. Two optimization algorithms were applied to this problem: Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO). Both algorithms were tested and evaluated using the AEE, which uses a set of performance evaluation metrics to quantify results.

II. BACKGROUND

Automatic event detection in power systems has engaged researchers lately and extensive work has been reported in literature on the topic. Contemporary work on power system event detection reported in literature is divided into three main categories: signal processing methods, statistical analysis methods, and machine learning methods.

A. SIGNAL PROCESSING BASED METHODS

Event detection techniques based on signal processing often use Discrete Wavelet Transform (DWT) [9]–[11] or discrete time filtering [12] to decompose frequency and voltage signals for detection of disturbances in a network. However, these schemes need measurement data from every bus in the network for reliable operation. Moreover, since the range of coefficient energy depends on window size, the results are highly susceptible to variations in window size. The performance of many signal processing based methods [9]–[12] is also affected by the requirement of a user-defined threshold for proper event detection. The threshold values depend on the quality and nature of PMU data and need to be configured for each PMU in the network, which is a challenge.

B. STATISTICS BASED METHODS

Principal Component Analysis (PCA)-based approaches have been widely used for event detection recently [13], [14]. Xu and Overbye proposed a scheme using PCA to analyze dynamic behavior of the system and highlight dominating buses after a disturbance [14]. An ensemble technique was proposed by Pandey et al. using several statistics and clustering techniques [15]. Recently, Zhu and Hill used a spatial-temporal data analysis based method for event detection [16].

Detection methods based on PCA are heavily dependent on training data. Performance of PCA-based supervised learning techniques is negatively affected with the selection of an improper and insufficient sample space. Moreover, statistical indices such as mean, variance, minimum, maximum, and correlation are used over a window to reduce computational complexity, but these values may vary even during the same event. Hence, it is very challenging to define a threshold.

C. SUPERVISED LEARNING METHODS

With recent developments in advanced data processing resources and machine learning techniques, deep learning algorithms such as Convolution Neural Networks (CNN), Stack Auto-Encoders (SAE), and Long Short-term Memory (LSTM) have been widely employed for solving various power system problems [17]–[20]. These supervised learning methods can suffer from an inadequate amount of labeled training data. Frequency events occur rarely, just a few per month in well-operated interconnections. Secondly, utility event logs often miss many events. Irregular labelling of learning data may result in a biased model. Inappropriate or inadequate selection of training data adversely affects the efficiency and accuracy of these models.

Apart from the limitations discussed above, the detection methods reported in literature do not allow customization of the algorithm to enable event detection as required by system operators in the context of their system conditions. The features that distinguish the approach presented in this paper from other methods are customization, robustness, simplicity, and not requiring multiphasor, multibus measurements.

III. EVENT DETECTION METHOD

The presented event detection algorithm uses a least-squares linear regression method. The regression detection algorithm is a threshold-based detection algorithm that uses Rate of Change of Frequency (ROCOF) derived from linear regression. A regression line is placed by minimizing the square of vertical distance from the data points, also known as variance.

$$Y = a + bX + u \tag{1}$$

Y is the dependent variable, X is the independent variable, a is the intercept, b is slope, and u is the regression residual.

The dependent variable is frequency and the independent variable is time. Since we are interested in determining a smoothed ROCOF, referred to as the slew rate, we use the slope of the regression line for our calculation, give by:

$$Slope\ (Slewrate) = \frac{N\sum(time.freq) - \sum(time)\sum(freq)}{N\sum((time)^2) - (\sum(time))^2} \tag{2}$$

$time$ is the independent variable, $frequency$ is the dependent variable, and N is the window size.

The purpose of using a linear regression rather than the derivative of frequency and ROCOF is to compensate for noise in the PMU data. The small deviations in frequency that exist at high sampling rates cause the derivative of frequency to fluctuate, making threshold detection challenging. Similarly, ROCOF data calculated by PMUs undergo continuous variation and are prone to noise. Fluctuations in frequency increase with the increase in sampling rate of PMUs, which amplifies the noise level. ROCOF does not provide a smooth estimation of frequency trend and suffers from fluctuations and uncertain values. Linear regression produces a smooth derivative of frequency data, eliminating the need to filter the raw PMU data before using it. Deviation in slew is an indication of frequency instability, as shown in Figure 1, lower plot.

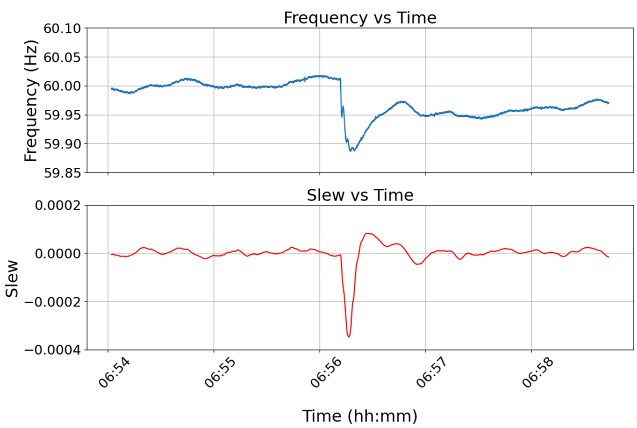

FIGURE 1: Frequency instability (blue) and the corresponding drop in slew rate (red)

A. DETECTION ALGORITHM PARAMETERS

The detection algorithm calculates slew rate of frequency data obtained from PMUs using Eq. 2. Under normal system conditions slew rate is almost constant. In case of an event, frequency changes abruptly and thus slew rate experiences a sudden rise or fall (Figure 1). Based on this sudden change in slew rate, an event is identified. The algorithm has five tunable parameters, which dictate its performance.

