Time Series Analysis Neural Networks for Detecting False Data Injection Attacks of Different Rates on Power Grid State Estimation

Time Series Analysis Neural Networks for Detecting FDIAs

DANUSHKA SENARATHNA

School of Electrical, Computer and Biomedical Engineering, Southern Illinois University Carbondale, Carbondale, Illinois, United States, h.senarathna@siu.edu

SPYROS TRAGOUDAS

School of Electrical, Computer and Biomedical Engineering, Southern Illinois University Carbondale, Carbondale, Illinois, United States, spyros@siu.edu

JASON WIBBENMEYER

Ameren Corporation Missouri, St Louis, Missouri, United States, jwibbenmeyer@ameren.com

NASSER KHDEER

Ameren Corporation Missouri, St Louis, Mississippi, United States, nkhdeer@ameren.com

False Data Injection Attacks (FDIAs) that target the state estimation pose an immense threat to the security of power grids. Deep Neural Network (DNN) based methods have shown promising results in detecting such FDIAs. Among existing state-of-the-art DNN models, time series analysis DNNs have demonstrated superior FDIA detection capability. This paper discusses the challenges associated with applying time series analysis DNNs for detecting FDIAs and emphasizes the impact of the attack rate on the detection rate of attacks. We demonstrate that existing time series analysis DNNs are highly vulnerable to FDIAs executed at low attack rates. This paper presents various alternative implementations for time series classifiers and time series predictors to improve the FDIA detection rate. A novel method is proposed to train time series classification neural networks to detect FDIAs of any attack rate with high efficiency. Subsequently, an enhanced FDIA detection framework that includes a time series classifier and multiple predictors is presented. Furthermore, an analytical criterion is derived to estimate the FDIA detection rate of time series analysis DNNs under any attack rate. Experimental results obtained on IEEE bus systems using state-of-the-art DNN architectures support the effectiveness of the proposed training method and the proposed framework. The proposed training method significantly improved the detection rate of FDIAs at low attack rates. Up to a 48% improvement in the FDIA detection rate was observed in the proposed framework when compared to the state-of-the-art.

CCS CONCEPTS • Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection systems

Additional Keywords and Phrases: Artificial Intelligence, Power Grid Cybersecurity, False Data Injection Attacks (FDIA), Long Short Term Memory (LSTM), Neural Networks, Smart Grid, Transformer Neural Network

1 INTRODUCTION

Reliability is one of the most crucial aspects of a power grid. The functionality of numerous essential sectors in every country, including healthcare, economy, national security, transportation, communication, etc., depends on a consistent and dependable operation of the power grid. If the reliability of the power

ACM 2471-2566/2025/03-ART
http://dx.doi.org/10.1145/3723164

grid is compromised, it can result in catastrophic consequences. Therefore, ensuring high system reliability in power grids is paramount for maintaining the stability of those critical sectors. Modern smart power grids are cyber-physical systems comprising advanced controlling, sensing, communication, and information technologies integrated with the physical infrastructure. The dependency of the smart grids on cyber resources makes them susceptible to cyber-attacks. There have been many attack incidents in history, such as the 2003 slammer worm attack at the Davis-Besse nuclear plant in Ohio, USA [1], the cyber-attack on the Ukrainian power grid in 2015 [2] followed by another attack in 2016 [3], etc.

The growing dependency of power grids on advanced technology infrastructure is leading to increased risks to grid security from cyber threats. There are various types of attacks targeting power grids [4]. They include malware-based attacks [5], phishing attacks [5], False Data Injection Attacks (FDIAs) [6], Denial of Service (DoS) attacks [7], Man-in-the-Middle (MitM) [8], and control system intrusions [9], among others. Attackers are becoming more sophisticated and continuously evolving, resulting in attacks that are more complex, precise, and advanced in nature. Numerous studies have shown that there is a significant occurrence of cyberattacks on power grids [12]. Therefore, it is of utmost importance to develop methods to detect cyberattacks to ensure the reliability of the power grids.

In FDIAs, the attacker manipulates the grid measurements by injecting malicious data, which causes an incorrect grid state estimation [10]. It may result in blackouts and damage to the infrastructure of the grid. While random FDIAs are detected by the Bad Data Detection (BDD) criterion in the state estimation tool, [6] showed that an attack can be systematically constructed to bypass the BDD. Such attacks cause a much higher threat to power grids when compared to randomly generated attacks. FDIA detection has been extensively investigated and there exist numerous approaches proposed in the literature to detect FDIAs in power grids [10]-[13]. FDIA detection methods can be divided into two categories [12], model-based algorithms and data-driven algorithms. Model-based algorithms rely on the topology of the grid and mathematical models to detect anomalies, using predefined rules and equations that describe the behavior of the power system. In contrast, data-driven algorithms detect anomalies using machine learning by identifying patterns in historical data. Machine learning-based data-driven methods are gaining more attraction and popularity. They utilize various machine learning techniques such as Principal Component Analysis [14], Clustering [15], Support Vector Machines [16], and DNNs [17]-[28], among others.

DNNs have shown great success in detecting FDIAs, outperforming other machine learning methods. Various architectures, including Fully Connected Neural Networks (FCNNs) [32], Convolutional Neural Networks (CNNs) [33], Graph Neural Networks (GNNs) [34], Recurrent Neural Networks (RNNs) [35], Gated Recurrent Unit (GRU) networks [36], Long Short Term Memory (LSTM) networks [37], Transformer networks [39], etc. are used in DNN-based approaches. They classify either a single measurement vector or a sequence of measurement vectors. Single measurement vector classification methods employ architectures such as FCNNs as in [17], CNNs as in [18], and GNNs as in [19], [20]. Measurement vector sequence classification methods employ time series analysis architectures such as RNNs as in [21], [22], GRUs as in [23], LSTMs as in [24]- [26], Transformers as in [27], and hybrid architectures as in [28], [29]. Spatial-temporal models combining GNN and time series analysis architectures have also been proposed in the literature for FDIA detection [30], [31]. Among existing state-of-the-art DNN models, time series analysis DNNs have demonstrated superior FDIA detection capability.

Our study specifically focuses on sequence classification-based FDIA detection methods that identify attacks by detecting anomalies in grid measurement sequences using time series analysis DNNs. Detecting the presence of an FDIA in a sequence of measurement vectors is a time series anomaly detection problem. Time series anomaly detection is one of the fundamental problems in deep learning [41]. Neural network architectures such as RNNs, GRUs, LSTMs, and Transformers are used for time series anomaly detection in different application domains, including power grids. Existing time series analysis DNN-based FDIA detection methods use either classification neural networks [21]-[24], [27] or prediction neural networks [25], [28]. Our recent approach in [26] combined a classifier with predictors to improve the detection rate of FDIAs. However, the approach in [26] often misclassified benign instances as attacked, i.e., had a high false positive rate, limiting its usability. The framework proposed in this paper addresses this limitation and has a low rate of false positives.

Four main factors need to be considered when applying time series analysis DNNs for power grid FDIA detection: DNN model, attack magnitude, detection window size, and attack rate. Many different DNN models already have been studied in the literature [23]-[28]. LSTM-based and Transformer-based models stand out with higher detection rates. Among others, [23] and [27] have shown that the detection rate decreases as the attack magnitude decreases, and were unable to detect a substantial percentage of small-magnitude attacks. The diagnostic resolution of the FDIA detection, which is the ability to locate the attacks in the time domain, depends on the detection window size. The smaller the window size, the better the diagnostic resolution. Therefore, it is important to achieve a higher detection rate on smaller window sizes, which is challenging. For instance, [28] reported a very high FDIA detection rate using time series analysis DNNs with a window size of 49, whereas studies such as [23] and [24] reported a significant reduction in the detection rate for window sizes below 10.

