Hindawi
International Transactions on Electrical Energy Systems
Volume 2023, Article ID 6098519, 24 pages
https://doi.org/10.1155/2023/6098519

Research Article

Detection of Data Integrity Attack Using Model and Data-Driven-Based Approach in CPPS

G. Y. Sree Varshini and S. Latha

Thiagarajar College of Engineering, Madurai-625015, India

Correspondence should be addressed to G. Y. Sree Varshini; gysreevarshini90@gmail.com

Received 24 June 2023; Revised 22 October 2023; Accepted 28 October 2023; Published 1 December 2023

Academic Editor: Arvind R. Singh

The cyber-physical power system (CPPS) is a modern infrastructure utilising information and communication technology that has become more vulnerable to cyberattacks in recent years. The attack magnitude injected by the adversary is stealthier and it cannot be detected using conventional bad data detection techniques. Protecting sensitive data from data integrity attacks (DIA) is essential for ensuring system security and reliability. A tragic event will occur if the attack goes unreported. Therefore, DIA detection is highly vital for the operator in the control centre to make important decisions. This paper addresses the attack impact on WAC applications and attack detection using the model-based method and data-driven-based methods. On the basis of the validation of performance indicators, various detection approaches are simulated and compared to determine the best detection strategy. Simulation results show that in the model-based anomaly detection method, the recursive polynomial model estimator (RPME) has better detection performance than the recursive least square estimator (RLSE). The convolutional neural network- (CNN-) based data-driven anomaly detection technique outperforms other machine learning (ML) techniques such as support vector machine (SVM), K-nearest neighbour (KNN), and random forest (RF). On the WSCC 3 machine 9-bus system, the efficacy of the suggested methods is evaluated.

1. Introduction

Cyber-attacks on CPPS have the potential to cause environmental problems by interfering with the monitoring and management of electricity generation and distribution [1, 2]. For instance, a cyberattack on a nuclear power facility would pose safety risks. Utility corporations, governmental organisations, and customers may suffer large financial losses as a result of cyberattacks. Repairing infrastructure, looking into occurrences, and compensating parties who were harmed can be expensive. Professional hackers, antagonistic insiders, and organised criminal gangs are all potential starters of cyberattacks. Energy theft, blackouts, and damage to crucial equipment are just a few of the ways that cyberattacks can have a significant negative impact on the grid. A recently identified virus called Industroyer has the power to control switches and breakers in substations. IEC 60870-5-101, IEC 60870-5-104, and IEC 61850 are only a few of the communication protocols that it might target. To keep the grid stable and keep supply and demand balanced, CPPS relies on real-time data and communication networks. These communications may be interfered with by cyberattacks, which may further jeopardise the grid's dependability and cause voltage instability or cascading failures. For example, cyber-attack events occurred in the western Ukraine power grid in December 2015, a malware attack in the Saudi Arabia oil refinery in 2017, a German power utility was affected by a DOS attack in 2012, on October 2019, a malware attack targeted the Kudankulam nuclear power station. On 12th October 2020, Mumbai was without electricity for more than 12 hours as a result of a cyber attacker inserting malware into the ventilation system. Due to the power outage during the COVID-19 situation, the circumstances grew worse. As a result, rapid and accurate attack detection is crucial for CPPS. One of the most significant cyber disturbances against CPPS wide-area control is a data integrity attack on communication infrastructure. In this study, it is assumed that the adversary interferes with the grid by sending manipulated data through the damping controller's wide-area communication signal. The communication network with a wide-area damping controller is located far away from the control centre and less secured due to weak network protection. So, attackers can simply modify the signal and disrupt the grid.

Farraj et al. [3] proposed an adaptive cyber-enabled parametric feedback linearization control scheme to stabilize power systems during cyber and physical disturbances. Sridhar and Manimaran [4] highlighted the attack impact targeted at voltage control devices (FACTS) using a sensitivity analysis technique. The author does not describe the impact of the data integrity attack on other system parameters such as tie-line power, generator speed deviation, and the active power output of the generator. Sargolzaei et al. [5] proposed a hybrid method of FDI detection using a neural network and a model-based method in a network control system. The developed anomaly detection algorithm, which consists of a NN observer and a Kalman filter-based observer, can simultaneously identify and mitigate the negative effects of system uncertainties and FDI attacks in real-time. In most cases, neural networks do not directly quantify the degree of uncertainty in the prediction. When precise uncertainty quantification is necessary for state estimation activities, this may be a restriction. Chen et al. [6] discussed the findings of experiments on the effects of cyberattacks on two voltage support devices, SVC and STATCOM, in an 8-bus test system, on transient angle and transient voltage stability. The author does not describe attack detection or mitigation techniques. Rawat and Bajracharya [7] described two detection methods for FDI attack, namely, the cosine similarity matching approach and chi-square detector simulated in IEEE 9-bus system. Chi-square detectors might not pick up on minor discrepancies or anomalies in the data, especially if the sample size is small. When identifying greater deviations, it typically performs better. Cosine similarity might not function effectively in high-dimensional spaces with sparse data, where the majority of the dimensions have zero values. The similarity measure may lose some of its significance if the vectors become less useful. Manandhar et al. [8] investigated that the statistically derived FDI attack cannot be detected using the chi-square detector; therefore, proposed an Euclidean detector for such a sophisticated attack. The drawback of the Euclidean detector is that the linear relationship between characteristics is assumed by the Euclidean distance, but this may not always be the case. It is possible that nonlinear relationships are not fully understood, which results in inadequate similarity measures. Basumalli et al. [9] proposed data-driven method of convolutional neural network for packet-data anomaly detection in a PMU-based state estimator with two scenarios namely without and with PMU measurement redundancy. Ashok et al. [10] explained an end-to-end attack-resilient cyber-physical security framework for WAMPAC applications that covers the entire security life cycle, including risk assessment, attack prevention, attack detection, attack mitigation, and attack resilience. Yu et al. [11] proposed the detection of FDI attacks in IEEE 118 and 300 bus systems using deep neural network and wavelet transform technique. Although the wavelet transform can record changes in signal qualities over a range of scales, it might not be the best option for extremely nonstationary data because the underlying statistical values change quickly. Boundary effects from the wavelet transform may appear, especially close to the boundaries of the signal being analysed. These edge effects may make it difficult to spot anomalies close to the signal boundaries. Deng et al. [12] addressed the theoretical foundation for developing preventative countermeasures and an analysis of attack behaviours in distribution systems. Konstantinou and Maniatakos [13] investigated a data-driven algorithm for anomaly detection to address the vulnerability of state estimators to false data injection attacks. The suggested method employs dimensionality reduction on grid measurements, density-based local outlier factor (LOF) analysis, and a feature bagging framework that combines predictions from several LOF outlier detection outputs. Local anomalies are anomalies that are isolated in nearby neighbourhoods, and LOF is specifically made for spotting them. It may not be well suited for detecting global anomalies or anomalies that are distributed across the entire dataset. Within a certain neighbourhood, LOF anticipates that data points should have comparable density. When the density is not constant or when several densities exist within the same dataset, LOF may yield less-than-ideal results. Tan et al. [14] demonstrated crucial insight into the physical impact of false data injections on power grids and provided a framework for analysis in the protection of sensor data linkages tested on a physical 16-bus system testbed and a 37-bus system model. The drawback of state estimate algorithms is that it is based on mathematical models of power system which do not incorporate specific cyber-attack scenarios. Therefore, it is difficult for the state estimate approach to identify novel types of attacks. Li et al. [15] explained the design of a reliable, computationally effective, and high-performance detector by implementing a sequential detector based on the generalised likelihood ratio. Regarding average detection time and robustness to different attack techniques, it performs noticeably better than the current first-order cumulative sum detector. When using online anomaly detection, GLR-based detectors frequently need to store and update sufficient statistics, which can result in significant memory needs. Furthermore, it can cost a lot of computer time to calculate likelihood ratios for high-dimensional data. Liu et al. [16] described integrated cyber-power modelling and simulation testbed, the effects of different cyber events on the physical power system. The author does not describe any detection and countermeasure technique against cyber events. Yin et al. [17] investigated deep learning techniques for modelling intrusion detection systems and developed a DL strategy for intrusion detection utilising RNN in binary class and multiclass classification. Long-term dependencies in data sequences may be challenging for RNNs to detect. The gradients in this situation, which is sometimes referred to as the "vanishing gradient" problem, get progressively smaller over time. The detection of abnormalities that depend on long-term patterns may therefore be challenging for RNNs. Particularly when working with lengthy data sequences, CNNs often demand less memory than RNNs. When resources are scarce, this memory efficiency can be crucial. Shone et al. [18] proposed a novel method of nonsymmetric deep autoencoder- (NDAE-) based deep learning classification model. The benchmark datasets KDD Cup'99 and NSL-KDD were used to test the performance of the proposed classifier, which has been implemented in TensorFlow with GPU support. In contrast to encoding, asymmetric decoding or reconstruction functions are frequently used by nonsymmetric autoencoders. This asymmetry may induce bias during the feature extraction process and may not accurately represent the entire data distribution. He et al. [19] utilised historical measurement data and deep learning techniques to identify the behavioural characteristics of FDI attacks and used the collected characteristics to detect FDI attacks in real time. The performance of the proposed method is validated in IEEE 118 and 300 bus test systems. Since CDBNs are primarily intended for static data, they might not be able to accurately capture temporal dependencies or sequential patterns in time-series data. When dealing with imbalanced datasets, anomaly detection frequently finds that the normal instances much outweigh the anomalies. As a result of CDBNs' potential bias towards the majority class and poor handling of class imbalance, there may be more false negatives for anomalies. Amin et al. [20] proposed a novel method for assessing how a power system will behave dynamically during a cyberattack. Different types of cyber-attacks are reviewed, and the dynamic effects of those attacks are simulated on the Western System Coordinating Council system. The article does not address the impact of the attack on other power system parameters like tie-line power flow and generator speed deviation. To create quick, scalable bad data/event identification for PMU data, the author [21] uses an unsupervised ensemble learning approach. They demonstrated that the ensemble model can be more effectively trained and achieve high accuracy in detecting a variety of errors/events than utilising a single independent detection approach using both simulated and real-world PMU data. Due to restrictions in memory, computing power, and storage, deploying ensemble models may not be possible in resource-constrained environments. When using complicated base models or big datasets, ensemble techniques sometimes involve training many base models, which can be resource and computationally intensive. Bhushan et al. [22] proposed a multilabel classification method based on deep learning techniques to identify coordinated attacks and tested it in IEEE 123-node and 240-node real distribution systems. Jin et al. [23] investigated the economic impact of FDI attacks using dynamics and system topology information on microgrid systems. According to the study [24], robust event-triggered LFC for CPPSs is recommended, along with an additional control loop to defend against DoS attacks. The author discussed a robust event-triggered communication architecture to reduce the additional control loop's reliance on communication resources during denial-of-service attacks. The study then develops a novel switching LFC system model that, unlike current event-triggered LFC systems, incorporates the resilient event generator into an additional control loop when susceptible to DoS attacks. To show the effectiveness of the suggested approach, one-area and two multiarea CPPSs are used. By taking into account potential dynamic behaviours [25], the study suggests a novel three-stage dynamic false data injection attack (DFDIA) model in CPPS. In order to locate the attack and improve the attack vector in order to cooperatively change the metre readings, two variations of restricted differential evolution are described. Then, a countermeasure based on interval state forecasting is suggested to find the established DFDIA. With this detector, the ensemble deep learning-based state forecasting approach is used to establish the variation boundaries of state values.

The CIA trinity, or the three pillars of information security, is data integrity, data confidentiality, and data availability. Specific forms of cyberattacks can target any one of these aspects:

(i) Attacks on data integrity attempts to undermine the accuracy, dependability, and reliability of data. An attacker may change, remove, or inject misleading information into a system, causing inaccurate decisions or actions based on the modified data.

(ii) Attacks on data confidentiality are attempts to get unauthorised access to private or sensitive data. Attackers try to access data they are not supposed to read or obtain, which could result in data breaches.

(iii) An attack on data availability aims to stop or prevent access to data or information. Attackers want to prevent authorised individuals from accessing data, which could have an impact on operations, finances, or safety.

Attacks on data integrity in CPPS are a severe problem since they may result in system downtime, equipment damage, and financial losses. Encryption, intrusion detection systems, and secure communication protocols are all crucial components of effective cybersecurity measures to protect against these attacks. Increasing the efficiency and security of the power grid through real-time monitoring, control, and decision-making is a primary goal of the CPPS. The CPPS needs to be able to autonomously recognise faults, respond to them, reorganise itself, and restart power distribution in the case of disruptions or outages in order to fulfil its goal of self-healing. Therefore, detecting anomalies is a crucial step in making judgements about how to restore the system. The strengths of both model-based and learning-based methods are combined in a hybrid approach to anomaly detection in the grid, which increases the effectiveness overall. The advantages of statistical modelling, domain knowledge, machine learning, and deep learning methods are all utilised in this strategy. Past research work involves the detection of the cyber-attack using the Kalman filter, cosine similarity approach, chi-square detector and machine learning techniques. The novelty of this paper is the recursive polynomial model estimator for online detection of different types of data integrity attacks, and its performance is validated based on estimation error. Utilising mathematical and statistical methods, a recursive estimator is used to continuously monitor system behaviour, spot abnormalities, and adjust to shifting attack patterns. On the basis of new data, statistical models are updated and improved using recursive estimators. Its capacity to adjust to changing attack methodologies is one of its main benefits. The estimator integrates fresh knowledge about what defines "normal" behaviour when new data are gathered and the model is updated, making it more successful at spotting new attack patterns. Thus, the recursive estimators are appropriate for real-time applications. In control systems, where accurate and timely information is critical, this capability is essential. Therefore, two types of recursive estimators, namely, RLSE and RPME, are used in model-based attack detection in CPPS. The majority of attack detection research is conducted on AGC and LFC applications [26] but lacks analysis of DIA in FACTS-based WAC applications [27]. Attackers can implement the DIA on any application [28] in CPPS such as economic dispatch, state estimation, SCADA measurements, PMU, voltage control devices, control input, and inertia control. In this article, it is considered that the attacker implements an integrity attack for signal manipulation on the FACTS-based damping controller. Therefore, the research work focuses on the analysis of DIA on STATCOM-based wide-area damping controller and its detection. The outline of the research work is shown in Figure 1 and the schematic representation of DIA is shown in Figure 2.