PARAMETER 1: WINDOW SIZE

Slew rate of the frequency is calculated over a sliding window using Eq. 2. The window moves forward in steps of one sample. Choosing an appropriate window size is important as it affects the detection speed and immunity to noise.

Let t_1, t_2, $t_3 ... t_N$ represent the timestamps of PMU frequency measurements and f_1, f_2, $f_3 ... f_N$ represent frequency measurements, then the slew rate λ calculated over a window of length N is given by:

$$\lambda_k = \frac{\sum_{i=1}^{N} t_i f_i - \sum_{i=1}^{N} t_i \sum_{i=1}^{N} f_i}{\sum_{i=1}^{N} t_i^2 - (\sum_{i=1}^{N} t_i)^2} \tag{3}$$

PARAMETER 2: POINT SEPARATION

With every new frequency measurement from PMU, the window moves forward in one sample step and calculates slew rate (λ_k) for the new window. To detect a rise or fall, two slew points are compared with each other. These compared values are separated at a distance defined by this parameter. A gap of more than one improves the detection speed.

Let λ_1, λ_2, $\lambda_3 ... \lambda_n$ represent slew calculated over each sliding window, then slew vector is given by:

$$\vec{\lambda} = [\lambda_1, \lambda_2, \lambda_3 ... \lambda_n] \tag{4}$$

The difference of slew is calculated between two values that are separated by a number of points defined by n_{ps}.

$$D_{slew,i} = |\lambda_{i+n_{ps}} - \lambda_i| \tag{5}$$

Calculation of λ and D_{slew} are performed in run time.

PARAMETER 3: SLEW DIFFERENCE THRESHOLD

The magnitude difference between any two slew points is compared with a threshold (Th_{sd}) to detect sudden rise or fall in a frequency trend. This threshold will only be exceeded in case of a frequency disturbance.

$$if\ D_{slew,i} > Th_{sd}$$
$$frequency\ deviation\ has\ started$$

PARAMETER 4: SERIES OVER THRESHOLD

Since every frequency instability does not necessarily lead to an event, series over value is incremented with every consecutive slew difference that exceeds Th_{sd}. The algorithm declares an event when series over value exceeds series over threshold (Th_{so}), which implies that there is a constant rise or fall in frequency.

$$if\ D_{slew,i} > Th_{sd}$$
$$Series\ Over = Series\ Over + 1$$

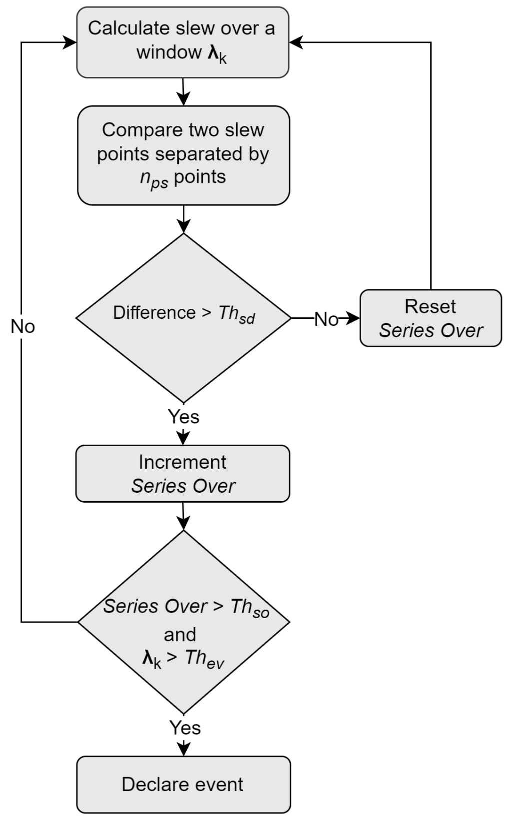

FIGURE 2: Event detection using a linear regression algorithm, detailing the five parameters of the algorithm.

PARAMETER 5: EVENT THRESHOLD

The slew deviation from the normal value achieved in a given time in case of an event is greater than in case of a non-event frequency deviation, owing to the rapid rate of change during an event. Event threshold (Th_{ev}) is another parameter that checks if, after series over threshold is exceeded, slew deviation has reached a certain level indicative of an event.

$$if\ Series\ Over > Th_{so}$$
$$check\ for\ Th_{ev}$$

Figure 2 presents a flowchart explaining the event detection process with five parameters of the regression algorithm.

B. ALGORITHM EVALUATION ENVIRONMENT

1) Frequency Archive and Test Station

Our research group maintains a data archive that contains around three years of PMU data. Frequency event dates and times were provided by Portland General Electric, which were used as reference to extract frequency data from the archive. These events were identified by both Portland General Electric and NERC. Using these test files, frequency events were replicated within an NHR 9410 Grid Simulator. The simulator is monitored by an SEL 351 PMU that is connected to an SEL 3555 Real-time Automation Controller

(RTAC) where the detection algorithm resides. This system, as shown in Figure 3, allows for reproducible, real-time event detection.