Even though the impacts of the DNN model, attack magnitude, and window size have been adequately investigated, the attack rate has been overlooked in previous studies. Most existing studies such as [22]-[25], [27], [28] assumed that the attacker injects false data into every measurement vector taken from the grid (i.e., every time frame). This assumption is overly optimistic and may not necessarily be true in a real attack situation. The attacker may inject false data sparsely, where only some of the measurement vectors are attacked, at an unknown attack rate. Therefore, it is essential to develop generalized FDIA detection methods that can cope with any attack rate. The studies in [24] and [26] have considered sparse attacks of some specific attack rates. To our knowledge, none of the existing studies have comprehensively investigated the ability of time series analysis DNNs to detect sparse FDIAs across different attack rates, nor have they provided generic solutions to cope with FDIAs at varying attack rates.

In particular, the existing time series classification networks for FDIA detection are trained assuming that the attack rate is known a priori. The conventional approach followed in the literature for time series classifiers is to use the same attack rate to create both the training and testing datasets. Furthermore, they are trained on a dataset of a specific attack rate and optimized only for that particular attack rate. Therefore, the existing time series classification networks do not provide a good detection rate for FDIAs across all attack rates. On the other hand, time series prediction networks for FDIA detection are typically trained using only benign data, with no need to account for the attack rate during training. However,

predictor-based FDIA detectors must be designed appropriately to achieve a higher detection rate at any attack rate, a factor that has not been considered in the literature.

Moreover, the FDIA detection rate of time series analysis DNNs must be systematically evaluated under different attack rates. Although studies such as [24] and [26] have presented the experimental results on some specific attack rates, they have not evaluated performance across a sufficient number of attack rates. For a precise evaluation of the detection rate, a time series analysis DNN-based FDIA detection method must be evaluated at many different attack rates with a substantial number of samples for each rate, which is a tedious task.

Our study shows that sparse attacks injected at low attack rates are much more difficult to detect, and existing time series analysis DNNs exhibit a very high failure rate against such attacks. We present various alternative implementations for time series classifiers and time series predictors to improve the FDIA detection rate. To alleviate the vulnerability against hard-to-detect sparse FDIAs executed at low attack rates, a novel method is proposed to train time series classification neural networks, such that the network learns to detect FDIAs injected at any rate with very high efficiency. We employ the proposed time series analysis DNN models and the classifier training method to design an FDIA detection framework with enhanced detection capability of sparse attacks injected at any rate. The proposed framework includes a time series classification network and multiple prediction networks that are working in conjunction. Furthermore, we present a criterion to analytically assess the FDIA detection rate of time series analysis DNNs under any attack rate efficiently, without excessive experimentation. We show experimentally that the FDIA detection rate achieved with the proposed training method and the enhanced FDIA detection framework outperforms the existing methods on the IEEE 14-bus system [44] and the IEEE 30-bus system [45]. Specifically, the detection rate of FDIAs of low attack rates improved significantly. Up to a 48% improvement in the detection rate was observed when compared to the state-of-the-art.

The rest of the paper is organized as follows. Section 2 provides preliminaries and Section 3 describes the proposed methodologies. Section 4 presents and analyzes experimental results, and Section 5 concludes.

2 PRELIMINARIES

2.1 False Data Injection Attacks on Power Grid State Estimation

The state of a power grid is determined using the state estimation [10] based on the measurements obtained from the grid. State estimation relies on the power flow model which is a set of equations that describes the energy flow through the grid. State estimation uses the same underlying equations of the power flow analysis. The AC power flow model describes the grid operation using a set of nonlinear equations considering real and reactive power. The DC power flow model describes the grid operation using a set of linear equations considering only real power. The state estimation that relies on the DC power flow model is referred to as DC state estimation. It is computationally efficient compared to AC state estimation. The experimental evaluations in our study are conducted considering the DC state estimation. The grid state in DC state estimation is defined by the phase angles of the buses. DC state

estimation measurements include real power injections of the buses and real power flows of the lines. Let vector $\mathbf{z} = (z_1, z_2, \ldots, z_m)^T$ denote the measurement vector consisting of m measurements z_i, $1 \le i \le m$, and vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ denote the grid state consisting of n state variables x_j, $1 \le j \le n$ where $m \ge n$. Grid state $\boldsymbol{x}$ and measurements $\boldsymbol{z}$ are related as in Equation (1),

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}, \quad (1)$$

where $\mathbf{H}$ denotes the measurement Jacobian matrix of the power grid of size $m \times n$, and $\mathbf{e} = (e_1, e_2, \ldots, e_m)^T$ denotes the measurement noises. State variables can be estimated using the Minimum Mean Squared Error (MMSE), assuming the measurement noises are normally distributed with a zero mean. Let σ_i^2 be the variance of i^{th} measurement and $\mathbf{W}$ be a diagonal matrix whose elements are the reciprocals of the measurement error variance. The estimated state $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{x}} = (\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{z}. \quad (2)$$

The measurement vector is identified as containing bad data by the BDD criterion if $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \boldsymbol{\tau}$, where $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ denotes the 2-Norm of measurement residuals and $\boldsymbol{\tau}$ is a predetermined threshold value.

Let $\mathbf{a} = (\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{m})^\mathrm{T}$ denote the attack vector, i.e., the malicious data added to the original measurements. Let $\mathbf{z_a}$ be the measurement vector with the malicious data, and $\hat{\mathbf{x}}_\mathbf{bad}$ be the grid state estimated using $\mathbf{z_a}$. We have that

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a}, \quad (3)$$

$$\hat{\mathbf{x}}_\mathbf{bad} = \hat{\mathbf{x}} + (\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{a}. \quad (4)$$

When the attack vector is a linear combination of the columns of $\mathbf{H}$, i.e. $\boldsymbol{a} = \mathbf{H}\mathbf{c}$ for some arbitrary vector $\mathbf{c}$, and given that the original measurement vector $\mathbf{z}$ passes through the BDD, i.e. $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \le \boldsymbol{\tau}$, then,

$$\|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{bad}\| = \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + (\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{a})\|$$
$$= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{H}\mathbf{c} - \mathbf{H}(\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathbf{T}\mathbf{W}\mathbf{H}\mathbf{c})\|$$
$$= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \le \boldsymbol{\tau}, \quad (5)$$

and $\mathbf{z_a}$ bypasses the BDD.

2.2 FDIA Detection Using Time Series Analysis DNNs

The objective of FDIA detection using time series analysis DNNs is to identify whether a given sequence of grid measurement vectors is benign or attacked. Let w denote the length of the measurement vector sequence being evaluated, and we refer to it as the detection window size. Let $\mathbf{z^t} = (z_1^t, z_2^t, \ldots . z_m^t)$ be the measurement vector taken from the grid at time t, where z_i^t denotes the i^{th} measurement of the grid at time t. The objective is to classify the sequence of measurement vectors $\mathbf{S^t} = \{\mathbf{z^{t-w+1}}, \mathbf{z^{t-w+2}}, \ldots . , \mathbf{z^t}\}$ at time t as benign or attacked. There are two learning techniques that are used to train DNNs for time series anomaly detection: supervised learning and unsupervised learning. Supervised learning employs times series classification DNNs trained on labeled data to detect anomalies, whereas unsupervised learning utilizes prediction DNNs trained on benign data to forecast future data points and detect anomalies by computing the deviation between the predicted and the observed data.