The contribution of this paper is summarized as follows:

(i) The WSCC system is modelled as CPPS with hybrid simulation using Simulink and SimEvents.

(ii) The system is simulated with different types of false data injection attacks, namely, step, ramp, impulse, random attack, and its wide-area attack impact are addressed.

(iii) In the model-based method, different data integrity attacks (step, random, impulse, and ramp) are detected using RPME and RLSE and their prediction accuracy is validated based on estimation error. In addition, the attacks are detected in the frequency domain by estimating power spectral density estimation using the Welch method.

(iv) In the learning-based approach, the effectiveness of several attack detection strategies including CNN, KNN, SVM, and RF are assessed through comparison of performance metrics (precision, accuracy, and $F1$ score).

Simulation results using MATLAB/Simulink are compared which shows that RPME performs better in detection than RLSE in the model-based method. In the learning-based method, CNN (deep learning technique) outperform the machine learning technique.

2. Mathematical Modelling of CPPS

Consider the WSCC system as CPPS with a STATCOM device. The wide-area control signal (speed deviation) is obtained through a communication channel and modulated by a FACTS device for control action to take place.

The mathematical equation of the physical system (power system) can be represented as

$$\begin{aligned} y = h(x, u), \dot{x} &\in F_P(x, u),\\ (x, u) \in \mathrm{C_P} &\subset \mathbb{Z}^{n_P} \times \mathbb{Z}^{m_P}, \end{aligned} \tag{1}$$

where $\mathbb{Z}^{n_P}$ is the Euclidean space for state space. $u \in \mathbb{Z}^{m_P}$ is the input signal for the physical system and $y \in \mathbb{Z}^{r_P}$ is the output of the physical system. h is the output function. x is the state of the physical system.

The mathematical equation of the cyber system (communication network) can be represented as

$$\begin{aligned} \zeta = K(\eta, \gamma), \eta^{+} &\in G_C(\eta, \gamma),\\ (\eta, \gamma) \in D_C &\subset \Upsilon \times \gamma, \end{aligned} \tag{2}$$

$\eta \in \Upsilon$ is the state of the cyber system. $\mathbb{Z}^{n_C}$ is the Euclidean space for the state space, $\gamma \in \mathsf{V} \subset \mathbb{Z}^{m_C}$ is the input signal for the cyber system, $\zeta \in \mathbb{Z}^{r_C}$ is the cyber system output defined by the function K. K is the function of input γ and the state η.

The information transferred over the communication network at time instant, $\{\tau_i\}_{i=1}^{p^*}, p^* \in \mathbb{N} \cup \{\infty\}$ satisfying

$$T_n^{*\min} \leq \tau_{i+1} - \tau_i \leq T_n^{*\max} \forall\, i \{1, 2, \ldots \mathrm{p}^* - 1\}, \tag{3}$$

$T_{\mathrm{n}}^{*\min}$ and $T_{\mathrm{n}}^{*\max}$ are constants satisfying the following constraints:

$$\begin{aligned} T_n^{*\min}, T_n^{*\max} &\in [0, \infty],\\ T_n^{*\min} &\leq T_n^{*\max}, \end{aligned} \tag{4}$$

where p^* is the no. of. transmission events. $T_n^{*\min}$ and $T_n^{*\max}$ is the minimum and maximum time between the transmission events.

The mathematical model of the communication network is

$$\begin{aligned} &\dot{\tau}_n = 1,\\ &\dot{m}_n = 0 \text{ when } \tau_n \in \left[0, T_n^{*\max}\right],\\ &\tau_n^{+} \in \left[T_n^{*\min}, T_n^{*\max}\right],\\ &m_n^{+} = v_n \text{when } \tau_n \leq 0,\\ &\dot{\lambda} \in F_I(\lambda, q) \text{ when } (\lambda, q) \in C_I,\\ &\lambda^{+} \in G_I(\lambda, q) \text{ when } (\lambda, q) \in D_I,\\ &\beta = \varphi(\lambda), \end{aligned} \tag{5}$$

where λ is the state, q is the input signal, β is the output signal, F_I is thecontinuous dynamics on C_I, and G_I is the discrete dynamics on D_I.

3. Communication Network Modeling Using SimEvents

SimEvents blocks [29] are added to the Simulink model to create communication network between the sensor and the

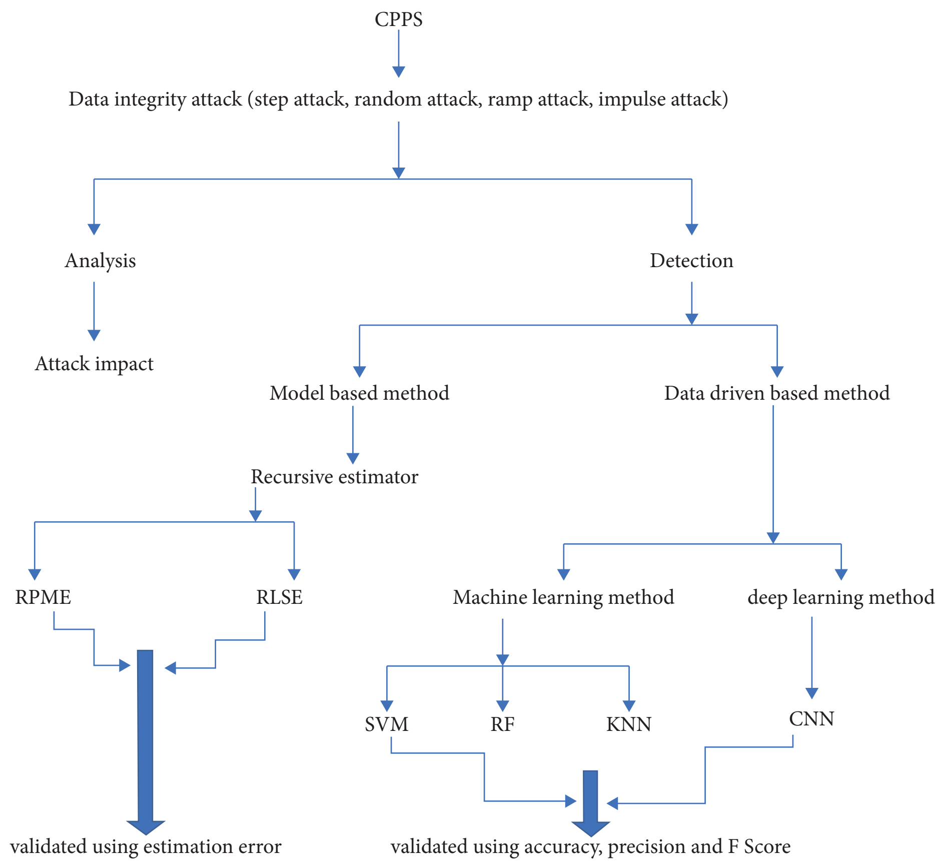

CPPS
Data integrity attack (step attack, random attack, ramp attack, impulse attack)
Analysis
Detection
Attack impact
Model based method
Data driven based method
Recursive estimator
RPME
RLSE
Machine learning method
deep learning method
SVM
RF
KNN
CNN
validated using estimation error
validated using accuracy, precision and F Score

Figure 1: Overview of the proposed work.

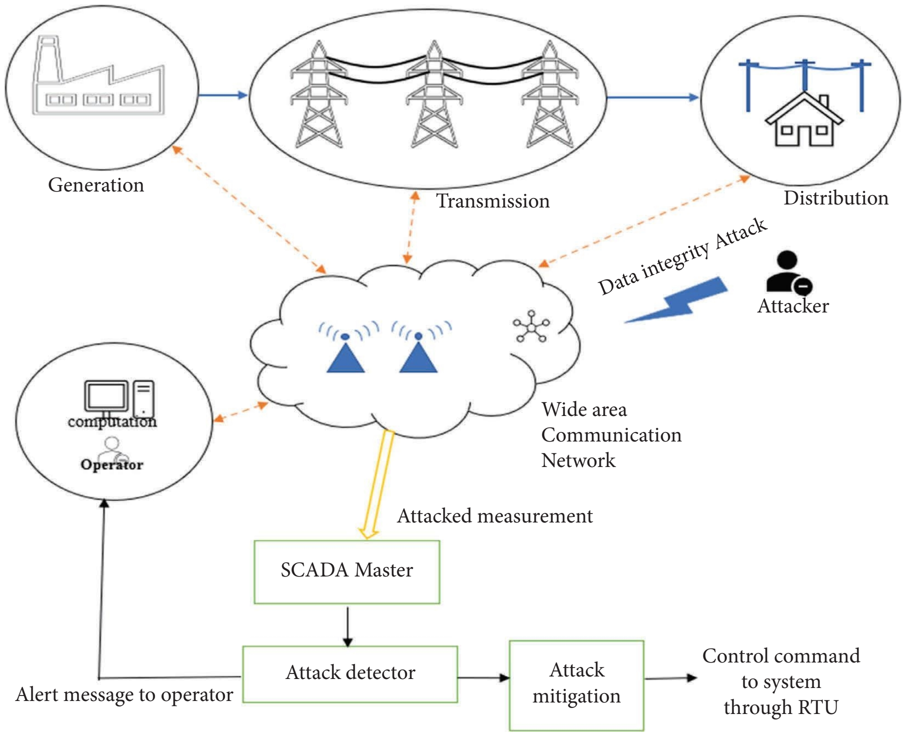


Figure 2: An illustration of a data integrity attack.

controller. Thus, the hybrid model is used to convert continuous signal to discrete event signal and also uses packet communication using entities. Therefore, it involves both time-based and discrete event-based modelling. From Figure 3, it is inferred that the rotor speed deviation signal from generator 1 and 2 (RSD1 and RSD2) are time domain signals which are converted to discrete event signal using a time-to-event signal block. To generate random packets or entities, time-based entity generator is used with exponential distribution of mean = 100. The attributes are attached to the packets using a set attribute block. FIFO queues are set to fix capacity of 25 each. Entities from different paths are merged using a path combiner, and out switch block enables routing of entities to final destination. The server serves the entities for a period of time, where it is converted again to a time domain signal in order to feed into the damping controller. Figures 4(a) and 4(b) represent received entities at the destination of communication network and time-stamped entities, respectively.

4. Model-Based Anomaly Detection

Consider the CPPS with following dynamics:

$$\dot{x}(t) = \mathrm{Ax}(t) + \mathrm{Bu}(t) + w(t), \quad (6)$$

$$y(t) = \mathrm{Cx}(t), \quad (7)$$

where $x(t) \in \mathbb{Z}^n$ is the state vector variable, $u(t) \in \mathbb{Z}^m$ is the control input, $w(t) \in \mathbb{Z}^p$ is the disturbance vector. A, B, and C are the system matrix, input matrix and output matrix of appropriate dimensions. Control centre collects the measurement from PMU across various location of system to compute estimate of unknown state variable x.The obtained measurements be $M = (m_1, m_2 \ldots m_k)^T$ and the state variables are $x = (x_1 + x_2 \ldots x_l)^T$. Measurement errors with gaussian noise is $\varepsilon = (\varepsilon_1 + \varepsilon_2 \ldots \varepsilon_k)^T$.

The measurement model is given by

$$M = \mathrm{Hx} + \varepsilon. \quad (8)$$

H denotes the Jacobian matrix of dimension $k \times l$ which shows the relationship between measurement set and the state vector.

The estimated state vector is given by

$$\hat{x} = \mathrm{MRH}^T \left(H^T \mathrm{RH}\right)^{-1}. \quad (9)$$

R is the diagonal matrix which is the reciprocal of variance σ.

Adding an attack vector to the measurement entails inserting malicious data.

The compromised measurement of damping controller signal is

$$M_{\mathrm{comp}} = M + a, \quad (10)$$

$$a = \mathrm{Hc}, \quad (11)$$

where a is the attack vector $a = (a_1 + a_2 \ldots a_q)^T$, c is the sparse matrix, and M_{comp} is the compromised measurement. The attack vector is added to compromised measurement as follows:

$$\widehat{M} = H(x + c) + \varepsilon, \quad (12)$$

$$v_i = \begin{cases} 0, & \text{(no attack)}, \\ 1, & \text{(compromised measurement)}, \end{cases} \quad (13)$$

v_i is the label set of measurements to differentiate normal and attacked samples.