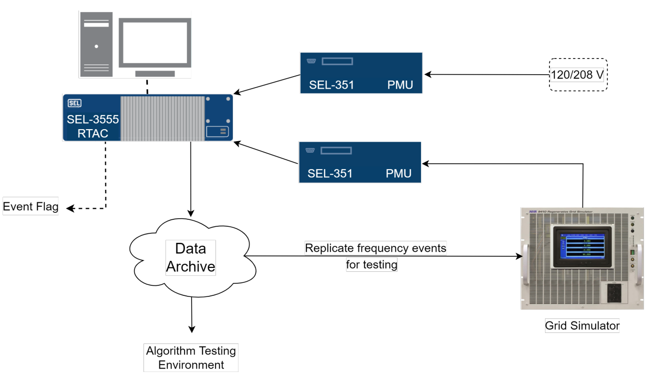

FIGURE 3: PSU real-time Frequency Response Test Station

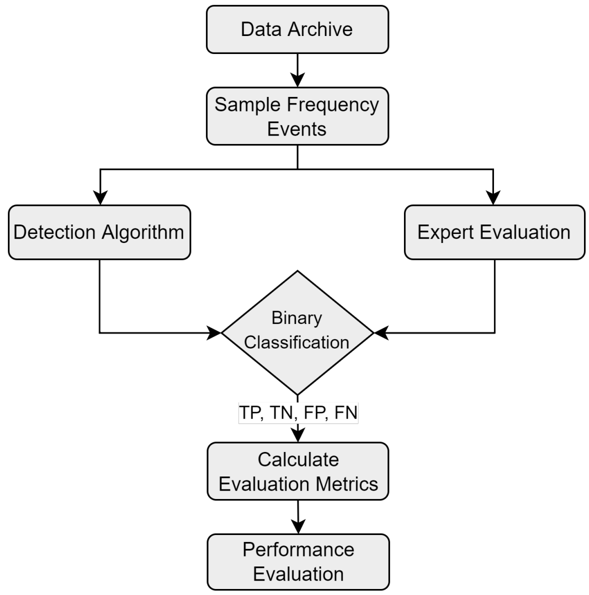

FIGURE 4: Performance evaluation of detection algorithm based on binary classification metrics

2) Expert Evaluation

The set of test files extracted from the archive was presented to a group of experts for evaluation. For improved performance of the algorithm, the group of experts can be chosen as industry or academic professionals with a relevant background. Moreover, a set of decision rules can be defined for declaring the file as either an event or not. This ensures consistency in evaluation pattern and avoids ambiguous cases. The evaluation was carried out using an online survey where each test frequency plot was presented to the experts and their evaluation was recorded. The survey considers the expertise level of every participant to weigh their assessments. The survey produces a human validation file containing event file names, experts names, and their assessments. A final evaluation of each frequency file was then provided using weighted assessment of the experts based on their expertise level. A sample human validation file is tabulated in Table 1. Such a human validation file can be created for any BA to specify the type of events to be arrested.

TABLE 1: A sample human validation file containing file names, experts' assessments, and final declarations

Name	Expert 1	Expert 2	Expert 3	Is_event
2019-08-02-15-34_11.csv	OF event	OF event	OF event	True
2019-10-03-18-20_5183.csv	UF event	UF event	UF event	True
2019-09-05-08-51_7722.csv	No event	No event	No event	False
⋮	⋮	⋮	⋮	⋮

3) Detection Algorithm Performance Evaluation

Performance of the detection algorithm is evaluated against human assessment using a suite of performance evaluation metrics, Table 2 [21]. Results from binary classification are used to calculate these evaluation metrics. One of four binary classifications are identified for each event. Figure 4 shows a flowchart of the process.

True Positive (TP): the algorithm correctly identifies an event, as declared by the expert.
True Negative (TN): the algorithm correctly identifies a non-event frequency deviation, as declared by the expert.
False Positive (FP): the algorithm incorrectly identifies an event that is declared as non-event by the expert.
False Negative (FN): the algorithm does not identify an event that is declared an event by the expert.

C. PARAMETER ADJUSTMENT

Selection of an adequate set of parameters for the proposed event detection algorithm is required to correctly identify events. Since the algorithm has five tunable parameters, a hypervolume of suitable solution sets likely exists. However, finding an optimal set of parameters in the vast five-dimensional search space is a challenging task. Manual adjustment of these parameters with a simple grid search would be a laborious and likely fruitless process. Therefore, an automated optimization process is preferred. Two meta-heuristics based on swarm theory, GWO and PSO, are considered in this work to adjust the parameters aiming to detect events as per the human validation. The motivation behind the use of meta-heuristics is their simplicity, flexibility, derivation-free mechanisms, and the ability to avoid local optima [22]. They have been gaining in popularity within the power community lately, and have produced superior results for various problems [23]–[25].

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3445312

TABLE 2: Performance Evaluation Metrics

$Accuracy = \frac{TP+TN}{Sample\ Size} \times 100\%$	Algorithm performance in terms of classifying events and non-events correctly.
$Sensitivity = \frac{TP}{TP+FN} \times 100\%$	Algorithm ability to correctly detect events.
$Precision = \frac{TP}{TP+FP} \times 100\%$	Measures how many of the positively identified events are actual events.
$Specificity = \frac{TN}{TN+FP} \times 100\%$	Measures the ability to correctly identify non-events.
$FDR = \frac{FP}{TP+FP} \times 100\%$	Measures the tendency to falsely identify an event.

1) Particle Swarm Optimization

PSO is a meta-heuristic based on swarm theory, proposed with the idea to produce computational intelligence by simulating collective and social intelligence of bird flocks or fish schools [26]. A set of particles, characterized by position and velocity vectors, moves through the search space and cooperatively searches for a solution. The movement of each particle in the search space is determined by using its current and previous best position with the best position obtained in the swarm so far, along with a random deviation. The swarm moves as a flock in search of an optimum. After a random initialization of the particle positions, velocity and position are updated in each iteration using the following expressions.