Time series classification DNNs used for FDIA detection are binary classification networks. They are trained using labeled data (supervise anomaly detection) to classify a sequence of measurement vectors as

benign or attacked. The classifier takes the sequence of measurement vectors within the detection window as the input and outputs a binary label indicating if the window is benign or attacked. Let $\mathbf{S_c^t}$ denote the input sequence for the time series classifier network at time t, i.e., $\mathbf{S_c^t} = \mathbf{S^t}$. The lookback window size w_c of the classifier is the same as the w.

A predictor DNN is trained using only the benign measurement vectors (unsupervised anomaly detection). It predicts the expected measurements for the current time frame t based on a sequence of past measurements up to $t-1$. Such a prediction neural network is referred to as a prediction autoencoder. Let $\mathbf{S_p^t}$ denote the input sequence for the predictor network at time t, where $\mathbf{S_p^t} = \{\mathbf{z^{t-w+1}}, \mathbf{z^{t-w+2}}, \dots, \mathbf{z^{t-1}}\}$. Let $\hat{\mathbf{z}}^t = (\hat{z}_1^t, \hat{z}_2^t, \dots, \hat{z}_m^t)$ denote the predicted measurement vector at time t. The measurement vector sequence $\mathbf{S^t}$ is classified as benign or attacked based on the prediction error, i.e., the difference between $\hat{\mathbf{z}}^t$ and $\mathbf{z^t}$. The lookback window size of the predictor w_p is $w-1$.

There are three different configurations of prediction neural networks: The Multivariate Multi-prediction model, denoted by M, takes time series data of multiple variables as the input and predicts multiple variables. The Multivariate Single-prediction model, denoted by MS, takes time series data of multiple variables as the input and predicts a single variable. Finally, the Univariate Single-prediction model, denoted by S, takes time series data of a single variable as the input and predicts the same variable.

LSTM [37] and Transformer [39] architectures are established as the state-of-the-art for time series data analysis problems as they have shown superior results compared to the other architectures. Network models based on LSTM and Transformer architectures are considered in this study. LSTM and Transformer architectures are explained in the following.

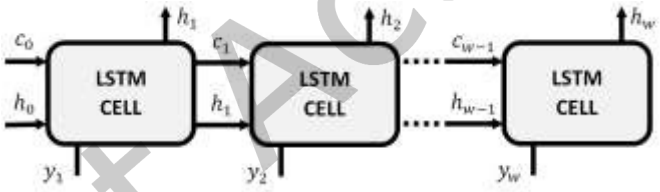


Figure 1: An unrolled view of an LSTM unit.

LSTM [37], [38] is a popular RNN architecture that can memorize information over a longer period and learn the long-term dependencies. An LSTM layer consists of multiple LSTM units. An LSTM unit, comprising a single cell, takes a sequence of data vectors and processes them sequentially. It produces a sequence of outputs corresponding to each position, i.e., each time step, of the input sequence. Let us consider an input data sequence of length w (lookback window size). Let y_i denote the data vector at position i in the input sequence where $1 \leq i \leq w$. Figure 1 shows the unrolled view of the operation of an LSTM unit. For some position i of the sequence, the inputs to the LSTM cell consist of the input vector y_i of the current position i, the hidden state of the previous position h_{i-1}, and the cell state of the previous position c_{i-1}. The cell generates the hidden state h_i, which is the output, and the cell state c_i for the current position i of the sequence. A multivariate LSTM has multiple variables in y_i, and a univariate LSTM has only one variable in y_i.

Despite the ability of the LSTMs to learn long-term dependencies better than the conventional RNNs, they suffer from vanishing or exploding gradient problems. In addition, since the cell state is computed sequentially for each time step, computations of an LSTM unit are not parallelizable. The Transformer neural network architecture was proposed to address these shortcomings.

The Transformer neural network architecture was proposed in [39]. Even though it was originally proposed as a large language model for language translation, variants of the Transformer were later effectively adapted in numerous applications for processing sequence data including time series data [40]. Unlike RNNs, the Transformer processes the entire input sequence simultaneously using the self-attention mechanism instead of sequential processing. This allows the network to learn the long-term dependencies more efficiently, with enhanced parallelizability.

For an input sequence of length w, where y_i denote the data vector at position i in the sequence, the self-attention operates as follows. Let d_{model} denote the dimension of the input vector y_i. A query vector q_i, a key vector k_i and a value vector v_i are computed for each input data y_i in the sequence by projecting the data using weight matrices obtained during the training. Inputs to the attention are the queries, keys, and values. The dimensions of the query and key vectors are the same. Let d_k be the dimension of the query and key vectors, d_v be the dimension of the value vectors. For each position i in the sequence, w score values are computed by taking the dot product of the query vector q_i with each key vector k_j where $1 \leq j \leq w$ in the sequence. The softmax function is applied across these scores after being divided by the $\sqrt{d_k}$. The self-attention output for position i is the weighted sum of the value vectors v_j in the sequence where the weight is the softmax output corresponding to position j. This computation is repeated for every position i in the sequence. The self-attention for a sequence is computed using matrix operations as in Equation (6) below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right)V. \quad (6)$$

A Multi-head self-attention module consists of multiple such attention modules referred to as heads. The multi-head output is obtained by concatenating the outputs of all the heads and projecting them with a weight matrix W^O as,

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \quad (7)$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right), \quad (8)$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are projection parameter matrices and h is the number of heads.

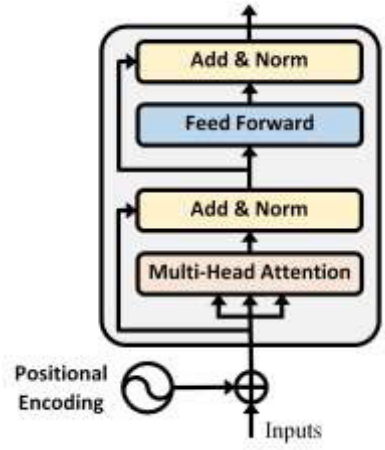


Figure 2: The structure of a Transformer encoder [39].

The Transformer encoder has a multi-head self-attention module followed by a position-wise fully connected feed-forward layer. The multi-head attention module and the feed-forward networks are followed by a residual connection and a layer normalization as shown in Figure 2. Positional encoding [42] is added to the inputs to incorporate the order of the data within the sequence using the criterion in [39]. Positional encodings consist of a sequence of vectors of the dimension d_{model}. The original Transformer architecture in [39] consists of a stack of encoders and decoders designed using multi-head self-attention. The decoder is designed for sequence generation tasks. Since the classification and prediction tasks in our study do not require sequence generation, the Transformer-based DNNs are implemented using only the Transformer encoders in this paper.

3 THE PROPOSED FDIA DETECTION METHODS AND EVALUATION CRITERION

Time series analysis DNN-based FDIA detectors work as shown in Figure 3. A measurement vector $\mathbf{z}^t$ obtained from the power grid at time t is first fed into the state estimation tool and then the BDD is applied. Thereafter, $\mathbf{z}^t$ is forwarded to the DNN-based FDIA detector which classifies the measurement vector sequence within the detection window as benign or attacked. A sequence of measurements is classified as attacked if either the BDD or DNN-based FDIA detector indicates the presence of malicious data. In the figure, the arrow labeled as "No" implies benign and the arrow labeled as "Yes" implies attacked. The DNN-based FDIA detector outputs a binary label (0 or 1) for the detection window indicating if the window is benign or attacked. In this paper, the benign instances are represented by label 0 and the attacked instances are represented by label 1. The following outlines alternative methods for implementing the DNN-based FDIA detector. In particular, Section 3.1 presents a time series classifier along with the novel training method and Section 3.2 presents a time series predictor. Section 3.3 presents the proposed enhanced approach that combines a time series classifier with multiple predictors. Finally, in Section 3.4, we present the analytical criterion to estimate the detection rate of time series analysis DNNs under any attack rate.