The STATCOM device represented by a discrete-time auto regressive polynomial model with the following form as

$$A\left(q^{-1}\right) y(t) = \sum_{i=1}^{\mathrm{mu}} B_i q^{-1} u_i\left(t - \mathrm{mk}_i\right) + e(t), \quad (14)$$

where

$$A\left(q^{-1}\right) = 1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3} \ldots + a_{n_a} q^{-\mathrm{na}}, \quad (15)$$

$$B\left(q^{-1}\right) = b_0 + b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3} \ldots + b_{n_b} q^{-\mathrm{nb}}. \quad (16)$$

The generic version of the recursive identification algorithm is provided by

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + K(t)[y(t) - \widehat{y}(t)]. \quad (17)$$

Estimated parameter $\widehat{\theta}(t)$ recursively computed by

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + K(t)\left[y(t) - \widehat{\theta}^T(t-1)\psi(t)\right], \quad (18)$$

where $\psi(t)$ can be computed as

$$\psi(t) = [y(t-1)\ldots y(t-n)u(t-1)\ldots u(t-n)]^T. \quad (19)$$

Online estimation is the best method for estimating minor variations in system's parameter values at a predetermined operating point. The recursive technique is generally used to accomplish online parameter estimation which employ the current estimation and measurements to estimate the parameter values for a given time step. It is effective in terms of memory utilisation and requires less computation. Recursive estimators continuously update their estimates as new data become available. As a result, the estimations are always based on the most recent and relevant data. This reactivity to incoming data may lead to more accurate estimations, particularly in dynamic environments where conditions change over time. They are suitable for embedded and online applications because of their efficiency. Two recursive estimators are used for DIA detection, namely, RPME and RLSE. The recursive least squares estimator (RLSE) [30] recursively determines the coefficients that minimise a weighted linear least squares cost function pertaining to the input signals. The Simulink model of DIA detection is shown in Figure 5. System identification technique and estimator are used to determine the parameters used in this model. Based on estimation error, the performance of both recursive estimators are discussed in results and discussion section. The block diagram for detection of DIA using recursive estimators is shown in Figure 6. The flowchart for RPME and RLSE is shown in Figures 7 and 8, respectively. Parameters used in RLSE and RPME are shown in Table 1 (Algorithm 1) Algorithm 1 describes the detection of DIA using model based method.

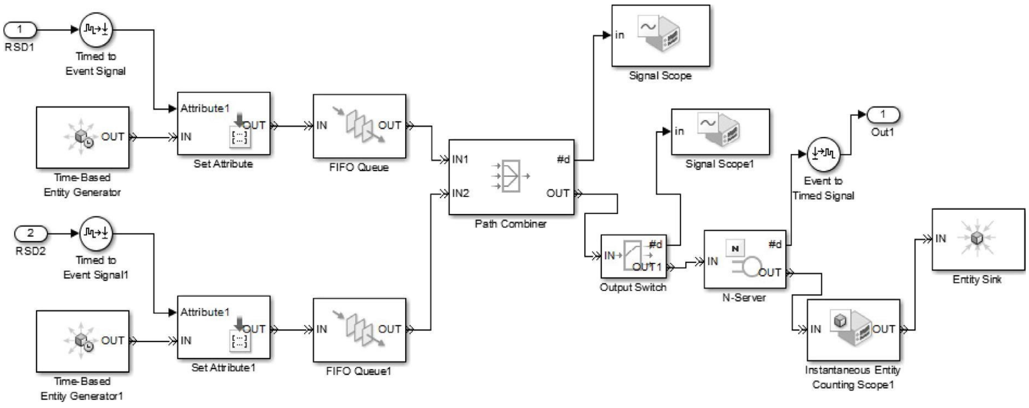
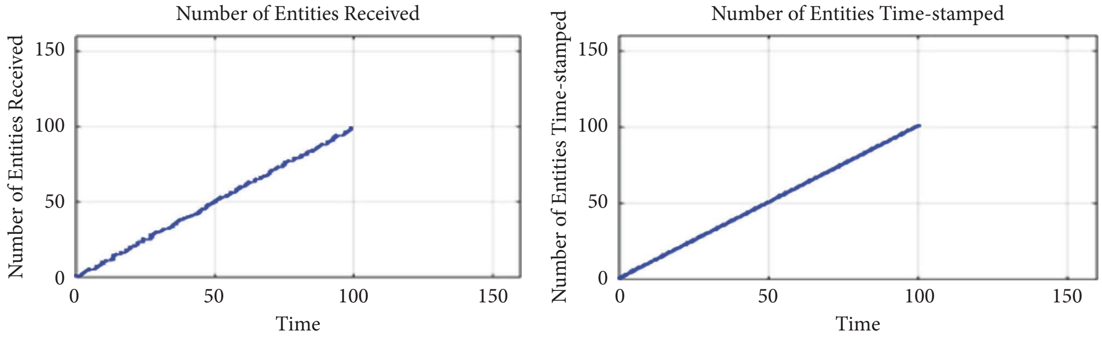


Figure 3: Discrete event based modelling of communication network.



(a)

(b)

Figure 4: (a) Number of entities received at the destination; (b) number of entities which are time stamped.

4.1. RLSE. The RLSE incrementally adjusts parameter estimations in response to new data points. Between the output that was expected and what was actually produced, the method seeks to minimise the sum of squared errors. RLS functions naturally as a noise filter by combining data from numerous time steps, which helps lessen the impact of measurement noise and enhances the accuracy of parameter estimates. RLSE can be used in the context of anomaly detection to adaptively update models of typical behaviour and detect deviations from these models. An attack or other abnormality in the system may be indicated by a notable departure from the parameter estimations or by an increase in the error variance.

Steps involved in RLSE for attack detection:

(1) Set up the settings for the RLSE algorithm's initialization, such as h, $\Theta(0)$, $p_0(0)$ and the forgetting factor $\lambda = 0.001$ (a number between 0 and 1 that regulates the impact of past data).

(2) Calculate the model parameters of RLS technique. It continually modifies the model's parameters in response to new data.

$$\text{Data input matrix}\ \phi(t) = [E(t-1)E(t-2)I(t)I(t-1)I(t-2)]^T,$$
$$\text{gain}\ K = \frac{p_0(t)\phi(t)}{\lambda + \phi(t)^T p_0(t)\phi(t)}. \quad (20)$$

(3) Determine the differences between the observed data and the predicted values from the estimated model. A significant deviation from the expected behaviour indicates an anomaly. For each iteration update the estimation parameter $\Theta(t)$

$$\begin{aligned}\varepsilon(t) &= u(t) - \Theta(t-1)^T\phi(t),\\ \Theta(t) &= \Theta(t-1) + K(t)\varepsilon(t).\end{aligned} \quad (21)$$

(4) Update covariance matrix and forgetting factor using the below equation:

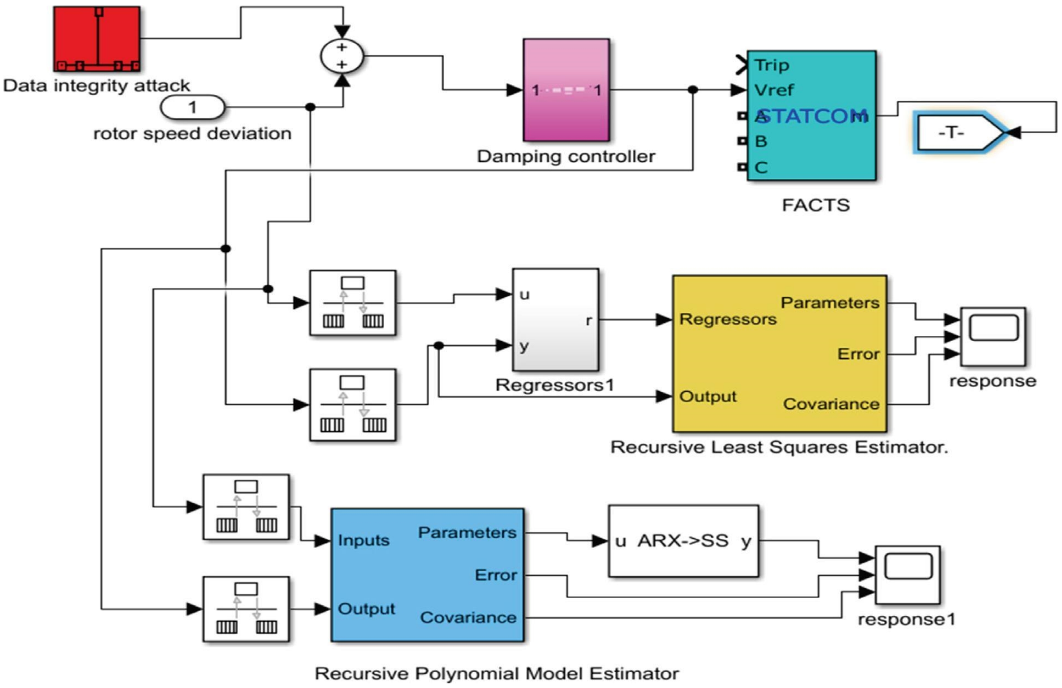


Figure 5: Simulink model of DIA detection using RPME and RLSE in the WSCC system.

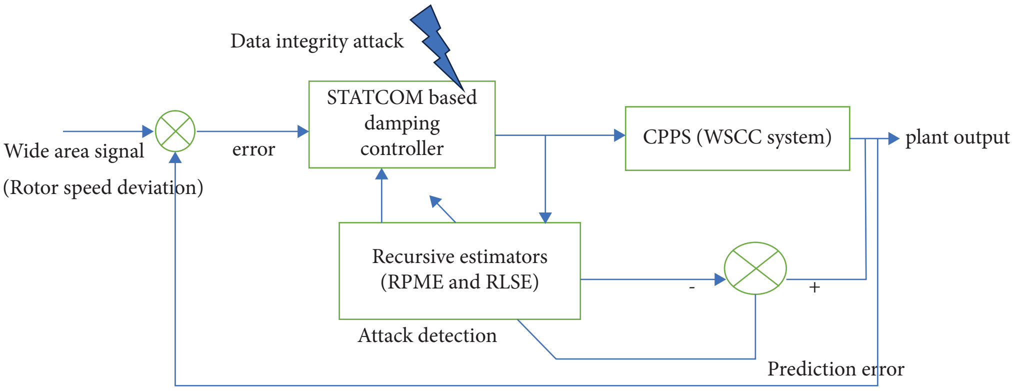


Figure 6: Block diagram of DIA detection using recursive estimators in CPPS.

$$p_0(t+1) = \frac{1}{\lambda}\left[1 - K(t)\phi(t)^T\right]p(t),$$
$$\lambda(t) = \lambda_{\min} + (1 - \lambda_{\min})\lambda^{e(t)}. \tag{22}$$

(5) Continually include new data to update the model's parameters. This constant learning process aids the model's adaptation to shifting attack patterns as well as changing system behaviour.

4.2. RPME. A recursive estimator used to estimate the coefficients of a polynomial model that describes a system is known as a recursive polynomial model estimator. The estimator can modify the polynomial model to better match the current system behaviour when new data are gathered. This is very useful when working with dynamic systems or nonstationary data. When compared to completely reestimating the model whenever new data are received, recursive polynomial model estimators update parameters incrementally, which can be less computationally challenging. Dealing with enormous datasets or having few computational resources makes this extremely helpful. An attack or a shift in the behaviour of the system could be indicated by a large departure in the calculated coefficients from the predicted polynomial behaviour.

Steps involved in RPME for attack detection:

(1) Set up the settings for a polynomial model with forgetting factor $\lambda = 0.01$, h, $\Theta(0)$, $p_0(0)$, and polynomial coefficients.

(2) For the polynomial model $y = a_0 + a_1 x + a_2 x^2 \ldots a_n x^n + \varepsilon$ use a recursive technique to calculate the parameters of polynomial coefficients as follows:

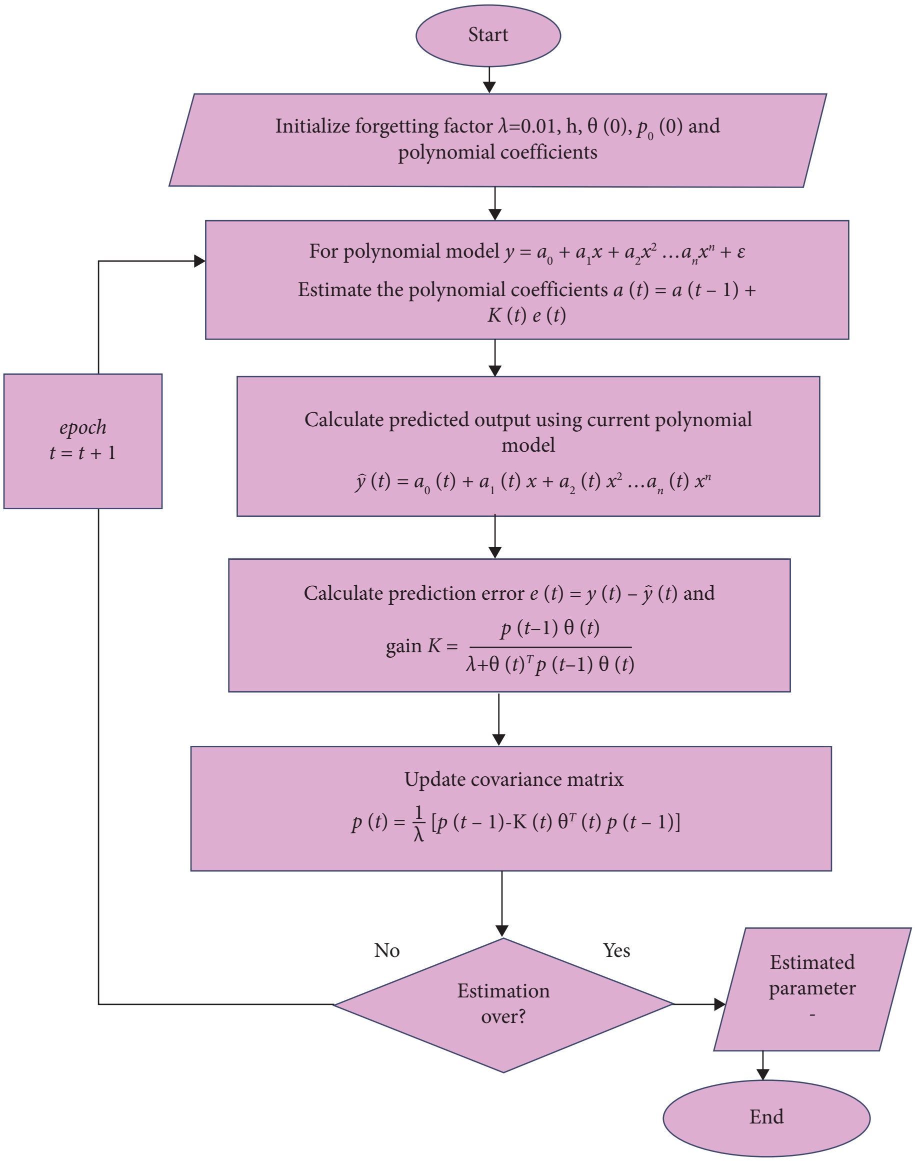

Start
Initialize forgetting factor λ=0.01, h, θ (0), p_0 (0) and polynomial coefficients
For polynomial model $y = a_0 + a_1x + a_2x^2 \ldots a_nx^n + \varepsilon$
Estimate the polynomial coefficients $a(t) = a(t-1) + K(t)e(t)$
epoch
$t = t + 1$
Calculate predicted output using current polynomial model
$\hat{y}(t) = a_0(t) + a_1(t)x + a_2(t)x^2 \ldots a_n(t)x^n$
Calculate prediction error $e(t) = y(t) - \hat{y}(t)$ and
gain $K = \frac{p(t-1)\theta(t)}{\lambda + \theta(t)^T p(t-1)\theta(t)}$
Update covariance matrix
$p(t) = \frac{1}{\lambda}[p(t-1)\text{-K}(t)\theta^T(t)p(t-1)]$
No
Yes
Estimation over?
Estimated parameter
-
End

Figure 7: Flowchart for RPME.

$$a(t) = a(t-1) + K(t)e(t),$$
$$K = \frac{p(t-1)\Theta(t)}{\lambda + \Theta(t)^T p(t-1)\Theta(t)}. \tag{23}$$

(3) Update the model parameters to take the most recent observations into account for each new data point. Recursive updating aids in adjusting to evolving behaviour over time.