$$\vec{v_i}^{t+1} = w\vec{v_i}^{t} + c_1 r_1^t \otimes (p\vec{B_i}^{t} - \vec{x_i}^{t}) + c_2 r_2^t \otimes (g\vec{B}^{t} - \vec{x_i}^{t}) \quad (6)$$

$$\vec{x_i}^{t+1} = \vec{x_i}^{t} + \vec{v_i}^{t+1} \quad (7)$$

Variable w is the inertia weight. c_1 and c_2 are acceleration coefficients. r_1 and r_2 are random numbers between 0 and 1. The operator $\otimes$ is element-wise multiplication. The vectors $\vec{v_i}^{t+1}$ and $\vec{x_i}^{t+1}$ are particle velocity and position at the next iteration, respectively. The vectors $p\vec{B_i}^{t}$ and $g\vec{B}^{t}$ represent the best position found by a given particle and the best position found by the swarm at iteration t, respectively. The setting of these simulation parameters i.e. w, c_1, and c_2, is well-discussed in the literature [27], [28]. High values of c_1 and c_2, and a low value of w will discourage exploration and cause premature convergence of the particles. To realize a balance between exploration and exploitation, both c_1 and c_2 are set to 2, and w is linearly decreased from a maximum set value (0.9) to a minimum set value (0.2) over the course of iterations [29].

pB_i and gB at iteration $t+1$ are given by Equations 8 and 9. N is the swarm size.

$$p\vec{B_i}^{t+1} = \begin{cases} \vec{x_i}^{t+1}, \ if\ Fitness\ >\ p\vec{B_i}^{t} \\ p\vec{B_i}^{t}, \ if\ Fitness\ \leq\ p\vec{B_i}^{t} \end{cases} \quad (8)$$

$$g\vec{B}^{t+1} = p\vec{B_i}(Max\ Fitness), \ for\ 1\ \leq\ i\ \leq\ N \quad (9)$$

The parameters considered for the 5-dimensional optimization problem are arranged in search agent positions as follows.

$$\mathbf{x} = [N,\ n_{ps},\ Th_{sd},\ Th_{so},\ Th_{ev}]^T \quad (10)$$

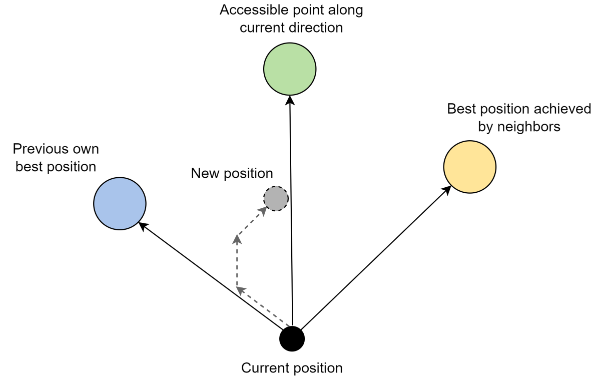

FIGURE 5: Movement of a particle in the search space [30]

Since some parameters in the problem are integer variables, the particle positions associated with these variables are rounded to the nearest integer, following a well-established process described in literature [31], [32].

The objective function is formulated using a sum of evaluation metrics, which makes the optimization process a maximization problem. The objective function is given by Eq. 11.

$$Max(Accuracy + Sensitivity + Precision + Specificity) \quad (11)$$

By maximizing the objective function, one can search for a set of parameters to enable a detection algorithm to detect only the events identified by the experts.

The event selection used in parameters adjustment is a crucial deciding aspect, given that event identification is dictated by the analysts, thereby defining the objectives behind implementation of the proposed method.

2) Grey Wolf Optimization

GWO is inspired by the hunting approach and social hierarchy of grey wolves [22]. The social hierarchy has four types of wolves: alpha, beta, delta, and omega. The group hunting behavior of grey wolves, which involves the following phases, is simulated for search space modeling of GWO.

- Searching and approaching the prey
- Encircling and harassing
- Attacking

Exploration of the search space is inspired by the first two phases. The last phase simulates exploitation. In the

PSO Algorithm

Initialize a population of particles with random positions and velocities.
main loop
  for every particle
    Define limits of the search space in each dimension.
    Evaluate position using the objective function.
    Compare fitness score with $pBestScore_i$. Update $p\vec{B}_i$ and $pBestScore_i$ if current fitness is better than the previous best.
    Update $g\vec{B}$ with the global best position.
    Update w, calculate velocities, and move particles using Eq. 6 and 7.
  end **for**
Terminate loop if end criteria is met (acceptable fitness score or maximum iterations).
end **main loop**
return $g\vec{B}$

GWO Algorithm

Initialize a random population of search agents.
Initialize a, A and C.
main loop
  for every search agent.
    Define limits of the search space in each dimension.
    Evaluate position using the objective function.
    Compare fitness score with α, β, and δ score. Update X_α, X_β, and X_δ based on the current fitness value.
    Update a, A, and C.
    Update position using Eq. 14.
  end **for**
Terminate loop if end criteria is met (acceptable fitness score or maximum iterations).
end **main loop**
return X_α

mathematical representation of social hierarchy of grey wolves, the best solution in the solution space is considered as alpha (α). The second and third best solutions are called beta (β) and delta (δ) respectively. α, β, and δ explore the search space during the optimization process. The remaining candidate solutions, represented as omega (ω), follow these search agents, as depicted in Figure 6.