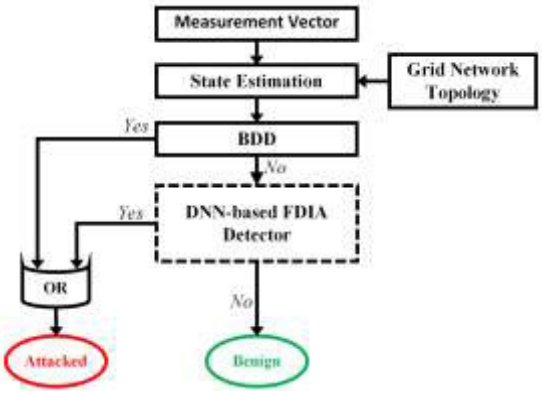


Figure 3: Time series analysis DNN-based FDIA detection.

3.1 FDIA Detection Using Time Series Classification Neural Networks

A classifier-based FDIA detector consists of a time series classification neural network, denoted as $\boldsymbol{\mathcal{C}}$ in Figure 4. The time series classification network outputs a binary label for the measurement vector sequence within the detection window indicating if the window is benign or attacked. Transformer, LSTM, and hybrid models that consist of both LSTM and Transformer can be used to implement the classifier $\boldsymbol{\mathcal{C}}$. The appropriate DNN model is selected experimentally. The proposed classifier $\boldsymbol{\mathcal{C}}$ is implemented using a Transformer neural network. The most challenging task in the implementation of a time series classification neural network for FDIA detection is to train the network to detect attacks injected at any rate with high efficiency. The following explains the modeling of the attack rate, challenges associated with training time series classifiers for FDIA detection, and the proposed training method.

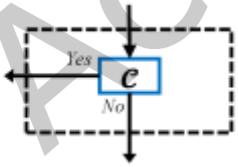


Figure 4: Time series classifier-based FDIA detector.

The attack rate can be modeled with the probability of the attacker injecting false data. Let p denote the attack rate, which is the probability of the attacker injecting false data at a given time. The number of attacks expected within the detection window depends on the attack rate. Assuming each time frame is independent, and the attack probability is constant at each time frame, the probability distribution of the number of attacks within the detection window follows a binomial distribution. Let λ denote the number of attacks within the detection window where $0 \leq \lambda \leq w$ and $F(\lambda)$ denote the probability distribution of λ. $F(\lambda)$ is a function of the attack rate p and detection window size w as in Equation (9).

$$F(\lambda) = p^{\lambda}(1-p)^{w-\lambda}\binom{w}{\lambda} \tag{9}$$

Figure 5 shows the distribution $F(\lambda)$ for three different attack rates $p = 25\%$, $p = 50\%$, and $p = 75\%$ when $w = 20$. The higher the attack rate, the higher the probability of having a higher number of attacks

within the detection window. The lower the attack rate, the lower the probability of having a higher number of attacks within the detection window.

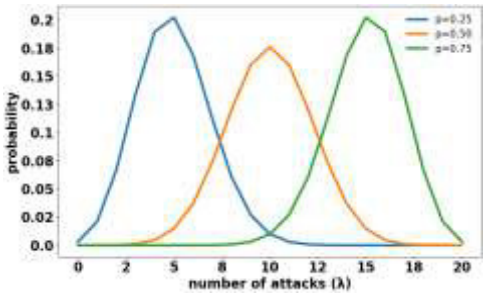


Figure 5: The probability distribution $\boldsymbol{F(\lambda)}$ of the number of attacks within the detection window $\boldsymbol{\lambda}$ for three different attack rates $\boldsymbol{p = 25\%}$, $\boldsymbol{p = 50\%}$, and $\boldsymbol{p = 75\%}$ when $\boldsymbol{w = 20}$.

Since the probability of having λ attacks within the detection window varies at different attack rates as shown in Equation (9), to achieve a high FDIA detection rate across all attack rates, a time series classifier must be able to detect instances with any number of attacks within the detection window. This makes training time series classifiers to detect FDIAs of any attack rate a challenging task. Since classification neural networks are trained using supervised learning, the training dataset should be balanced among the classes. Additionally, the training dataset should include an adequate number of samples from all the different variants of each class. When considering the attack class, there are $2^w - 1$ possible ways that attacks can be placed at different time frames within the detection window. Even for a small window size, it is inefficient to create a training dataset consisting of many samples from every possible permutation to train a time series classification neural network for detecting FDIAs of different attack rates.

However, our experiments in Section 4 show that the classifier detection rate is correlated with the number of attacks within the detection window λ. The detection becomes harder as the number of attacks within the window decreases. Detecting an instance with a smaller number of attacks within the window is much more difficult. On the other hand, it is easier to detect an instance with many attacks within the window. This correlation suggests the possibility of creating a training dataset by considering the number of attacks within the window, rather than considering every possible permutation of the attacks within the window. Thereby, the necessity to consider every possible permutation of the attacks in classifier training is avoided. The dataset is created in the proposed training method by ensuring a sufficient number of samples for each λ such that the detection rate is maximized for every λ.

Furthermore, we found that a model trained on a dataset of a low attack rate provides sufficiently high detection accuracy for higher attack rates as well. For instance, let us consider the aforementioned three attack rates $p = 25\%$, $p = 50\%$, and $p = 75\%$. A network trained on $p = 25\%$ dataset provides high accuracy for $p = 50\%$ and $p = 75\%$. This is because the network extracts the features of attacks more efficiently when there are fewer attacks within the detection window. Thereby, the classifier correctly identifies the easy-to-detect attack instances at higher attack rates. Based on this, one could train the classifier using an attack dataset that follows the distribution of a low attack rate. The lowest attack rate that ensures the false positive rate is within an acceptable range is used to create the training dataset. In

conventional time series classifier training setups, the number of samples for each λ within the attacked data of the training dataset resembles the $F(\lambda)$ of some attack rate p. In the proposed method, the number of samples in the attacked data of the training dataset for each λ where $1 \leq \lambda \leq w$ follows the shifted binomial distribution $G(\lambda)$ in Equation (10) below.

$$G(\lambda) = \hat{p}^{\lambda-1}(1-\hat{p})^{w-\lambda}\binom{w-1}{\lambda-1}. \quad (10)$$

$\hat{p}$ is the minimum probability that ensures the false positive rate remains below a user-defined maximum acceptable percentage. A shifted binomial distribution is used because $\lambda > 0$ for attacked data in the dataset. That way, the training dataset is created to extract the features of hard-to-detect attacks more efficiently. Figure 6 shows the distribution $G(\lambda)$ for w = 20 with $\hat{p} = 10\%$.

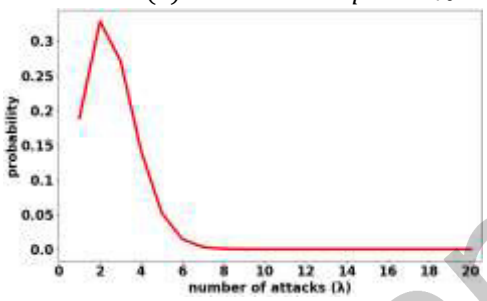


Figure 6: The probability distribution $\boldsymbol{G(\lambda)}$ of the number of attacks within the window $\boldsymbol{\lambda}$ with the proposed training dataset creation method for $\boldsymbol{w = 20}$.