(4) Determine the differences between the calculated polynomial model predictions and the observed data and update the covariance matrix as follows. As it shows a significant departure from the usual behaviour, a higher divergence could be an anomaly.

$$e(t) = y(t) - \hat{y}(t),$$
$$p(t) = \frac{1}{\lambda}\left[p(t-1) - K(t)\Theta^T(t)p(t-1)\right]. \tag{24}$$

(5) Using past data, periodically update the model's inputs. This ongoing learning process guarantees that the model will stay flexible to adapt to shifting system behaviour.

5. Data-Driven-Based Anomaly Detection

All methods and algorithms that enable computers to automatically learn from massive datasets by using mathematical models are collectively referred to as machine learning (ML), a subset of artificial intelligence. In recent years, there has been a rise in trends and interest in using machine learning and deep learning-based anomaly detection to address cyber-attack issues. In order to increase the capability of intrusion detection systems to identify malicious attacks, numerous ML- and DL-based algorithms have been published by researchers. For predicting and controlling, AI approaches can provide swift and accurate

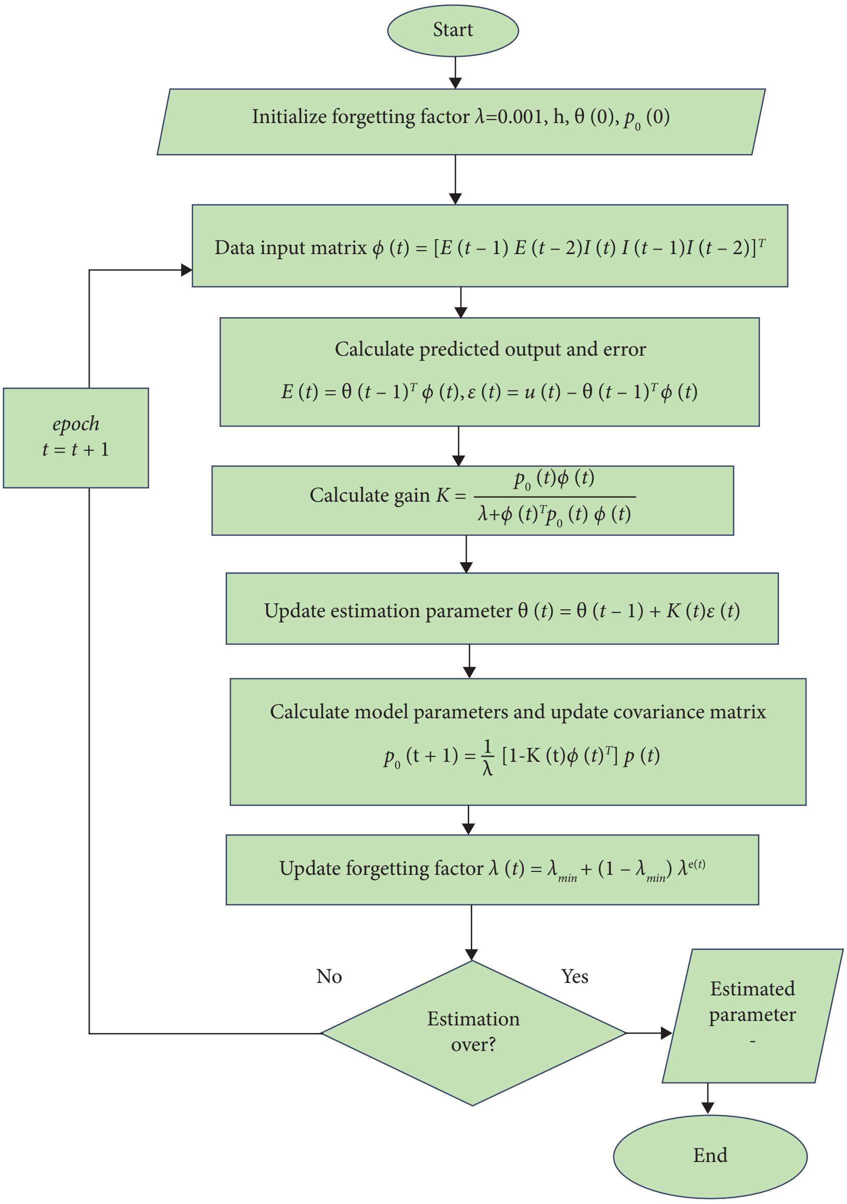

Start
Initialize forgetting factor λ=0.001, h, θ (0), p_0 (0)
Data input matrix $\phi (t) = [E (t-1)\ E (t-2) I (t)\ I (t-1) I (t-2)]^T$
Calculate predicted output and error
$E (t) = \theta (t-1)^T \phi (t), \varepsilon (t) = u (t) - \theta (t-1)^T \phi (t)$
Calculate gain $K = \frac{p_0 (t)\phi (t)}{\lambda + \phi (t)^T p_0 (t)\ \phi (t)}$
Update estimation parameter $\theta (t) = \theta (t-1) + K (t)\varepsilon (t)$
Calculate model parameters and update covariance matrix
$p_0 (\mathrm{t}+1) = \frac{1}{\lambda} [1\text{-K} (\mathrm{t})\phi (t)^T]\ p (t)$
Update forgetting factor $\lambda (t) = \lambda_{min} + (1-\lambda_{min})\ \lambda^{e(t)}$
epoch
$t = t + 1$
No
Yes
Estimation over?
Estimated parameter
-
End

Figure 8: Flowchart for RLSE.

Table 1: Parameters used in recursive estimators.

Parameters of RPME	Parameters of RLSE
(i) Model structure: ARX	(i) No of parameters = 2
(ii) No of parameters in $A(q) = 3$	(ii) Parameter covariance matrix = 1
(iii) No of parameters in $B(q) = 3$	(iii) Sample time = 0.5
(iv) Parameter covariance matrix = 1	(iv) Estimation method: forgetting factor
(v) Sample time = 0.5	(v) Forgetting factor: 0.001
(vi) Estimation method: forgetting factor	
(vii) Forgetting factor: 0.01	

information-driven answers [19, 31–33]. Machine learning is an information analysis technique that leverages learner interaction to direct a computer to carry out certain tasks. Let us discuss some of the ML- and DL-based methods. Figure 9 represents detection of DIA using the data-driven method.

5.1. SVM. In a S-dimensional subspace, SVM attempts to define a hyperplane that divides data points. A low-dimensional input vector is first mapped using the kernel function into a high-dimensional feature space. Using the support vectors, an ideal maximum marginal hyperplane is produced that serves as a decision boundary [34]. Support

```
Data: $A, B, C, D$, are known
Begin: Initialize parameters $\widehat{x}(0), \varepsilon(0), A(q), B(q)$
for $t$ do
    Calculate $\widehat{x}(t)$ using (6) and (7);
    Compute $M$ using (8);
    Update $\widehat{x}(t)$ using (9);
    Compute $\widehat{M}(t)$ using (12);
    Compute $A(q^{-1}), B(q^{-1})$ using (15), (16);
    Update $y(t)$ using (14);
    compute $\widehat{\theta}(t), \psi(t)$ using (17) and (19);
    Estimate the DIA attack using (18);
    end;
```

Algorithm 1: For detection of DIA.

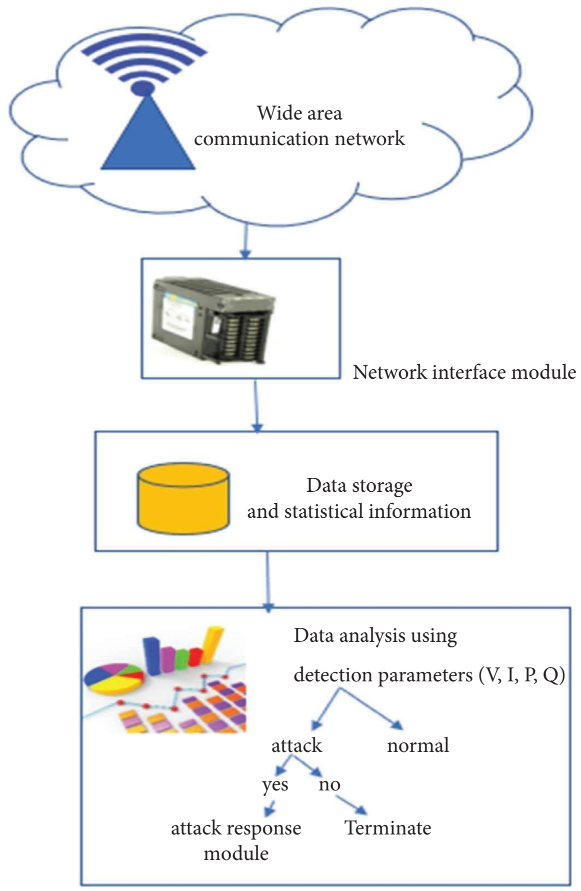


Figure 9: Illustration of DIA detection using learning-based method.

vector machines (SVMs) can improve anomaly detection accuracy by effectively separating between normal and anomalous cases using an optimal decision boundary. It is crucial to choose the correct kernel function and hyperparameters in order to achieve the best performance. SVMs seek to locate a hyperplane that maximises the margin between various classes. There may be a larger difference between average occurrences and anomalies because anomalies are frequently observed in less populated areas. The kernel function and hyperparameters (such as the regularisation parameter "C") that are chosen with attention can have a significant impact on SVM's performance. By adjusting these hyperparameters, the model's ability to accurately detect anomalies can be improved. The SVM is hence better equipped to distinguish between normal and anomalous samples.

In this classifier the regularisation parameter is 1 with degree of 2 is used.

The mathematical formulation of the classification approach is

$$f(u, v) = \sum_{i=1}^{S} v_i x_i(u) + \mathrm{K}, \tag{25}$$

v_i defines the prediction parameters in S dimension space. The data distribution and classification variables together determine K.

$$k(r, z) = \exp\left(-\Theta\|r - z\|^2\right). \tag{26}$$

In order to give identical data points in a dataset, this function is used in conjunction with SVM. The flowchart for SVM is shown in Figure 10.

5.2. KNN.

KNN, one of the most straightforward supervised machine learning algorithms, uses the idea of "feature similarity" to ascertain the class of a specific data sample [35]. A quick method of classifying new points is to categorise query points according to how close they are to points in a training dataset. By calculating how far away a sample is from its neighbours, it can identify that sample's identity. The sample case might not be correctly classified if the k value is chosen with a very wide range. Therefore, k value should be chosen appropriately. The selection of the "k" parameter can have a significant impact on the accuracy of the KNN method. The best "k" parameter depends on the type of dataset and the distribution of classes. When "k" is low (for instance, 1 or 3), the prediction is impacted by the proximate characteristics of the nearest neighbours. Because of this, the decision boundary could become less steady and more "jumpy," making it more susceptible to noise or outliers. Larger "k" values typically reduce the risk of overfitting but may result in poorer accuracy. Finding the