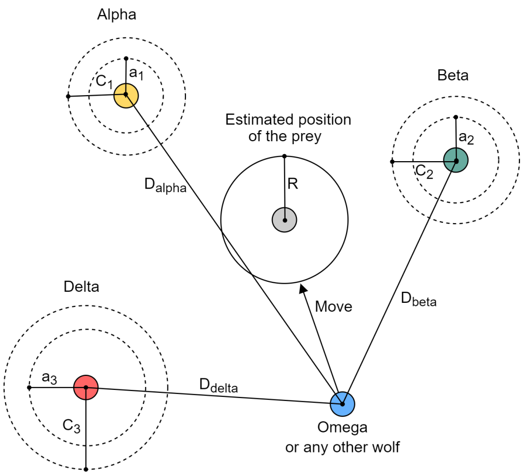

FIGURE 6: Omega or any other wolf updating position with respect to alpha, beta, and delta [22]

After a random initialization of wolves (the search agents), the position of each search agent is updated in every iteration using the following expressions. These illustrate that a wolf moves randomly within a circle around the prey, whose position is estimated by alpha, beta, and delta.

$$\vec{D_\alpha} = |\vec{C_1}\vec{X_\alpha} - \vec{X}|, \vec{D_\beta} = |\vec{C_2}\vec{X_\beta} - \vec{X}|, \vec{D_\delta} = |\vec{C_3}\vec{X_\delta} - \vec{X}| \tag{12}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1}\vec{D_\alpha}, \vec{X_2} = \vec{X_\beta} - \vec{A_2}\vec{D_\beta}, \vec{X_3} = \vec{X_\delta} - \vec{A_3}\vec{D_\delta} \tag{13}$$

$$\vec{X}^{t+1} = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{14}$$

$$\vec{A} = 2\vec{a}\vec{r_1} - \vec{a} \tag{15}$$

$$\vec{C} = 2\vec{r_2} \tag{16}$$

$\vec{X_\alpha}$, $\vec{X_\beta}$, and $\vec{X_\delta}$ are the position vectors of α, β, and δ, which represent three best solutions during the search. $\vec{X}$ is the position vector of all other wolves. $\vec{D_\alpha}$, $\vec{D_\beta}$, and $\vec{D_\delta}$ calculates the distance of α, β, and δ from the rest of the wolves, respectively. $\vec{X_1}$, $\vec{X_2}$, and $\vec{X_3}$ are the position vectors calculated with respect to the positions of α, β, and δ, respectively. $\vec{A}$ and $\vec{C}$ are coefficient vectors, t indicates current iteration, $\vec{r_1}$ and $\vec{r_2}$ are random vectors between 0 and 1, and $\vec{a}$ is an important parameter whose values is linearly decreased from 2 to 0 over the optimization process.

IV. RESULTS AND ANALYSIS

A Python development environment was used to develop and test the event detection and optimization algorithms. The detection algorithm was then implemented in the RTAC to validate its performance in real-time. A data set of 50 frequency files from the PMU data archive, with each file containing 10 minutes of frequency data, was used to run simulations. The data set contains obvious events, non-events, and quasi-events, all recorded at 30 frames per second.

A. PERFORMANCE EVALUATION

The data set of 50 files consists of 21 events and 29 non-events, as declared by the experts. The boundary of the solution space for each of the five parameters is given in Table 3. The corresponding human validation file for the data set was created from the evaluation of three experts. Table 4 shows the adjusted parameters using GWO and PSO. Comparison of the best solutions obtained using GWO and PSO along with the simulation parameters is presented in

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3445312

Table 5. The maximum possible fitness value is 400, as given by Eq 11.

The convergence curves for both GWO and PSO are shown in Figure 7. The optimized parameters were used in the detection algorithm and tested for the 50 files. GWO-optimized parameters produced superior results. Out of 21 events, 20 were detected correctly. Since occurrence of events in power systems is a rare phenomenon, one of the most important features of a detection algorithm is to identify normal frequency deviations and not issue false detection flags. *Precision* and *Specificity*, which account for FPs, are the best metrics for considering issuance of false detections. The proposed algorithm was capable of differentiating between events and minor frequency deviations. Performance evaluation metrics and binary classification for offline testing are given in Table 6. Online testing produced similar results, except for one additional FP. As can be seen in Figure 8, the event that triggered this FP has an unusually high noise level, likely induced by nearby equipment, which caused unwarranted triggering of the online testing system.

Considering *Precision*, a value of 95% implies that one out of 29 non-events was falsely given as an event. This can happen if the frequency deviation is severe like an event. The frequency file that was falsely detected as an event is shown in Fig 9. The frequency increases from 59.965 Hz to 60.01 Hz (0.045 Hz) in 9 seconds. The abrupt deviation in frequency gives rise to a spike in the slew rate, although the frequency is oscillating within a permissible band. Some utilities might be interested in detecting such

TABLE 3: Boundary of the solution space for the five dimensional optimization problem

	Window (samples)	Point Separation	Slew Diff	Series Over	Event Threshold
Upper Bound	250	30	0.0002	30	0.0001
Lower Bound	100	3	1×10^{-7}	3	1×10^{-6}

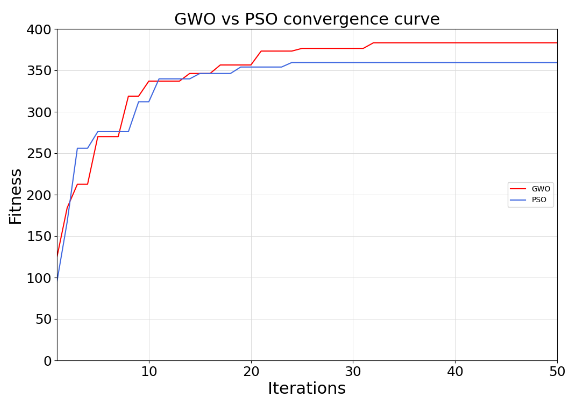

FIGURE 7: Convergence curve for GWO and PSO for the data set with 50 files. GWO achieved higher fitness value.