For a given detection window size w, the probability $\hat{p}$ is calculated as described in Algorithm 1. Its inputs are the DNN model $\mathbb{M}$, a validation dataset $\mathbb{D}_{valid}$ that consist of only the benign measurements, and three user-defined parameters. The user-defined parameters are the maximum acceptable false positive rate α, the initial value of $\hat{p}$ denoted as $\hat{p}_0$, and the step size Δ for changing $\hat{p}$. The algorithm computes the optimal probability $\hat{p}$ and the set of weights $\mathbb{W}$ for the model $\mathbb{M}$ that is trained with the dataset created using $\hat{p}$. The initial probability $\hat{p}_0$ is arbitrarily chosen, but a smaller value is assigned for a faster convergence. The create_dataset() function in lines 1 and 11 creates the training dataset $\mathbb{D}_{train}$ in which the number of samples for each λ of the attacked data follows the proposed distribution $G(\lambda)$ of a given $\hat{p}$ value. The model $\mathbb{M}$ is trained at the beginning using a dataset created with $\hat{p}_0$. The train_model() function in lines 2 and 12 trains the classifier model and returns the weights. $\mathbb{W}_{temp}$ is a temporary variable used to store the intermediate weights of the model. After the initial training of the model, the initial false positive rate f_0 is computed in line 3. The sign s_0 of the difference $(f_0 - \alpha)$ is computed in line 4. The sign is computed using the function sign(), which returns $+1$ for positive values, 0 for zero, and -1 for negative values. The model is iteratively trained by adjusting $\hat{p}_0$ by an amount of $\pm\Delta$ based on the sign s of the difference $(f - \alpha)$ until the sing s becomes 0 or changes from its initial value s_0 as described in lines 9 through 20.

For a given detection window size w and attack rate $\hat{p}$, the training dataset for an FDIA detection time series classifier is created as follows. A sequence of measurement vectors corresponding to a particular time period is considered for training. Training data samples are created by applying a moving window of

size w through the measurement vector sequence selected for training. To make sure the training dataset is balanced between the two classes, 50% of the sampled windows are kept benign while the other 50% have attacks injected. For better generalization of the model, either the benign or the attacked sample of a particular detection window is included in the training dataset. Attacks are injected into the selected windows to be attacked, such that the number of attacks within the window, λ, follows the distribution $G(\lambda)$. This proposed training method is independent of the attack rate and provides high detection rates across all attack rates. Furthermore, it is applicable to any time series classification DNN architecture.

ALGORITHM 1: Time series classifier training method

train_DNN($\mathbb{M}$, $\mathbb{D}_{valid}$, α, $\hat{p}_0$, Δ):
1 $\mathbb{D}_{train}$ = create_dataset($\hat{p}_0$)
2 $\mathbb{W}_{temp}$ = train_model($\mathbb{M}$, $\mathbb{D}_{train}$)
3 f_0 = false_positive_rate($\mathbb{M}$, $\mathbb{W}_{temp}$, $\mathbb{D}_{valid}$)
4 $s_0 = s$ = sign(f_0 - α)
5 If $s_0 == 0$:
6 $\hat{p} = \hat{p}_0$
7 $\mathbb{W} = \mathbb{W}_{temp}$
8 else:
9 while $s_0 == s$:
10 $\hat{p}_0 = \hat{p}_0 + s.\Delta$
11 $\mathbb{D}_{train}$ = create_dataset($\hat{p}_0$)
12 $\mathbb{W}_{temp}$ = train_model($\mathbb{M}$, $\mathbb{D}_{valid}$)
13 f = false_positive_rate($\mathbb{M}$, $\mathbb{W}_0$, $\mathbb{D}_{valid}$)
14 s = sing(f - α)
15 if $s == 0 \,||\, s_0 == s$:
16 $\hat{p} = \hat{p}_0$
17 $\mathbb{W} = \mathbb{W}_{temp}$
18 end
19 end
20 end
21 return $\hat{p}$, $\mathbb{W}$
end

3.2 FDIA Detection Using Time Series Prediction Neural Networks

A predictor-based FDIA detector consists of a time series prediction neural network $\boldsymbol{\mathcal{P}}$ followed by a prediction error vector classifier $\boldsymbol{\mathcal{C}_P}$ as shown in Figure 7. The predictor network $\boldsymbol{\mathcal{P}}$ takes the first $w - 1$ measurement vectors within the detection window and predicts the measurement vector for the last time frame (i.e., the current time frame). The $\boldsymbol{\mathcal{C}_P}$ assigns a binary label to the detection window based on the difference between the predicted measurement vector and the observed measurement vector of the last time frame of the window.

The predictor network $\boldsymbol{\mathcal{P}}$ is trained to predict the expected measurement vector for the current time frame t. It is trained using only the benign measurement vectors. Since the predictor is trained using only benign data, the error between the predicted and the observed measurements becomes larger if there exists an attacked measurement within the lookback window or in the current time frame i.e., predicted time frame. Transformer, LSTM, and hybrid models that consist of both LSTM and Transformer can be used for the predictor. The appropriate model for the predictor is selected experimentally.

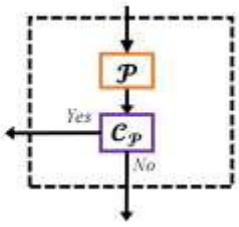


Figure 7: Time series predictor-based FDIA detector.

In addition, the alternative multivariate and univariate configurations M, MS, and S of Section 2.2 were also considered in implementing the predictor. An important factor in the predictor network is the sensitivity of the model for the attacks. The dimension of the input data space in a multivariate predictor is m times higher than that of a univariate predictor. Small perturbations of the measurements caused by the attacks get masked as they propagate through a multivariate predictor since the attacks are typically sparsely distributed within the detection window. Therefore, the prediction error due to false data injections is more pronounced in a univariate predictor than in a multivariate predictor. Our experiments reveal that the univariate LSTM outperforms the alternative schemes. The proposed predictor $\boldsymbol{\mathcal{P}}$ is implemented using a univariate LSTM model.

The $\boldsymbol{\mathcal{C}_{\mathcal{P}}}$ module calculates the difference between the predicted and the observed measurement vectors, and classifies the measurements within the detection window as either benign or attacked. Let $\hat{e}_i^t = \hat{z}_i^t - z_i^t$ denote the prediction error of the i^{th} measurement and $\hat{E}^t = \{\hat{e}_1^t, \hat{e}_2^t \dots \hat{e}_m^t\}$ denote the prediction error vector at time t. Various methods can be used in $\boldsymbol{\mathcal{C}_{\mathcal{P}}}$ to classify the prediction error vector and four alternative approaches are presented. The first method, called Thresholding and referred to as *Thresh* applies a threshold on each measurement to determine whether $\hat{E}^t$ indicates an attack. In particular, let $\hat{e}_i^{max}$ denote the maximum expected prediction error of the i^{th} measurement for benign data. If there exist $\hat{e}_i^t$ such that $\hat{e}_i^t > \hat{e}_i^{max}$ for any $1 \leq i \leq m$ at time t, then the measurement vector sequence within the detection window is labeled as attacked. The second method, referred to as *2-Norm*, computes the 2-Norm of the $\hat{E}^t$, denoted as $\left\|\hat{E}^t\right\|_2$, and the measurement vector sequence is labeled as attacked if the norm exceeds a predetermined threshold τ. In particular, we have an attack only when $\left\|\hat{E}^t\right\|_2 > \tau$. The third method, referred to as *FCNN*, uses a binary classification FCNN to classify the $\hat{E}^t$ as benign or attacked. The fourth method referred to as *KNN*, $\hat{E}^t$ is classified as benign or attacked using the K-Nearest Neighbor (KNN) [43]. The KNN algorithm computes the distance between the input data point and the data points in the training dataset and finds the nearest K data points to the input data. The label

of the class that has the majority among the nearest K data points is assigned to the input data. The $\mathcal{C}_{\mathcal{P}}$ module of the proposed predictor-based FDIA detector is implemented using the *Thresh* approach.