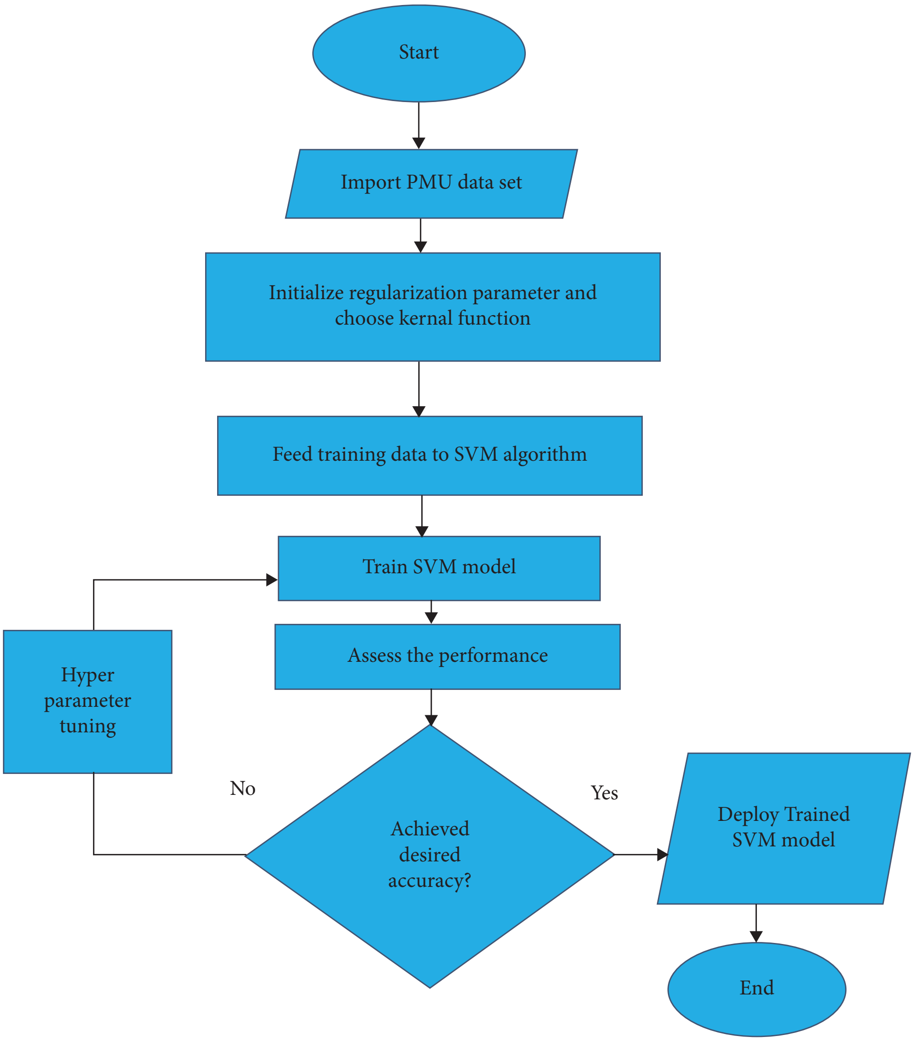

Start
Import PMU data set
Initialize regularization parameter and choose kernal function
Feed training data to SVM algorithm
Train SVM model
Assess the performance
Hyper parameter tuning
No
Achieved desired accuracy?
Yes
Deploy Trained SVM model
End

Figure 10: Flowchart for SVM.

best "k" value that balances precision and generalisation for particular problem frequently needs testing and validation. Therefore, the different values for "k" are implemented to find the optimal value. Then, the KNN model is trained using the optimal "k" value on the full training dataset once it has been determined through analysis. The flowchart for KNN is shown in Figure 11.

An attack's membership in the class with the most members among its k nearest neighbours is determined by the majority vote of its neighbours. In this classifier, a 5 number of neighbours with cosine metric are used. The classification technique calculates the distance between each sample in a dataset and updates the data using Euclidean distance. For n number of training dataset with m attributes $\{x_{i1}, x_{i2}, \ldots x_{\text{im}}\}$ and testing dataset $\{y_1, y_2, \ldots y_m\}$ with label l_i, where $i \in [1, n]$. The Euclidean distance between the training and testing dataset is computed as

$$d(x, y) = \sqrt{\sum_{j=1}^{n} \left(x_j - y_j\right)^2}. \tag{27}$$

5.3. RF. In an effort to rectify the overfitting of a single decision tree, the random forests technique combines several decision tree classifier models. The random forest method constructs a decision tree from a sample of data, forecasts each one, and then votes on the best outcome. A random forest R is composed of the k decision tree model. Each decision tree makes a different prediction for the input testing data before a simple majority vote is utilised to determine the outcome. In order to categorise a new object x, RF aggregates the votes from all of the decision trees (k) in the forest (R). The class that frequently appears in the random forest and receives the majority of votes is the projected class of x determined by the tree. According to the definition of the simple majority voting formula

$$R(X) = \operatorname{argmax} \sum_{i=1}^{k} I\left(r_i(X) = Y\right), \tag{28}$$

where $X = (x_1, x_2, \ldots x_m)^T$, X is the dataset with m number of samples. It is based on the idea of ensemble learning, which is the act of integrating various classifiers to solve

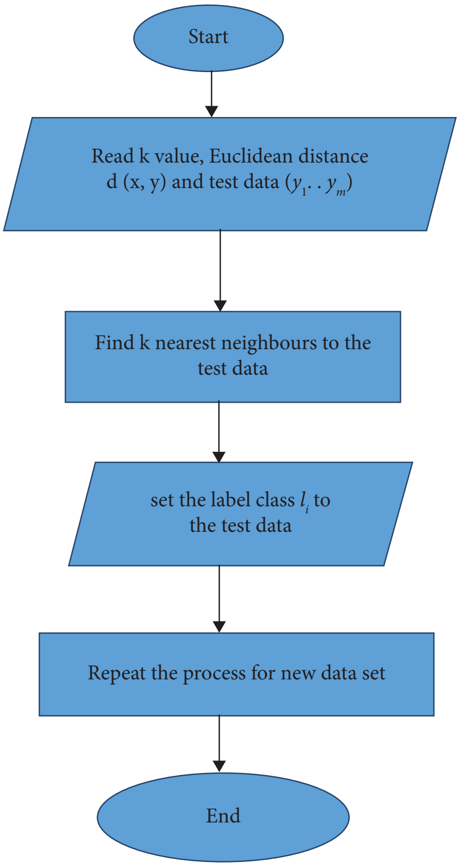

Start
Read k value, Euclidean distance d (x, y) and test data ($y_1 . . y_m$)
Find k nearest neighbours to the test data
set the label class l_i to the test data
Repeat the process for new data set
End

Figure 11: Flowchart for KNN.

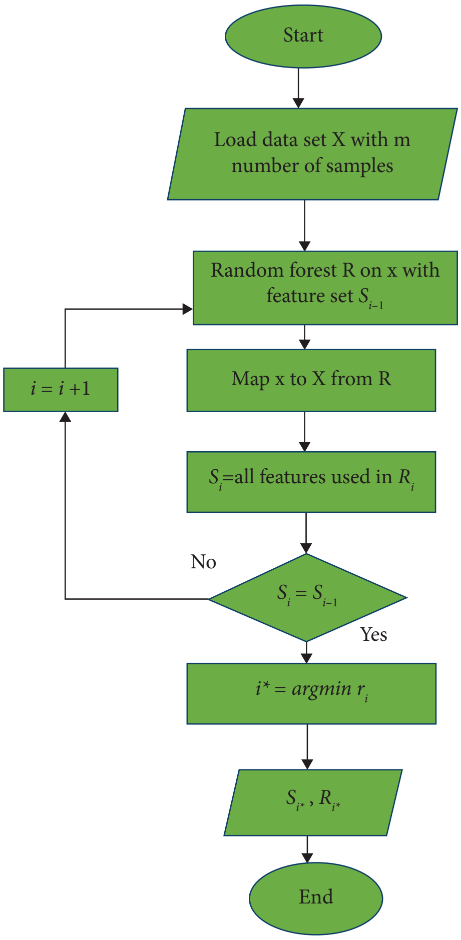

Start
Load data set X with m number of samples
Random forest R on x with feature set S_{i-1}
$i = i +1$
Map x to X from R
S_i=all features used in R_i
No
$S_i = S_{i-1}$
Yes
$i^* = argmin\ r_i$
S_{i^*}, R_{i^*}
End

Figure 12: Flowchart for RF.

a complex problem and enhance the performance of the model [36, 37]. In contrast to individual tree predictions, the prediction of RF is obtained by majority vote. By averaging the results from numerous DT classifier fits on different subsamples of the dataset, the prediction accuracy is increased. When splitting a node, it looks for the best feature from a random subset of features rather than the most essential feature. In this classifier, a 20 number of trees with maximum depth of 15 are considered. The flowchart for RF is shown in Figure 12.

5.4. CNN. CNN structure composed of input layer, a stack of convolutional and pooling layers, and then a fully connected layer and a SoftMax classifier in the classification layer [38–40]. Time series have a strong 1-D locality that can be retrieved by convolutions, making CNN an effective tool for processing time series data. Depending on the data's input dimension and processing power, different numbers of filters and convolution layers can be used. The 1-D data are used by each neuron in the fully connected layer to generate its own score as determined by the subsequent equation:

$$y_i = \sum_{j=1}^{m} w_{i,j} x_j + b_s, \tag{29}$$

where y_i is the fully-connected layer output in the i^{th} neuron, m is the 1-D input data length (x), $w_{i,j}$ is the weight of neuron between j^{th} input value i^{th} neuron, and b_s is the bias.After the computing the above value of y_i, it will use an activation function to send the value to the associated units in the higher layer to see how much it affects the prediction of the following step. The activation function is provided as follows:

$$\begin{aligned} o_i &= f\left(y_i\right) \\ &= \max\ \left(0, y_i\right). \end{aligned} \tag{30}$$

The output of activation function $f\left(y_i\right)$ is o_i.Rectified Linear Unit (ReLU), which only activates positive values, is used as the activation function which prevents overfitting problem. By linking each neuron to its neighbour neurons, CNN overcomes the drawbacks of conventional neural networks. After conversion of one-dimensional time series data to two-dimensional data, the convolution process will then be performed with the input 2-D data using a filter with the same size receptive field. Features are extracted from the input by a 2-D convolution layer. The greatest value of the field covered by the pooling filter will be selected by the pooling layer.

$$b = \begin{bmatrix} b_{1,1} & \cdots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \cdots & b_{m,n} \end{bmatrix}. \quad (31)$$

After max pooling process, the feature map b becomes

$$\hat{b} = \max(b). \quad (32)$$

The choice of filter size depends on the dataset's features and the problem to be solved. Lower-level features can only be captured by smaller filters, but larger filters can capture higher-level temporal structures. Larger filter sizes are preferred in deeper layers to capture longer-term dependencies. Filter size is one aspect of the architecture that affects how well the model performs. Additional factors like network depth, pooling methods, and activation functions have impact on how well CNN performs in terms of accuracy. This combination of factors typically results in a CNN design that is effective at handling high-dimensional data. Therefore, experimenting with various filter sizes to find which choice is appropriate for a specific dataset is important in a CNN classifier. In this classifier, a 10 number of filters with kernel size of 2 are considered. The ReLU activation function is used and the learning rate is 0.0100. The flowchart for CNN is shown in Figure 13.

5.4.1. Evaluation Metrics. The following metrics used to assess the performance of classifier [31].

5.4.2. Precision. It measures the proportion of attacks that were accurately predicted to all the samples that were attacked.

$$\text{Precision, } P = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False positive (FP)}}. \quad (33)$$

5.4.3. Accuracy. It is the proportion of occurrences that were correctly categorised to all of the instances.

$$\text{Accuracy} = \frac{\text{number of correct predicted data}}{\text{number of testing data}}. \quad (34)$$

5.4.4. Recall. In a dataset, recall quantifies the model's capacity to accurately identify every pertinent event or true positive. It measures how well the model is able to identify and accurately collect positive cases.

R can be calculated as

$$\text{R} = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False Negative (FN)}}. \quad (35)$$

5.4.5. F_1 Score. A statistic used to assess a binary classification model's accuracy is the F-score, sometimes referred to as the F_1 score. To get a single score that strikes a balance between recall and precision, it considers both. Preciseness and recall are balanced by a single number called the F_1 score. Algorithm 2 describes the selection of best detection strategy from various classifiers based on F1 score.

The following formula determines the F_1 score:

$$F_1 = \frac{2\text{PR}}{P + R}, \quad (36)$$

where R represents recall and P represents precision

5.4.6. $F_{0.5}$ Score. Another version of the F_1 score that prioritises precision over recall is the $F_{0.5}$ score. It works best for tasks when we wish to minimise false positives at the expense of some false negatives, since it is especially helpful when we want to place a higher value on precision. The following formula determines the $F_{0.5}$ score:

$$F_{0.5} = \frac{1.25\text{PR}}{0.25P + R}. \quad (37)$$

5.4.7. F_2 Score. Recall is given more weight than precision in the F_2 score, which is an additional variation of F_1. This makes it more appropriate for situations where we wish to minimise false negatives at the expense of some false positives. It is especially helpful when we want to assign more weight to recall (Algorithm 2). The following formula determines the F_2 score:

$$F_2 = \frac{5\text{PR}}{4P + R}. \quad (38)$$

6. Results and Discussion

Consider the system with a STATCOM-based damping controller connected at midpoint of transmission line near bus 5 to achieve equal compensation on both sides of system. Different attack templates are considered, where the attackers target the measurement signal of STATCOM-based supplementary damping controller. Figure 14 represents different DIA magnitude applied to time domain simulation.