TABLE 4: Adjusted parameters using GWO and PSO

	Window (samples)	Point Separation	Slew Diff	Series Over	Event Threshold
GWO	216	3	0.00000300	13	0.0000378
PSO	150	25	0.00005966	29	0.0000001

TABLE 5: Comparison of best solutions obtained with GWO and PSO

Algorithm	Iterations	Search Agents	Fitness	data set Size	No. of Events	No. of non-events
GWO	50	10	383	50	21	29
PSO	50	10	359	50	21	29

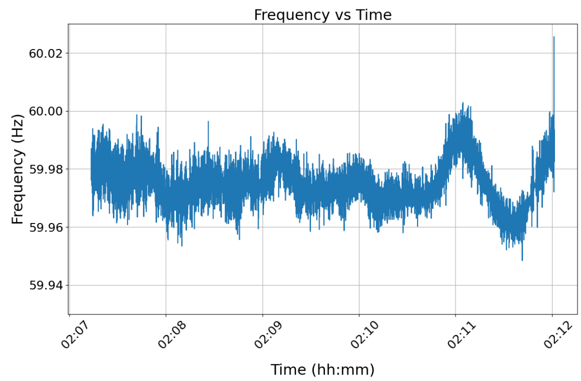

FIGURE 8: FP in online testing caused by noisy PMU data.

abrupt deviations, depending on the system conditions.

The superior performance of GWO over PSO for this particular problem may be explained by the No Free Lunch Theorem [33], which proved that no single meta-heuristic

TABLE 6: Performance evaluation metrics obtained using optimized parameters

Algorithm	Accuracy (%)	Sensitivity (%)	Precision (%)	Specificity (%)	FDR (%)	TP	FP	FN	TN
GWO	96	95	95	97	5	20	1	1	28
PSO	88	71	100	100	0	15	0	6	29

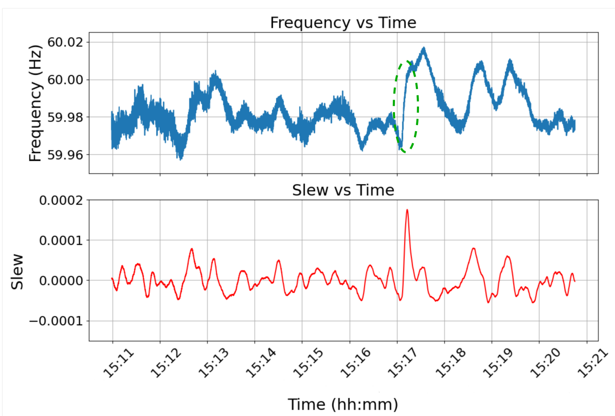

FIGURE 9: An abrupt deviation in frequency giving rise to a spike in the slew rate and false detection signal.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3445312

can perform effectively for all optimization problems. This particularly means that an algorithm may have superior performance for some set of optimization problems but poor performance for another set.

B. EVENT DETECTION SPEED

GWO-optimized parameters were used for determining the speed of the detection algorithm and the frequency samples between event inception and detection points were measured.

For online testing of the detection algorithm, samples are measured of streaming PMU data from an NHR 9410-12 Grid Simulator. To determine the detection speed for a higher sampling rate, the same frequency data files were subject to linear interpolation to effectively increase the sample rate to 60 samples per second. GWO was then performed on these interpolated files to obtain updated detection parameters that accommodate the temporal closeness of the frequency samples. The updated parameters (with a smaller window size) for the highly sampled files produced the same fitness score but improved the detection time. Figure 10 shows an interpolated file against the original file. Table 7 presents detection speed in terms of mean and standard deviation of number of samples for both online and offline testing.

TABLE 7: Mean and Standard deviation of event detection speed for files recorded at 30 and 60 samples per second.

Sampling rate	Test mode	Mean (No. of Samples)	Standard Deviation (No. of Samples)
30 Hz	Offline	41	21
	Online	42	22
60 Hz	Offline	48	24
	Online	50	25

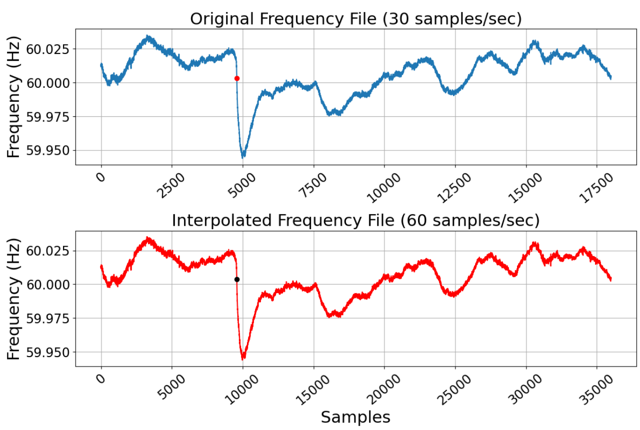

FIGURE 10: An interpolated frequency file against the original file. Red and black dots highlight the sample at which event is detected, respectively.

The number of samples it takes for the detection algorithm to identify an event remained almost the same for highly sampled files and hence absolute detection speed was improved. With the advent of synchrophasor technology, PMUs can now record data at up to 120 samples per second. Thus, detection speed can be further improved with higher PMU sampling rates, which can be used in low inertia systems. The decline in system synchronous inertia causes frequency events to exhibit higher ROCOF, thus necessitating higher detection speed.

C. EFFECT OF INCONSISTENT EXPERT EVALUATION

The consistency in event assessment during expert evaluation is of critical importance. If two similar frequency curves are assessed differently by human experts, the detection algorithm will not be able to differentiate between them and we can expect a FP or FN signal. To demonstrate this point, the same data set was assessed by a group of semi-experts (somewhat-trained undergraduates). In the new human validation file, 29 files were assessed as events and 21 as non-events. The same data set with the new human validation file was processed by the GWO algorithm. Due to inconsistency in the event assessment process, the maximum fitness achieved was 345, as presented in Table 8.

TABLE 8: Fitness value obtained for the same data set with semi-expert event assessment.