3.3 The Enhanced FDIA Detection Framework

Typical time series analysis DNN-based FDIA detection methods consist of either a classifier or a predictor. In contrast, the proposed framework consists of a detection mechanism implemented using a time series classifier and multiple univariate predictors to detect FDIAs. Two orthogonal detection methods, one that uses a classifier and the other that uses predictors, working in parallel are used in the framework to increase the detection rate. See also Figure 8. The proposed DNN-based enhanced FDIA detector consists of a time series classifier $\mathcal{C}$, a time series predictor $\mathcal{P}$, and a prediction error vector classifier $\mathcal{C}_{\mathcal{P}}$. The classifier $\mathcal{C}$, predictor $\mathcal{P}$, and prediction error vector classifier $\mathcal{C}_{\mathcal{P}}$ of the proposed framework are implemented as described in Sections 3.1 and 3.2. The classifier network $\mathcal{C}$ and the prediction error vector classifier $\mathcal{C}_{\mathcal{P}}$ assign binary labels to the detection window indicating if the window is benign or attacked. Measurement vectors within the detection window are labeled as attacked if either one of the binary labels from $\mathcal{C}$ or $\mathcal{C}_{\mathcal{P}}$ indicate attacked, otherwise, they are labeled as benign. The proposed framework is more robust and dependable because it generates multiple outputs and uses them in conjunction to make the decision.

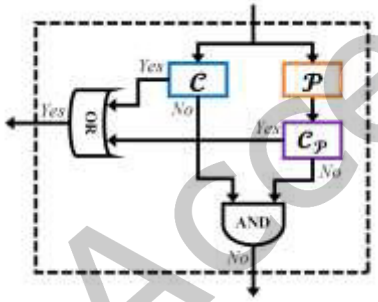


Figure 8: The proposed enhanced FDIA detector that consists of a time series classifier and multiple univariate predictors.

3.4 Estimating the FDIA Detection Rate of Time Series Analysis DNNs at Different Attack Rates

The expected detection rate at a given attack rate p is estimated considering the probability $F(\lambda)$ of having λ attacks within the detection window and the detection rate of the DNN-based detector when there are λ attacks within the window. At first, the detection rates are obtained for each λ, where $1 \leq \lambda \leq w$, by ensuring an ample number of samples for each λ in the testing dataset. Thereafter, detection rates of different λ are scaled by the $F(\lambda)$ as in Equation (9) to obtain the expected detection rate for a given attack rate p.

Let R_λ denote the observed detection rate of the time series analysis DNN-based FDIA detection method, when there are λ attacks within the detection window of size w. The expected detection rate R_p at the attack rate of p is,

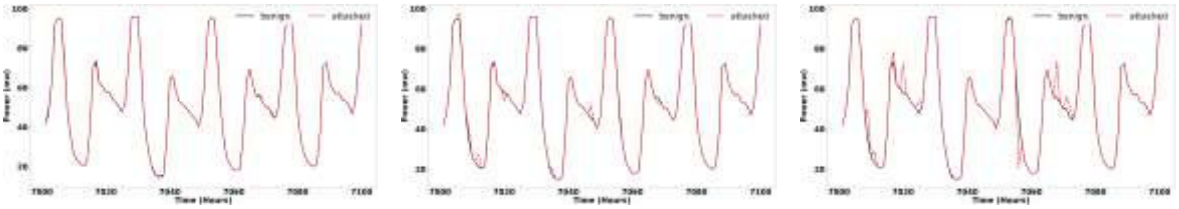
$$R_p = \frac{\sum_{\lambda=1}^{w} F(\lambda) R_\lambda}{\sum_{\lambda=1}^{w} F(\lambda)}. \quad (11)$$

4 EXPERIMENTAL RESULTS

Experiments were conducted on the IEEE 14-bus system [44] and the IEEE 30-bus system [45]. The IEEE 14-bus system has 14 buses and 20 lines, and the IEEE 30-bus system has 30 buses and 41 lines. DC state estimation was considered for the experimental evaluation. However, the methodologies and theories presented in this paper are not limited to DC state estimation and are also applicable to AC state estimation. Time series data of the grid measurements of the two IEEE bus systems, obtained from [46], were used in the experiments. The measurement dataset in [46] was generated by running the DC power flow simulation, and the load dataset for power flow was created by adopting the load profile data retrieved from [47]. Each measurement dataset consisted of 8,760 measurement vectors, representing hourly measurements over a year. Measurements included real power injections of the buses and the real power flows of the lines (forward line flow and backward line flow). The IEEE 14-bus system has 54 measurements (i.e., $m = 54$) and the IEEE 30-bus system has 112 measurements (i.e., $m = 112$). Gaussian noise up to 4% of the true values were added to all the measurements to mimic the presence of noise in real power systems [27]. The attack vector datasets of the two systems used in [48] were used in the experiments. The attacks were generated following the stealthy attack generation criterion described in Section 2.1. It was assumed that the attacker could compromise a maximum of five buses at a time [48]. The time series data of the grid measurements and the attack vector dataset used in our study are available in [52].

Attack vectors were categorized into three categories based on the magnitude, and performance was evaluated separately for those three attack magnitude categories. In previous studies, such as [23] and [24], attacks were categorized based on the deviation in the vector $\mathbf{c}$ due to the attack. We chose to differentiate the attacks based on the values in the attack vector $\mathbf{a}$, because the detection rate of a DNN depends more on the vector $\mathbf{a}$ than on vector $\mathbf{c}$. Attacks with $\mathbf{a_{max}} \leq 4\text{MW}$ were considered as weak attacks, attacks with $4\text{MW} < \mathbf{a_{max}} \leq 10\text{MW}$ were considered as medium attacks, and attacks with $\mathbf{a_{max}} > 10\text{MW}$ were considered as strong attacks.

The attack datasets obtained from [48] had very few attack vectors of weak and medium categories. To ensure a sufficient number of distinct attack vectors in all magnitude categories, the attack dataset of each IEEE bus system was extended as follows. Every attack vector in the dataset was scaled down by the factors of $1/2$, $1/4$, and $1/8$ to create stealthier attack vectors. Let A denote the original attack dataset. The extended attack dataset was $\{A, A/2, A/4, A/8\}$. Note that $\mathbf{a} = \mathbf{Hc} \Rightarrow \boldsymbol{\mu}\mathbf{a} = \mathbf{H}\boldsymbol{\mu}\mathbf{c}$ and $\mathbf{a}' = \boldsymbol{\mu}\mathbf{a}$, where $\boldsymbol{\mu} \in \mathbb{R}$, is also a valid stealthy attack vector. Figure 9 illustrates weak, medium, and strong attack scenarios considering the power injection measurement at bus 4 of the IEEE 14-bus. Benign data is plotted with a solid black line, while attacked data is plotted with a dotted red line. In the weak attack scenario, the difference between attacked and benign measurements was hardly perceptible compared to the medium and strong attack scenarios, which makes the detection of weak attacks more challenging.



(a) weak attack (b) medium attack (c) strong attack

Figure 9: An example of (a) weak, (b) medium, and (c) strong attack scenarios considering the power injection of bus 4 of the IEEE 14-bus system with an attack rate $\boldsymbol{p = 25\%}$.

The training dataset for every DNN was created using the measurement data up to the first 50% of the hours, the validation dataset was created using the data up to the next 25% of the hours, and the testing was done using the data of the remaining hours. For each benign window instance of size w, a corresponding attacked instance was created. The complete dataset consisted of a total of $2 \times (8760 - w + 1)$ samples (window instances). The classifier and predictor networks were trained following the methods described in Sections 3.1 and 3.2 respectively. The validation dataset was used for setting up the $\boldsymbol{C_P}$ module of the predictor-based FDIA detectors.