6.1. Ramp Attack. Attack signal is increased or decreased in order to modify true signal. Slope of 0.05 is applied at $t = 25$ sec for manipulation of signal. The ramp attack vector a_{ramp} is given by

$$a_{\text{ramp}} = \begin{cases} 0, & 0 \geq \triangle_t \leq 24, \\ 0.05, & 25 \geq \triangle_t. \end{cases} \quad (39)$$

6.2. Step Attack. Attack signal is given by adding positive or negative value to the true signal. Therefore, ±0.5 step magnitude is applied at $t = 6$ sec to $t = 7$ sec for manipulation of signal. The step attack vector a_{step} is given by

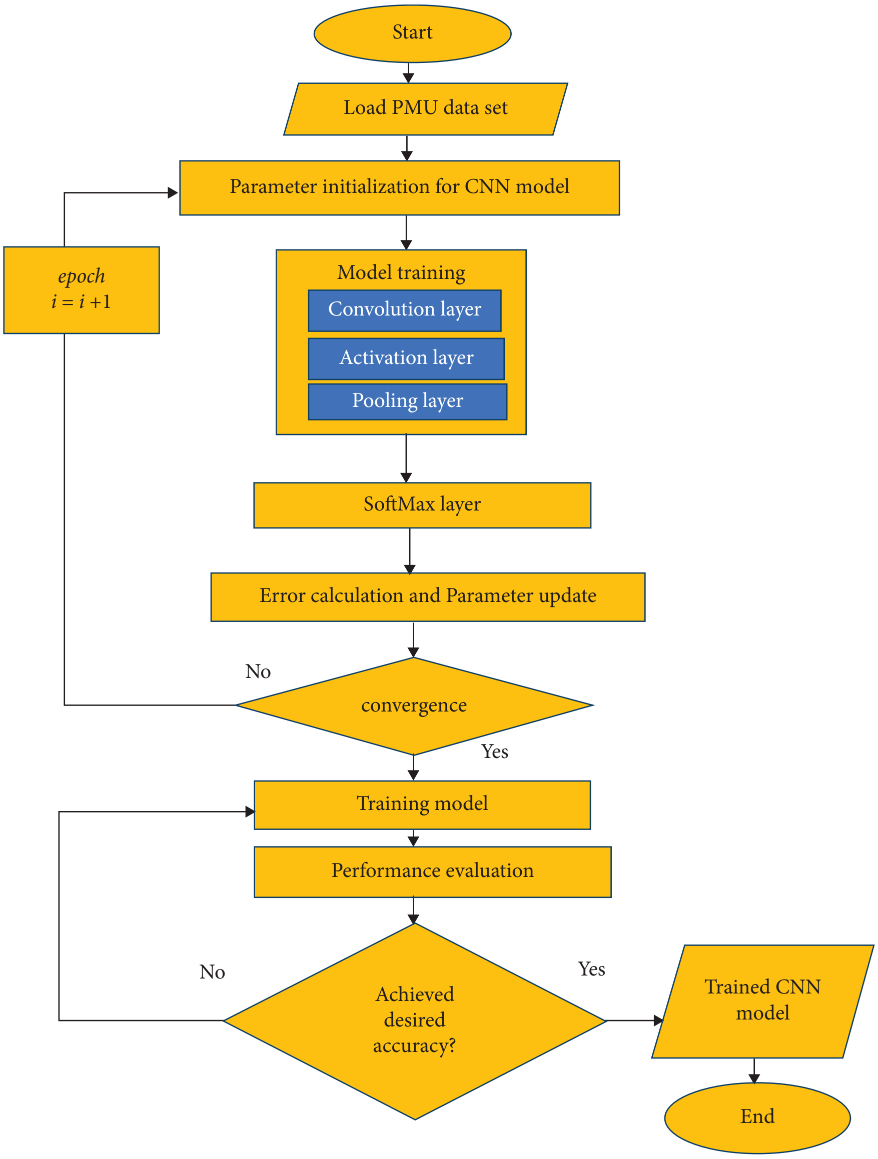


Figure 13: Flowchart for CNN.

```
Input: The trained dataset ($D_1$) and tested dataset ($D_2$) with label $y_j \in \psi = \{0, 1\}$
Output: For attack detection, the best classifier $l^* \in L$
Initialize $F_1 = 0$;
  for $i \in L$ do
    train the classifier $j$ with trained dataset $D_1$;
    test the classifier with tested dataset $D_2$;
    calculate F1 score $\text{F1}_{j;}$
      if $\text{F1}_j > \text{F1}$ then
      $\text{F1} = \text{F1}_{j;}$
      $\text{l}^* = j$;
    end if
    end for
  return $\text{l}^*$;
```

Algorithm 2: For choosing the best attack detection strategy from various classifiers.

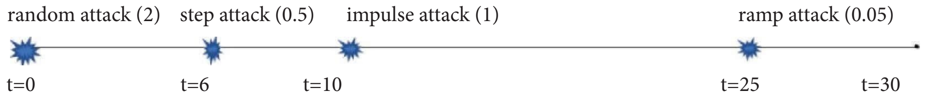


Figure 14: Different DIA applied to time domain simulation of the WSCC system.

$$a_{\text{step}} = \begin{cases} 0, & 0 \geq \triangle_t \leq 5, \\ \pm 0.5 & 6 \geq \triangle_t \leq 7, \\ 0 & 8 \geq \triangle_t. \end{cases} \tag{40}$$

6.3. Random Attack. Attack signal is injected by adding a random value to true signal. A constant value of 2 is added for manipulation of signal. The random attack vector a_{random} is given by

$$a_{\text{random}} = 2 \text{ for } 0 \geq \triangle_t. \tag{41}$$

6.4. Impulse Attack. Attack signal is injected by sudden increase of signal at $t = 10$ sec for signal manipulation. The impulse attack vector a_{impulse} is given by

$$a_{\text{impulse}} = \begin{cases} 0, & 0 \geq \triangle_t \leq 9, \\ 1, & 10 \geq \triangle_t. \end{cases} \tag{42}$$

6.5. Model-Based Method. From Figure 15, all the 9 buses of the WSCC system, ramp attack causes increase of voltage magnitude whereas for random attack there is a decrease of voltage magnitude from normal value compare to step and impulse attack. Impulse attack has very low impact on affecting bus voltage. In step attack bus 2, 3, 7, 8, and 9 have decreased voltage magnitude, and the remaining bus voltage measurements have slight deviation from nominal value.

Due to limitation of space, simulation result of all the generators and transmission line parameters are not shown. Therefore, some important power system parameters like tie-line power flow, voltage magnitude, speed deviation of generator, and active power flow are analysed for attack impact. In Figure 16, the active power flow of G2 is 0.93 pu under normal condition, but for random attack, it is raised to 0.99 pu. The attacked response for other DIA are within 0.94 pu which shows that the attack impact has very small variation from nominal value and less distortion than random attack.

In the above Figure 17, for all type of attack there is an increase of overshoot of oscillation for rotor speed deviation of generator 1. The random attack has highest overshoot than other attack types.

In Figure 18, it is shown that all attack types have impact the system with varying magnitude of tie-line power flow. The random and ramp attack shows decreased magnitude of tie-line power flow from nominal value.

In Figure 19, after the onset of ramp attack takes place at 25th second the voltage magnitude deviated from nominal value of 1.03 pu to 1.08 pu. Whereas for random attack the measured voltage magnitude of STATCOM is 1.09 pu starting from initial condition.

If the integrity attack occurs on the wide-area communication network of STATCOM-based damping controller, the control signal of damping controller will be varied, followed by change in voltage magnitude of transmission line, change in the tie-line power flow, and then change in rotor speed. Finally, it will affect the active power output of generator. As time progresses, it eventually causes grid collapse. From the above analysis of attack impact, it is inferred that step and impulse attack has very little impact to system. Ramp attack and random attacks have highest impact on system because all the parameters are drastically changed from nominal value. Thus, the random and ramp attack have more chances to disrupt the grid than other attack does.

From Figures 20(a), when the random attack occurs there is distorted waveform from the initial condition, where RPME slowly reaches the nominal value after 10 seconds. In Figure 20(b), the impulse attack causes the amplitude of signal raises to value of 6 and reaches the nominal point at $t = 18^{\text{th}}$ second in RPME. But the RLSE shows less deviation response to impulse attack. In Figure 21(a), for the 0.05 slope of ramp attack signal in RPME, the amplitude varies between 1.9 and 0.4 and settles at 30th second. Whereas in RLSE it takes long time to settle to nominal point when it encounters ramp attack. In Figure 21(b), for step attack, RPME and RLSE are used to estimate the system parameters in online in order to detect the anomaly. By monitoring system parameters online, it is evident that RPME locates the attack accurately. The performance of RPME and RLSE are validated based on estimation error is shown in Figure 22. RPME has least estimation error than RLSE.

The distribution of power into the various frequency components that make up a signal is described by the time series' power spectrum. In order to estimate the spectral density of a random signal from a series of time samples, spectral density estimation is used.

The average power (P) of N periodic signal is given by

$$\begin{aligned} P &= \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2, \\ P &= \frac{1}{\omega} \int_{-\omega/2}^{\omega/2} I(\omega) d\omega, \end{aligned} \tag{43}$$

where $I(\omega) = (1/N)|\sum_{n=0}^{N-1} x(n) e^{-j\omega n t}|^2$

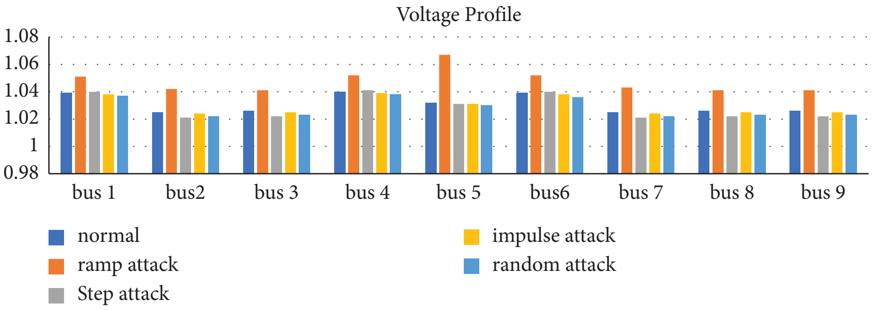


Figure 15: Voltage magnitude (per unit) at all the buses.

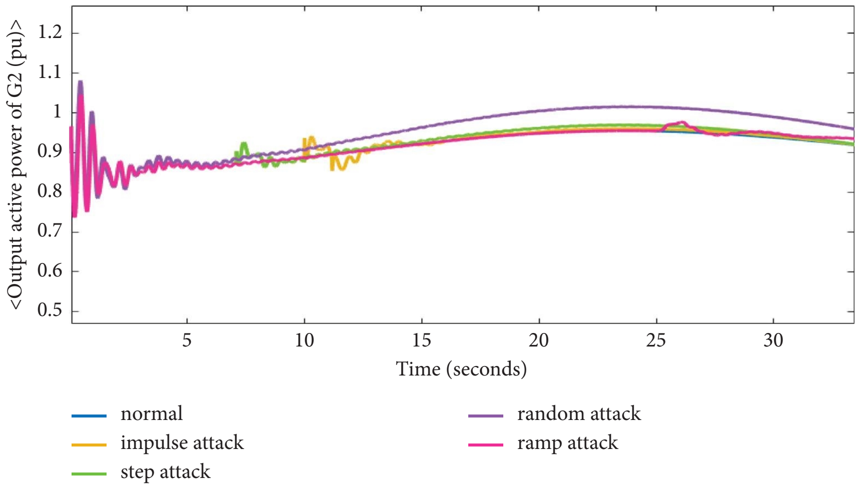


Figure 16: Active power of generator 2.

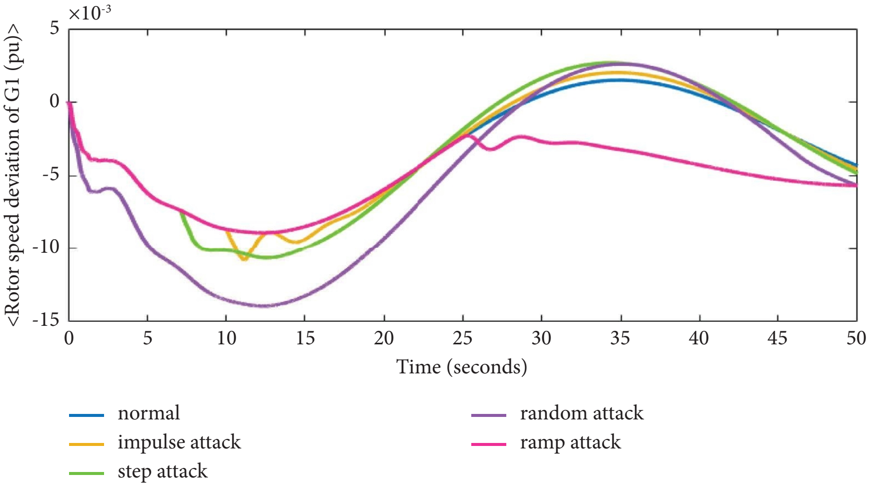


Figure 17: Rotor speed deviation of generator 1.

Sum of powers in various frequency components is represented by above equation.

In Welch's method, the data is divided into overlapping segments, a modified periodogram is calculated for each segment, and the periodograms are then averaged in order to measure the power spectral density. The methodology is based on the notion of utilising periodogram spectrum estimates, which are the outcome of converting time domain signals to frequency domain signals. The frequency domain response for different data integrity attack is shown in Figure 23. During normal system operation the power per frequency is above -50 dB/radians per sample, whereas in DIA scenario it is noted that the power per frequency decreases for all types of

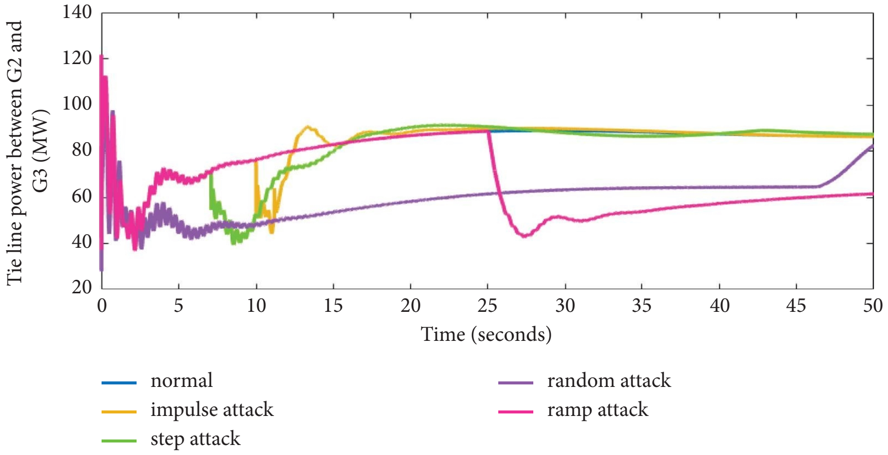


Figure 18: Tie line power flow between generator 2 and generator 3.