Iterations	Search Agents	Fitness	data set size	No. of Events	No. of non-events
50	10	345	50	29	21

TABLE 9: Performance evaluation metrics obtained for the same data set with semi-expert event assessment.

Accuracy (%)	Sensitivity (%)	Precision (%)	Specificity (%)	FDR (%)	TP	FP	FN	TN
84	79	92	90	8	23	2	6	19

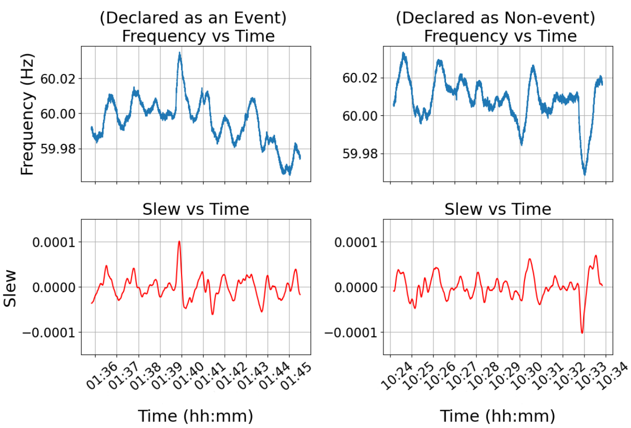

FIGURE 11: Examples of inconsistent event assessments.

The new adjusted parameters were used in the detection algorithm and tested. Table 9 presents the performance evaluation metrics and binary classification. False detection of two non-events is reflected in the decline of both *Precision* and *Specificity*. FPs need to be avoided as they lead to the unwarranted triggering of frequency response assets.

Such an ambiguity caused by the inconsistency in the event assessment process is shown in Figure 11. Although the frequency deviation and the rate of change of frequency are almost the same for both plots, the frequency instability on the left is assessed as an over-frequency event whereas the one on the right is assessed as a non-event by the semi-experts. In such a situation, the optimization algorithm cannot reach a best possible solution that can clearly differentiate between the two cases. Hence, the fitness value is low.

V. CONCLUSION

The definition of a frequency event may vary between different balancing authorities depending upon their critical stability limits. This work presents a configurable real-time event detection method using least-sum-of-squares linear regression. Algorithm parameters can be tuned to match the definition of frequency events as specified by experts. The linear regression algorithm is one of many detection algorithms that can detect frequency events. The contribution of this paper is the use of swarm-intelligence algorithms to tune parameters of any detection algorithm, not just the linear regression algorithm. Two swarm intelligence-based optimization algorithms, GWO and PSO, were applied to automate the parameter tuning process and determine optimal parameters and thresholds.

The method is applied to frequency data obtained from the PMU installed at Portland State University. Results show that the proposed method is able to efficiently identify events, as reported by experts, within an acceptably short period of time following event onset. Detection speed can be further improved with higher PMU sampling rates. Applying optimization algorithms to define parameters through data training not only addresses a typical concern of industry on how to define these parameters for real time implementation but also improves detection metrics.

Future work will improve the expert assessment survey process to better present a data set of candidate frequency plots to a experts using an online database. Weighting of performance metrics within the objective function is also an active area of research.

We have recently completed development of a real-time frequency response test bed, Figure 3. The test bed includes an NHR grid simulator, which replays frequency events in real-time, and an SEL RTAC, which host the frequency response algorithm. Preliminary results show that well-tuned algorithms are able to detect events within 500 to 700 ms of the onset of an event, and with both high precision and specificity. This is well before the nadir of frequency events, typically two to three seconds after onset of the event. We are now in the process of coupling this real-time event detection system to a battery energy storage system. This will allow us to investigate event detection time, communication lags, and asset response delays that occur between the onset of an event and the moment when restoration by a dispatchable frequency response asset begins. We plan to present these findings in a future publication.

REFERENCES

[1] H. H. Alhelou, An Overview of Wide Area Measurement System and Its Application in Modern Power Systems. Handbook of Research on Smart Power System Operation and Control, IGI Global, 2019.

[2] H. Song and M. Kezunovic, “A new analysis method for early detection and prevention of cascading events,” Elec. Power Sys. Res., vol. 77, pp. 1132–1142, June 2007.

[3] U. Farooq and R. B. Bass, “Frequency event detection and mitigation in power systems: A systematic literature review,” IEEE Access, vol. 10, pp. 61494–61519, 2022.

[4] P. Kundur, N. J. Balu, and M. G. Lauby, Power system stability and control. EPRI power system engineering series, 1994.

[5] A. Saleki, M. T. Bina, and A. Shahirinia, “Suggesting hybrid HB and three-quarter-bridge MMC-based HVDC systems: Protection and synchronous stability under DC faults,” IEEE Trans. on Power Delivery, vol. 37, no. 4, pp. 2693–2703, 2022.

[6] M. A. Adham, M. Obi, and R. B. Bass, “A field test of direct load control of water heaters and its implications for consumers,” in IEEE Power & Energy Soc. Gen. Meeting, 2022.

[7] U. Farooq, “Development of a configurable real-time event detection framework for power systems using swarm intelligence optimization,” Master’s thesis, Dept. of Elec. & Comp. Eng., Portland State Uni., Portland, OR, 2022.

[8] NERC, “Frequency response standard, background document,” 2012.

[9] A. Anshuman, B. K. Panigrahi, and M. K. Jena, “A novel hybrid algorithm for event detection, localisation and classification,” in 9th IEEE Int. Conf. Power Syst., Dec. 2021.

[10] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, “Wavelet-based event detection method using PMU data,” IEEE Trans. Smart Grid, vol. 8, no. 3, pp. 1154–1162, 2017.