The proposed DNN-based FDIA detection methods were evaluated through extensive experimentation along various parameters. The parameters varied and their values are shown in Table 1. The evaluation was done for the three attack magnitudes considering three different detection window sizes and the attack rates between $5\% - 100\%$. Four DNN models were considered for the implementation of the classifier and the predictor: LSTM denoted as L, Transformer denoted as T, Transformer followed by an LSTM denoted as TL, and LSTM followed by a Transformer denoted as LT. Furthermore, different implementations of the predictor and the prediction vector classifier $\boldsymbol{C_P}$ were also considered.

Table 1: Parameters considered in the experimental evaluation.

Parameter	Values
Grid topology	IEEE 14-bus, IEEE 30-bus
Window size (w)	10, 20, 30
Attack rate (p)	5% - 100%
Attack magnitude	weak, medium, strong
Classifier model	L, T, TL, LT
Predictor model	L, T, TL, LT
Predictor configuration	M, MS, S
$\boldsymbol{C_p}$ method	*Thresh, 2-Norm, KNN, FCNN*

Table 2 shows the four DNN model architectures used for the classifier and predictor in detail. The table shows the layer type and the dimension of the layer output. Each network takes a sequence of measurements of dimension $w \times v$ as the input, where w denotes the lookback window size and v denotes the number of variables. Parameters l, f, d_1, and d_{out} denote the number of LSTM units, the size of the first feed-forward layer in the Transformer encoder, the number of neurons in the first fully connected layer, and the number of outputs, respectively. The parameter values for classifier networks were $w = w$, $v = m$, $l = 128$, $d_1 = 32$, and $d_{out} = 1$. The parameter values for predictor networks were: $w = w - 1$,

$l = 100$, and $d_1 = 50$. When the predictor configuration was M, $v = d_{out} = m$, when it was MS, $v = m$ and $d_{out} = 1$, and when it was S, $v = d_{out} = 1$. Residual$_1$ and Residual$_5$ denote the residual connections from layers 1 and 5, respectively. A Global Average pooling [49] layer was used in the T and LT models to reduce the Transformer encoder output to a vector. A single encoder with 20 heads of head size 5 was used in all Transformer-based models. Rectified Linear (ReLU) activation was used in the hidden layers. Classification networks were trained for 75 epochs using a batch size of 32 with an initial learning rate of 10^{-4}. Predictors were trained for 25 epochs using a batch size of 50 with an initial learning rate of 10^{-3}. The Adams optimizer [50] was used in all the training. Implementation was done in Python programming language using the Keras Application Programming Interface (API) [51].

Table 2: Architectures of the four DNN models used in the experiments.

Index	LSTM (L)	Transformer (T)	Transformer-LSTM (TL)	LSTM-Transformer (LT)
1	Input $w \times v$	Input $w \times v$	Input $w \times v$	Input $w \times v$
2	LSTM l	Positional Encoding $w \times v$	Positional Encoding $w \times v$	LSTM $w \times v$
3	Fully Connected d_1	Multi-head Attention $w \times v$	Multi-head Attention $w \times v$	Multi-head Attention $w \times v$
4	Fully Connected d_{out}	Residual$_1$ $w \times v$	Residual$_1$ $w \times v$	Residual$_1$ $w \times v$
5		Layer Normalization $w \times v$	Layer Normalization $w \times v$	Layer Normalization $w \times v$
6		1D Convolution $w \times f$	1D Convolution $w \times f$	1D Convolution $w \times f$
7		1D Convolution $w \times v$	1D Convolution $w \times v$	1D Convolution $w \times v$
8		Residual$_5$ $w \times v$	Residual$_5$ $w \times v$	Residual$_5$ $w \times v$
9		Global Average Pooling w	LSTM l	Global Average Pooling w
10		Fully Connected d_1	Fully Connected d_1	Fully Connected d_1
11		Fully Connected d_{out}	Fully Connected d_{out}	Fully Connected d_{out}

Three user-defined parameters of Algorithm 1 were set to $\alpha = 2\%$, $\hat{p}_0 = 20\%$, and $\Delta = 2\%$. A shallow FCNN with two fully connected layers and one output neuron ($50 \rightarrow 32 \rightarrow 1$) was used in the *FCNN* method of $\boldsymbol{C_p}$, and KNN was implemented with $K = 21$ (experimentally found optimal value) using the Euclidean distance in the *KNN* method of $\boldsymbol{C_p}$.

FDIA detection performance was evaluated by considering the number of True Positives (TP), False Negatives (FN), True Negatives (TN), and False Positives (FP). TP and TN indicate correctly identified attacked and benign instances, respectively. FN and FP indicate incorrectly identified attacked and benign instances, respectively. Based on these detection scenarios, three evaluation metrics, Recall (REC), Precision (PR), and True Negative Rate (TNR), were computed, where $\text{REC} = \frac{TP}{TP+FN}$, $\text{PR} = \frac{TP}{TP+FP}$ and $\text{TNR} = \frac{TN}{TN+FP}$. The REC, also referred to as the detection rate, is the percentage of correctly identified attacked instances, the PR is the percentage of instances identified as attacked that are correct, and the TNR is the percentage of correctly identified benign instances. The REC indicates the ability to identify attacked samples correctly, PR indicates the validity of the detected attacked samples, and TNR indicates the ability to identify benign samples correctly. Since the testing dataset was balanced between attacked and benign instances, the average of REC and TNR reflects the overall accuracy.

The remainder of the experimental results section is organized into four subsections. Section 4.1 and Section 4.2 present results for time series classifier-based and predictor-based FDIA detection, respectively.

Section 4.3 presents the results of the proposed FDIA detection framework that uses a classifier and predictors. Section 4.4 analyzes the results using the proposed evaluation criterion.

4.1 Time Series Classification-based FDIA Detection

To select the appropriate DNN model for the classifier, we first evaluated the four DNN model architectures trained with the conventional approach, in which the training and testing datasets are created using the same attack rate. Since the primary focus is to improve the detection rate at lower attack rates, an attack rate p of 25% was considered with detection window size w of 20. Attacked data in the training and testing datasets followed distribution $F(\lambda)$ in Equation (9). Tables 3 and 4 show the detection results of the IEEE 14-bus system and the IEEE 30-bus system, respectively, using different classifier models. It was observed that the REC, PR, and TNR of the classifiers depend on the attack magnitude. In both the IEEE bus systems, the detection rate, i.e., REC, decreased as the attack magnitude decreased. The classifier had a higher REC, PR, and TNR for strong and medium attacks compared to weak attacks.

Table 3: FDIA detection results of L, T, TL, and LT time series classification models. The networks were trained using the conventional training approach. IEEE 14-bus system, detection window size $\boldsymbol{w = 20}$, and attack rate $\boldsymbol{p = 25\%}$.

c	Weak			Medium			Strong		
	REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
L	76.94%	99.35%	99.50%	81.10%	100.00%	100.00%	89.73%	100.00%	100.00%
T	88.45%	98.88%	99.00%	98.45%	98.67%	98.68%	99.73%	100.00%	100.00%
TL	85.07%	99.84%	99.86%	95.71%	99.76%	99.77%	99.77%	100.00%	100.00%
LT	76.48%	100.00%	100.00%	94.20%	98.52%	98.58%	98.86%	100.00%	100.00%

Table 4: FDIA detection results of L, T, TL, and LT time series classification models. The networks were trained using the conventional training approach. IEEE 30-bus system, detection window size $\boldsymbol{w = 20}$, and attack rate $\boldsymbol{p = 25\%}$.