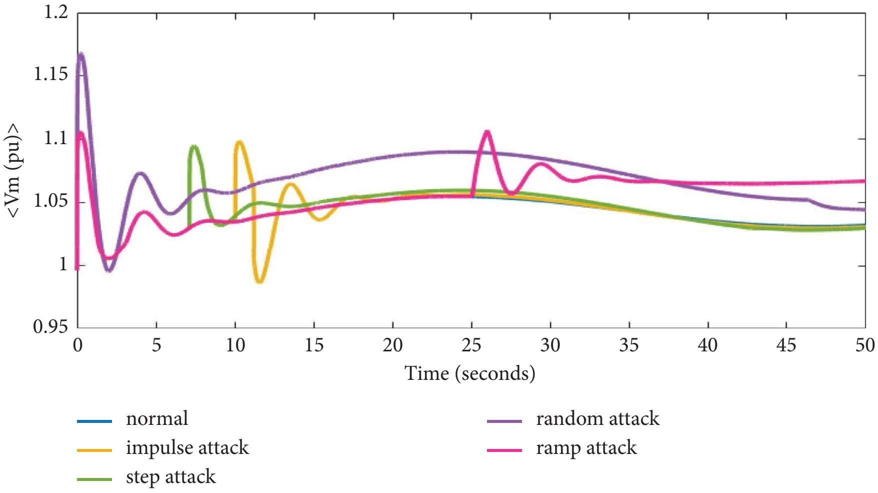
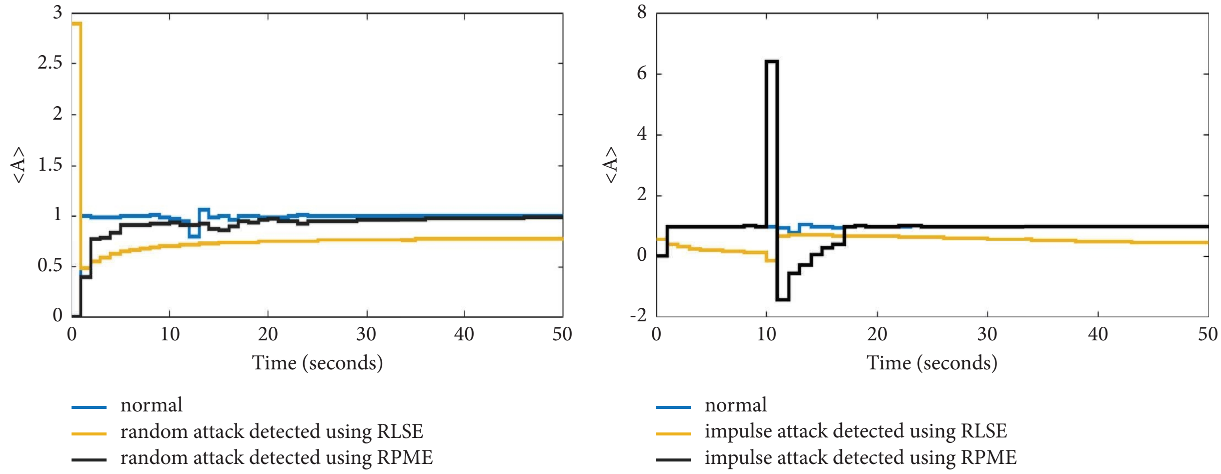


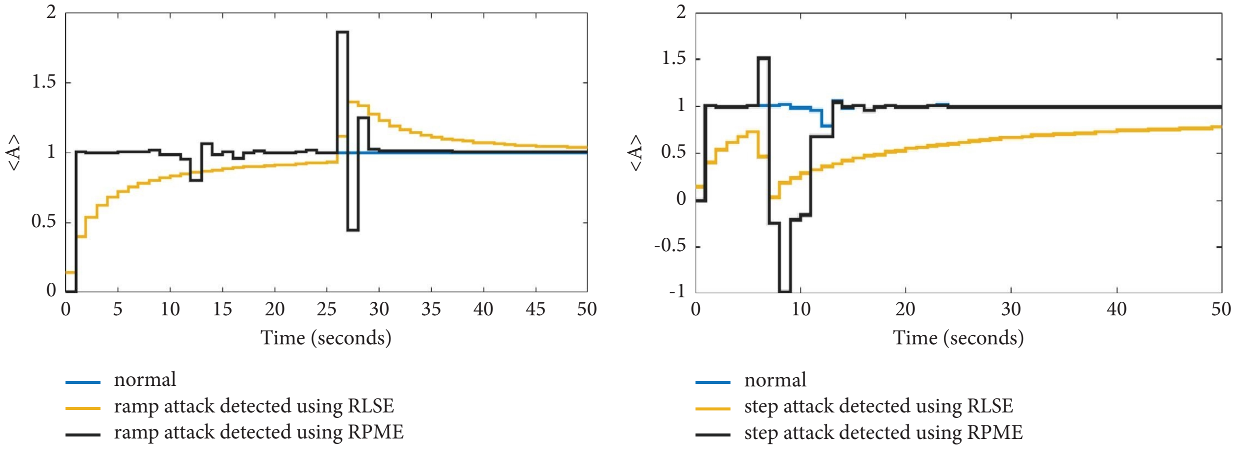
Figure 19: Measured voltage at STATCOM.



(a)

(b)

Figure 20: Estimated parameters of system matrix under random attack and impulse attack.



(a)

(b)

Figure 21: Estimated parameters of system matrix under ramp attack and step attack.

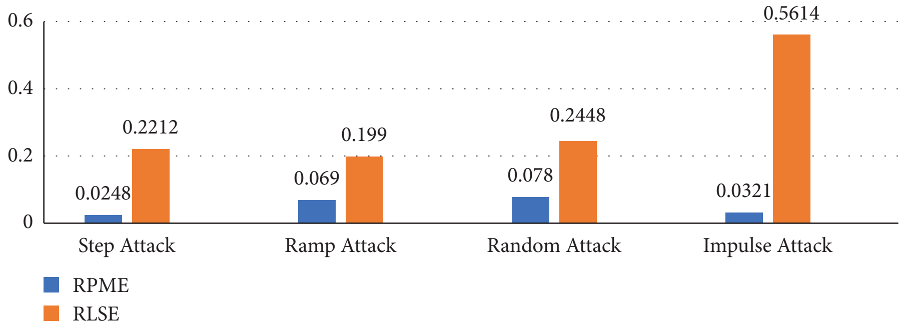


Figure 22: Estimation error of RPME and RLSE for different types of DIA.

attack. It is clearly shown that analysing the signal in frequency domain will efficiently detect the change in system operation during cyber-attack.

6.6. Data-Driven Method. Among 25000 of PMU measurement data, 20000 data are used to train and 5000 data are used for testing the classifier. Data samples collected during each attack response are less compare to normal response. This causes imbalanced dataset; therefore, by changing the decimation in "To workspace" block of the Simulink model the balanced dataset can be obtained. For data preprocessing, normalisation and standardisation are not needed because the trained data samples are in per unit values which lies between 0 and 1. Consider the dataset D, where it each data consist of PMU measurement represented as d_j^i for m number of instances, $i = \{1, 2, 3\ldots m\}$. Depending on buffer length and sampling rate of PMU, each time series length d_j^i, $j = \{1, 2, 3\ldots n\}$ that corresponds to time stamps $\{1, 2, 3\ldots t\}$.

For a set of PMU measurements D, datasets are classified into p different classes, namely,

$$\begin{aligned}
C_1 &= \left\{D_{C_1}^1, D_{C_1}^2, D_{C_1}^3 \ldots .D_{C_1}^{p1}\right\},\\
C_2 &= \left\{D_{C_2}^1, D_{C_2}^2, D_{C_2}^3 \ldots .D_{C_2}^{p2}\right\},\\
C_3 &= \left\{D_{C_3}^1, D_{C_3}^2, D_{C_3}^3 \ldots .D_{C_3}^{p3}\right\},\\
C_4 &= \left\{D_{C_4}^1, D_{C_4}^2, D_4^3 \ldots .D_{C_4}^{p4}\right\},\\
C_5 &= \left\{D_{C_5}^1, D_{C_5}^2, D_{C_5}^3 \ldots .D_{C_5}^{p5}\right\}.
\end{aligned} \tag{44}$$

The five classes, namely, normal, step attack, random attack, ramp attack, and impulse attack, respectively, correlate to various events. Precision, accuracy, and recall of different classifiers are shown in Figures 24–26, respectively. F_1 score, $F_{0.5}$ score, and F_2 score are shown in Figures 27–29, respectively.

From Table 2, high F_1-score shows that the CNN model is minimising false positives and false negatives while efficiently recognising positive samples for all attack cases. From Table 3, the results obtained indicate that the aforementioned strategies appear promising for detecting injection attacks. The detection accuracy is based on the type of

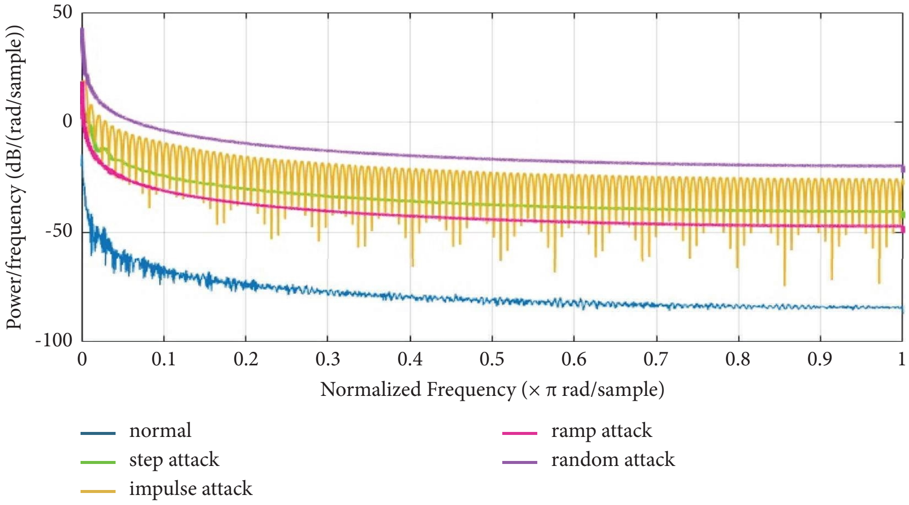


Figure 23: Normal and attacked response in frequency domain.

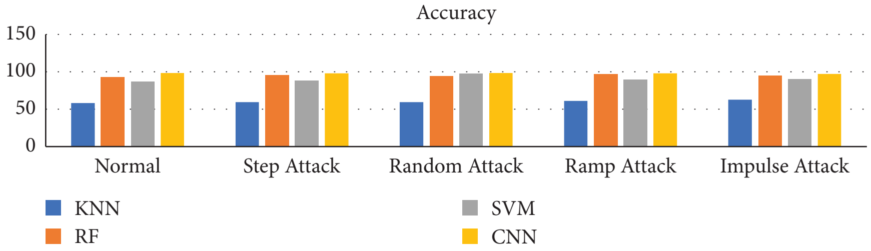


Figure 24: Accuracy of different classifiers.

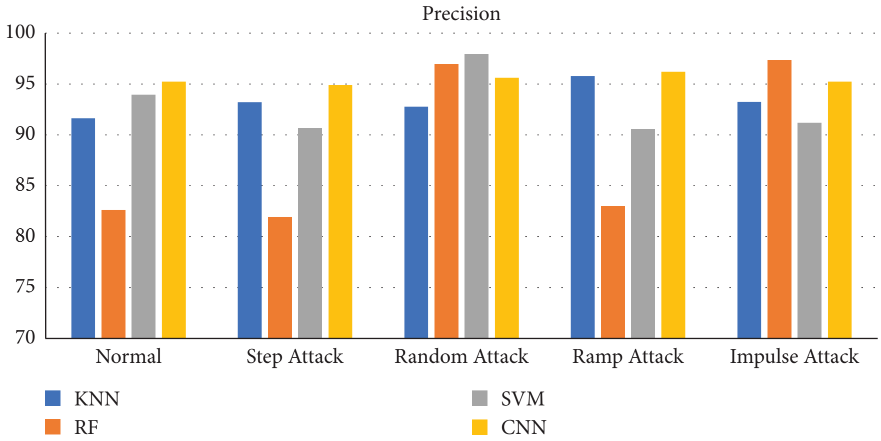


Figure 25: Precision of different classifiers.

dataset, extracted features, and network design, and varies amongst algorithms depending on these factors. Simulations are done on MATLAB 2021 in Intel Core i5-1135G7 CPU with 16 GB of RAM DELL laptop. The analysis shown above demonstrates that CNN accurately studies PMU data to find instances of DIA resulting from malicious measurements. It is concluded that DL performs better than the ML technique based on the examined performance metrics. Due to the fact that the ML-based strategy mostly depends on feature engineering to extract pertinent information. Without the requirement for feature engineering, the deep structure of the DL-based method enables them to automatically learn complex features from the raw data. Moreover, the raw datasets will be processed using DL methods to identify and extract relevant patterns. Therefore, it is efficient in detecting the attack due to its deep structure and handling of large dataset.

When a random attack occurs, there is a decrease in voltage magnitude at all the buses, a decrease in tie-line power between G1 and G2, followed by an increase of

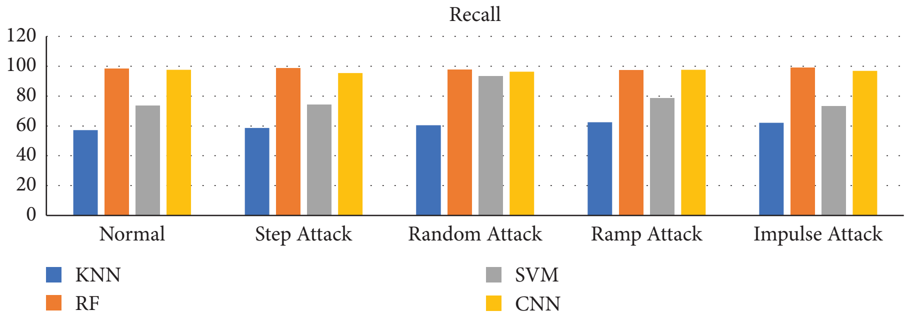


Figure 26: Recall of different classifiers.