[11] R. Vaz, G. R. Moraes, E. H. Arruda, J. Terceiro, A. F. Aquino, I. C. Decker, and D. Issicaba, “Event detection and classification through wavelet-based method in low voltage wide-area monitoring systems,” Int. J. of Elect. Power & Energy Sys., vol. 130, Sept. 2021.

[12] S. A. R. Konakalla and R. de Callafon, “Optimal filtering for grid event detection from real-time synchrophasor data,” Procedia Comp. Sci., vol. 80, pp. 931–940, June 2016.

[13] L. Xie, Y. Chen, and P. R. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” IEEE Trans. Power Sys., vol. 29, pp. 2784–2794, Nov. 2014.

[14] T. Xu and T. Overbye, “Real-time event detection and feature extraction using PMU measurement data,” in IEEE Int. Conf. Smart Grid Comm., pp. 265–270, Nov. 2015.

[15] S. Pandey, A. K. Srivastava, and B. G. Amidan, “A real time event detection, classification and localization using synchrophasor data,” IEEE Trans. Power Sys., vol. 35, pp. 4421–4431, Nov. 2020.

[16] L. Zhu and D. J. Hill, “Spatial-temporal data analysis based event detection in weakly damped power systems,” IEEE Trans. Smart Grid, vol. 12, pp. 5472–5474, Nov. 2021.

[17] V. Miranda, P. A. Cardoso, R. J. Bessa, and I. Decker, “Through the looking glass: Seeing events in power systems dynamics,” Int. J. of Elect. Power & Energy Sys., vol. 106, pp. 411–419, 2019.

[18] W. Wang, H. Yin, C. Chen, A. Till, W. Yao, X. Deng, and Y. Liu, “Frequency disturbance event detection based on synchrophasors and deep learning,” IEEE Trans. Smart Grid, vol. 11, pp. 3593–3605, July 2020.

[19] Y. Tang and J. Yang, “Dynamic event monitoring using unsupervised feature learning towards smart grid big data,” in Int. Joint Conf. Neural Net., (Anchorage, AK, USA), pp. 1480–1487, May 2017.

[20] B. Wang, Y. Li, and J. Yang, “LSTM-based quick event detection in power systems,” in IEEE Power & Energy Soc. Gen. Meeting, 2020.

[21] J. Jiang, X. Zhao, S. Wallace, E. Cotilla-Sanchez, and R. Bass, “Mining PMU data streams to improve electric power system resilience,” in 4th

IEEE/ACM Int. Conf. Big Data Comp., Appl. & Tech., pp. 95–102, Dec. 2017.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46–61, Mar. 2014.

[23] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the gray wolf optimizer for solving optimal reactive power dispatch problem," Appl. Soft Comp., vol. 32, pp. 286–292, 2015.

[24] H. Verdejo, V. Pino, W. Kliemann, C. Becker, and J. Delpiano, "Implementation of particle swarm optimization (PSO) algorithm for tuning of power system stabilizers in multimachine electric power systems," Energies, vol. 13, no. 8, p. 2093, 2020.

[25] B. Yang, X. Zhang, T. Yu, H. Shu, and Z. Fang, "Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine," Energy conversion and management, vol. 133, pp. 427–443, 2017.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in IEEE Int. Conf. Neural Net., pp. 1942–1948, Nov. 1995.

[27] S. Sengupta, S. Basak, and R. A. Peters, "Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives," Machine Learning & Knowledge Extraction, vol. 1, no. 1, pp. 157–191, 2018.

[28] T. M. Blackwell, J. Kennedy, and R. Poli, "Particle swarm optimization," Swarm Intelligence, vol. 1, no. 1, pp. 33–57, 2007.

[29] M. Juneja and S. Nagar, "Particle swarm optimization algorithm and its parameters: A review," in Int. Conf. Control, Comp., Commu. & Materials, (Allahbad, India), Oct. 2016.

[30] H. Kraiem, F. Aymen, L. Yahya, A. Triviño, M. Alharthi, and S. S. Ghoneim, "A comparison between particle swarm and grey wolf optimization algorithms for improving the battery autonomy in a photovoltaic system," Appl. Sci., vol. 11, no. 16, p. 7732, 2021.

[31] E. Laskari, K. Parsopoulos, and M. Vrahatis, "Particle swarm optimization for integer programming," in Congr. Evol. Comp., (Honolulu, HI, USA), pp. 1582–1587, May 2002.

[32] S. Strasser, R. Goodman, J. Sheppard, and S. Butcher, "A new discrete particle swarm optimization algorithm," in Genetic & Evolutionary Comp. Conf., pp. 53–60, July 2016.

[33] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comp., vol. 1, pp. 67–82, Apr. 1997.

UMAR FAROOQ received his B.S. degree in Electrical Engineering from University of Engineering and Technology Peshawar, Pakistan, in 2013, and M.S. degree in Electrical and Computer Engineering from Portland State University, Oregon, USA, in 2022, where he remained a Fulbright scholar. He is currently working as Senior Engineer Planning at National Grid ESO, UK. He has held various industry positions in Power System Protection, Maintenance, Power Dispatch, and Planning since 2015. His research interests include applications of swarm intelligence optimization and advanced synchrophasor technologies to Power System Planning and Control.

ROBERT B. BASS (M'06) received his Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in 2004. Since 2011, he has been an Associate Professor with the Electrical & Computer Engineering Department at Portland State University, Portland, Oregon. His research addresses engineering challenges to the electric power system that arise from large-scale societal issues such as natural disasters, climate change, and cyber-physical security threats. His research includes service-oriented distributed energy resource aggregation for providing grid services, distributed trust systems for quantifying energy transaction trustworthiness, high-power electric vehicle charging impacts on distribution systems, and cyber-physical system threat detection.