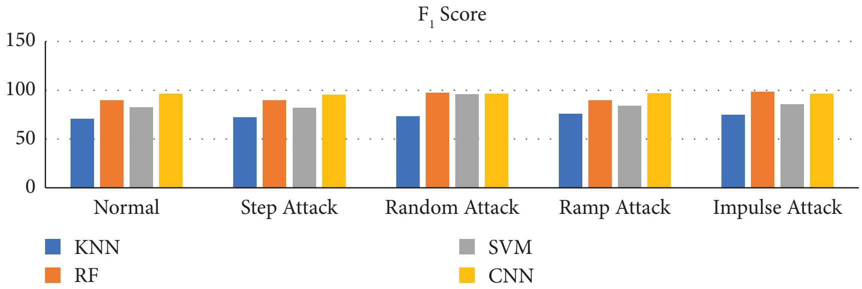


Figure 27: F_1 score of different classifiers.

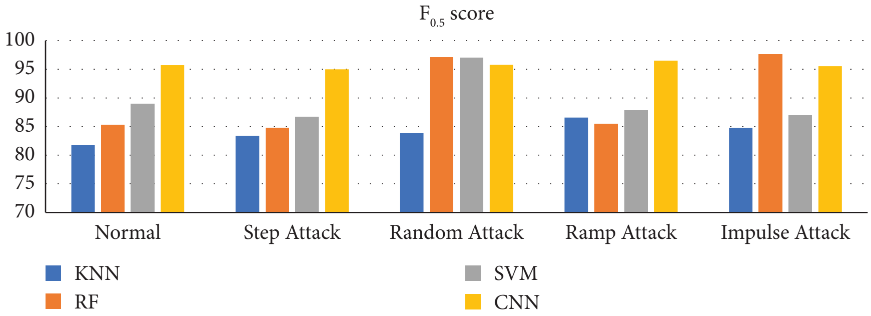


Figure 28: $F_{0.5}$ score of different classifiers.

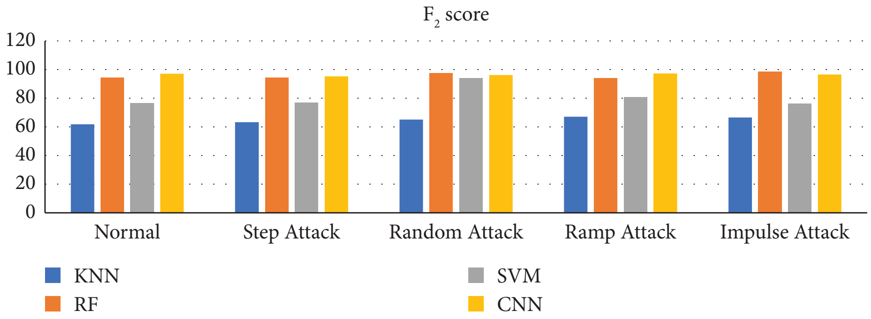


Figure 29: F_2 score of different classifiers.

Table 2: $F_{0.5}$ score, F_1 score, and F_2 score of different classifiers.

Response	$F_{0.5}$ score				F_1 score				F_2 score			
	KNN	RF	SVM	CNN	KNN	RF	SVM	CNN	KNN	RF	SVM	CNN
Normal	81.75	85.33	88.98	95.7	70.36	89.72	82.46	96.41	61.75	94.5	76.82	97.12
Step attack	83.37	84.77	86.7	94.98	71.98	89.42	81.63	95.12	63.31	94.6	77.02	95.25
Random attack	83.83	97.07	96.99	95.74	73.24	97.30	95.62	95.95	65.01	97.5	94.28	96.15
Ramp attack	86.55	85.5	87.84	96.5	75.63	89.64	84.1	96.89	67.15	94.17	80.66	97.29
Impulse attack	84.75	97.62	86.94	95.55	74.56	98.2	85.2	96.04	66.55	98.5	76.28	96.52

Table 3: Accuracy, precision, and recall of different classifiers.

Response	Accuracy				Precision				Recall			
	KNN	RF	SVM	CNN	KNN	RF	SVM	CNN	KNN	RF	SVM	CNN
Normal	58.12	93.25	87.16	98.32	91.63	82.63	93.94	95.23	57.10	98.14	73.48	97.61
Step attack	59.36	95.63	88.01	97.96	93.21	81.93	90.65	94.89	58.62	98.41	74.24	95.35
Random attack	59.45	94.46	97.68	98.63	92.78	96.92	97.93	95.61	60.49	97.68	93.41	96.29
Ramp attack	61.23	96.96	89.66	98.21	95.78	82.97	90.53	96.21	62.48	97.47	78.52	97.57
Impulse attack	62.66	95.2	90.25	97.43	93.23	97.33	91.2	95.24	62.11	98.8	73.28	96.85

oscillation in generator speed response. Moreover, the active power generation increases due to the above impact as shown in Figure 16. This can mislead the entire power system operation and lead to system collapse. Ramp attacks also affect the system by increasing the voltage magnitude followed by a change in tie-line power too. But the impact of step attack and impulse attack lasts for a few seconds and the system response regains to nominal value. From the simulation results of all power system parameters, it is inferred that the random attack and ramp attack have most impacted the system. Therefore, considering the random and ramp attack severity the operator can give priority to mitigate these two attacks before it leads to system instability or collapse. Some of the mitigation steps for DIA are discussed below.

7. Mitigation

Some of the mitigation methods for data integrity attack are as follows:

(i) Network mapper is an open-source Linux command utility that scans IP addresses in networks and enables administrators to identify network problems. Alerts can be delivered to the control system operator to make situational awareness.

(ii) Controlled islanding can be implemented to isolate the attacked region.

(iii) System reconfigurations such as system restoration may be made during a DOS and data integrity attack.

(iv) Smart grid privacy can be preserved by implementing secure communication with encryption in vulnerable areas.

(v) To prevent the grid against cyber threats, encryption algorithms such as 3DES, DES, AES, symmetric cyphers, or cryptography are utilised.

(vi) IP fast hopping can be utilised to conceal user communication times and important data to thwart malware. It restricts access for online attackers, allowing only authorised users to access information.

(vii) Isolating the vulnerable PMU and increase the observability of islands.

8. Conclusion

Using model- and data-driven-based methods in the MATLAB/Simulink environment, various types of data integrity attacks are addressed, and their impact and detection are evaluated. The impact analysis led to the conclusion that random and ramp attacks had a greater effect on the system than step and impulse attacks. Simulation results demonstrated that RPME-based detection methods had higher detection accuracy and reduced estimation error when compared to RLSE-based detection methods. Due to impressive characterization accuracy in detecting the attack, CNN-based anomaly detection has a more promising performance in data-driven methods than ML-based methods. In addition, the DL method outperforms other ML-based methods in terms of detection performance because to its automatic understanding of inherent variance in data and automated feature selection. The physical principles of the grid are best understood using model-based approaches, whereas subtle, data-driven anomalies are best detected using learning-based techniques. Thus, the hybrid technique can improve the accuracy and reliability of anomaly detection in CPPS while also giving it the flexibility to adapt to changing circumstances by combining the capabilities of model-based approach and learning-based methods, which can capture complex and dynamic patterns. This work can be utilised for other bus systems. The identification and mitigation of data integrity attacks in large-scale systems will be the focus of our upcoming study in CPPS.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Supplementary Materials

Five different datasets obtained from the simulation of the WCSCC 3 machine 9-bus system, normal and attacked measurement, along with their label, are given in tabular form in supplementary materials. (*Supplementary Materials*)

References

[1] R. V. Yohanandhan, R. M. Elavarasan, P. Manoharan, and L. Mihet-Popa, "Cyber-physical power system (cpps): a review on modeling, simulation, and analysis with cyber security applications," *IEEE Access*, vol. 8, pp. 151019–151064, 2020.

[2] R. V. Yohanandhan, R. M. Elavarasan, R. Pugazhendhi et al., "A specialized review on outlook of future Cyber-Physical Power System (CPPS) testbeds for securing electric power grid," *International Journal of Electrical Power and Energy Systems*, vol. 136, Article ID 107720, 2022.

[3] A. Farraj, E. Hammad, and D. Kundur, "A distributed control paradigm for smart grid to address attacks on data integrity and availability," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 70–81, 2018.

[4] S. Sridhar and G. Manimaran, "Data integrity attack and its impacts on voltage control loop in power grid," in *Proceedings of the 2011 IEEE Power and Energy Society General Meeting*, pp. 1–6, Detroit, MI, USA, December 2011.

[5] A. Sargolzaei, K. Yazdani, A. Abbaspour, C. D. Crane, and W. E. Dixon, "Detection and mitigation of false data injection attacks in networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4281–4292, 2020.

[6] S. Chen, K. L. Butler-Purry, and D. Kundur, "Impact of cyber-attacks on transient stability of smart grids with voltage support devices," in *Proceedings of the 2013 IEEE Power & Energy Society General Meeting*, pp. 1–5, Vancouver, Canada, July 2013.

[7] D. B. Rawat and C. Bajracharya, "Detection of false data injection attacks in smart grid communication systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1652–1656, 2015.

[8] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, 2014.

[9] S. Basumallik, R. Ma, and S. Eftekharnejad, "Packet-data anomaly detection in PMU-based state estimator using convolutional neural network," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 690–702, 2019.

[10] A. Ashok, M. Govindarasu, and J. Wang, "Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1389–1407, 2017.

[11] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3271–3280, 2018.

[12] R. Deng, P. Zhuang, and H. Liang, "False data injection attacks against state estimation in power distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2871–2881, 2019.

[13] C. Konstantinou and M. Maniatakos, "A data-based detection method against false data injection attacks," *IEEE Design and Test*, vol. 37, no. 5, pp. 67–74, 2020.

[14] R. Tan, H. H. Nguyen, E. Y. S. Foo et al., "Modeling and mitigating impact of false data injection attacks on automatic generation control," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1609–1624, 2017.

[15] S. Li, Y. Yılmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2725–2735, 2015.

[16] R. Liu, C. Vellaithurai, S. S. Biswas, T. T. Gamage, and A. K. Srivastava, "Analyzing the cyber-physical impact of cyber events on the power grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2444–2453, 2015.

[17] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[18] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[19] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[20] B. R. Amin, S. Taghizadeh, M. S. Rahman, M. J. Hossain, V. Varadharajan, and Z. Chen, "Cyber-attacks in smart grid–dynamic impacts, analyses and recommendations," *IET Cyber-Physical Systems: Theory and Applications*, vol. 5, no. 4, pp. 321–329, 2020.

[21] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee, "Ensemble-based algorithm for synchro phasor data anomaly detection," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2979–2988, 2019.

[22] N. Bhusal, M. Gautam, R. M. Shukla, M. Benidris, and S. Sengupta, "Coordinated data falsification attack detection in the domain of distributed generation using deep learning," *International Journal of Electrical Power & Energy Systems*, vol. 134, p. 107345, 2022.

[23] B. Jin, C. Dou, and D. Wu, "False data injection attacks and detection on electricity markets with partial information in a micro-grid-based smart grid system," *International Transactions on Electrical Energy Systems*, vol. 30, no. 12, 2020.

[24] K.-D. Lu and Z.-G. Wu, "Resilient event-triggered load frequency control for cyber-physical power systems under DoS attacks," *IEEE Transactions on Power Systems*, vol. 38, no. 6, pp. 5302–5313, 2023.

[25] K.-D. Lu, Z.-G. Wu, and T. Huang, "Differential evolution-based three stage dynamic cyber-attack of cyber-physical power systems," *IEEE*, vol. 28, no. 2, pp. 1137–1148, 2023.

[26] Y. Zhang, T. Yang, and Z. Tang, "Active fault-tolerant control for load frequency control in multi-area power systems with physical faults and cyber attacks," *International Transactions on Electrical Energy Systems*, vol. 31, no. 7, 2021.

[27] A. S. Leger and J. James, "Cyber-physical systems approach for wide area control applications," in *Proceedings of the IEEE Texas power and energy conference (TPEC)*, pp. 1–6, College Station, TX, USA, February 2018.

[28] Y. Xu, "A review of cyber security risks of power systems: from static to dynamic false data attacks," *Protection and Control of Modern Power Systems*, vol. 5, no. 1, p. 19, 2020.

[29] M. Simevents, *Users Guide*, The Mathworks, Natick, MA, USA, 2015.

[30] C. Brown, F. Milton, S. de Souza, J. Cesar, and J. Glover, "Roots, achievements, and prospects of power system state estimation: a review on handling corrupted measurements," *International Transactions on Electrical Energy Systems*, vol. 29, no. 1, p. e2779, 2018.

[31] M. Esmalifalak, N. T. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 808–813, Atlanta, GA, USA, December 2013.

[32] D. H. Lakshminarayana, J. Philips, and N. Tabrizi, "A survey of intrusion detection techniques," in *Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1122–1129, Boca Raton, FL, USA, December 2019.

[33] M. Ma, A. Lahmadi, and I. Chrisment, "Detecting a stealthy attack in distributed control for microgrids using machine learning algorithms," in *Proceedings of the 3rd IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, Wuhan, China, June 2020.

[34] R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1148–1153, Jaipur, India, December 2016.

[35] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 240217, 8 pages, 2014.

[36] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[37] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Computer Science*, vol. 89, pp. 213–217, 2016.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[39] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[40] J. Yang and J. Li, "Application of deep convolution neural network," in *Proceedings of the 2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 229–232, Chengdu, China, December 2017.