c	Weak			Medium			Strong		
	REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
L	95.98%	100.00%	100.00%	94.47%	100.00%	100.00%	97.76%	100.00%	100.00%
T	97.85%	100.00%	100.00%	97.85%	100.00%	100.00%	99.63%	99.54%	99.54%
TL	95.75%	100.00%	100.00%	98.58%	99.68%	99.68%	99.45%	100.00%	100.00%
LT	98.04%	100.00%	100.00%	98.40%	100.00%	99.09%	99.41%	100.00%	100.00%

The Transformer model showed consistently high REC, PR, and TNR across all the attack categories when compared to the other models. From Tables 3 and 4, it can be observed that the detection rate, REC, of the Transformer for weak, medium, and strong attacks on the IEEE 14-bus were 88.45%, 98.45%, and 99.73% respectively, and for the same attack categories on the IEEE 30-bus system, it was 97.85%, 97.85%, and 99.63% respectively. The PR of the Transformer was no less than 98.67%, and TNR was no less than 98.68% for any attack category considering both grid systems. The results show that the Transformer outperforms the other models. However, despite having the highest detection rate, the Transformer model was still unable to detect a significant number of instances that consisted of some weak attacks when trained with the conventional approach. It failed to detect 11.55% and 2.15% of the weak attack instances on the IEEE 14-bus and IEEE 30-bus, respectively. These results indicate that an

attacker can bypass the BDD and the classifier both with a high success rate by injecting weak attacks, and therefore, the detection rate needs further improvement.

Next, we present experimental evidence to show that there is a correlation between the number of attacks within the detection window λ and the detection rate R_λ, i.e., REC. Figure 10 shows the detection rate of the Transformer classifier for weak attacks at different λ, trained on a dataset of $p = 100\%$ which is similar to the training process of most existing approaches that include [24], [27], and [28]. Weak attacks on the IEEE 14-bus were considered for this analysis with a window size w of 20. The model was tested on a dataset in which the number of samples for different λ was uniformly distributed. It was observed that the detection rate R_λ decreased as the λ decreased. The drastic drop in the detection rate for small λ values emphasizes the necessity to create more efficient training approaches. These results are also an indication of the vulnerability of the recent approaches such as [24] to sparse attacks of small magnitudes injected at low rates.

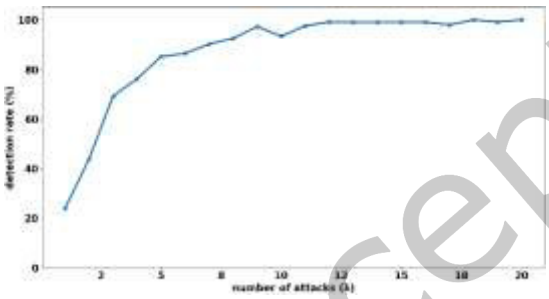


Figure 10: The number of attacks within the detection window $\boldsymbol{\lambda}$ vs the detection rate $\boldsymbol{R_\lambda}$ of the Transformer classifier trained with a $\boldsymbol{p = 100\%}$ dataset for weak attacks when $\boldsymbol{w = 20}$.

The following presents results pertaining to the proposed classifier training method in Algorithm 1. For the three detection window sizes $w = 10$, $w = 20$, and $w = 30$, the probability $\hat{p}$ of $G(\lambda)$ obtained using Algorithm 1 were 8%, 10%, and 10% for the IEEE 14-bus, and 14%, 16%, and 16% for the IEEE 30-bus, respectively. Figure 11 shows the distribution $G(\lambda)$ of the number of attacks within the detection window, among the attacked samples in the training dataset. The figure includes the distributions of the three window sizes, considering the IEEE 14-bus. In the training datasets created using the proposed method, the number of attacks within the detection window λ of the attacked samples follows the distribution in Figure 11, corresponding to the window size.

We evaluated the detection rate of the Transformer classifier trained using the proposed training method in Algorithm 1 across four different attack rates: $p = 100\%$, $p = 75\%$, $p = 50\%$, and $p = 25\%$. The detection window size of 20 was used in this experiment, and weak attacks were considered. To illustrate the effectiveness of the proposed training method, we trained the same classifier following the conventional approach using four different datasets corresponding to the aforementioned attack rates. Tables 5 and 6 show the detection rate, i.e., REC, of the Transformer classifier trained with the proposed method and the conventional approach with datasets of different attack rates, on IEEE 14-bus and IEEE 30-bus, respectively. It was observed that a model trained using a low attack rate dataset provides a high

detection rate on high attack rates as well. Interestingly, in all the experiments, given an attack rate p, a higher detection rate was observed when the classifier was trained on a dataset of an attack rate less than p compared to the classifier trained on a dataset of attack rate p or higher. These experimental results support the arguments in Section 3.1. Furthermore, the proposed training method had the highest detection rate for all four attack rates, proving that a classifier was trained to efficiently detect attacks across all the attack rates using the proposed method. Moreover, the detection rate of the classifier trained using the $p = 100\%$ dataset, which corresponds to the training process used in existing approaches, drastically dropped at low attack rates which is an alarming concern. A PR of 99.70% and a TNR of 99.73% for the IEEE 14-bus, and a PR of 99.09% and a TNR of 99.09% for the IEEE 30-bus, were observed in this experiment with the proposed training method, and they were comparable with the other training datasets.

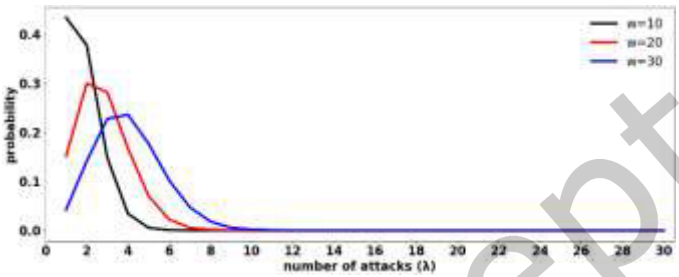


Figure 11: Distribution $\boldsymbol{G(\lambda)}$ of the number of attacks within the window $\boldsymbol{\lambda}$ in the proposed training method for different window sizes.

Table 5: The detection rate (**REC**) of the Transformer classifier using the proposed training method and the conventional training approach with datasets of different attack rates. IEEE 14-bus system, weak attacks, and detection window size $\boldsymbol{w = 20}$.

		Training Dataset				
		$p = 100\%$	$p = 75\%$	$p = 50\%$	$p = 25\%$	proposed
Test Dataset	$p = 25\%$	66.16%	75.25%	82.97%	88.45%	92.24%
	$p = 50\%$	91.42%	94.84%	98.68%	100.00%	100.00%
	$p = 75\%$	98.68%	99.22%	99.95%	100.00%	100.00%
	$p = 100\%$	100.00%	100.00%	100.00%	100.00%	100.00%

Table 6: The detection rate (**REC**) of the Transformer classifier using the proposed training method and the conventional training approach with datasets of different attack rates. IEEE 30-bus system, weak attacks, and detection window size $\boldsymbol{w = 20}$.

		Training Dataset				
		$p = 100\%$	$p = 75\%$	$p = 50\%$	$p = 25\%$	proposed
Test Dataset	$p = 25\%$	83.52%	88.13%	93.88%	97.85%	99.36%
	$p = 50\%$	98.63%	98.95%	99.45%	99.63%	99.73%
	$p = 75\%$	99.22%	99.54%	100.00%	100.00%	100.00%
	$p = 100\%$	100.00%	100.00%	100.00%	100.00%	100.00%

Figure 12 shows the detection rate R_λ, i.e., REC, of the Transformer classifier at different numbers of attacks within the detection window λ, trained with the proposed method in Algorithm 1, for weak attacks on the IEEE 14-bus. The detection rates of the smaller λ values were improved significantly compared to the detection rates shown in Figure 10 of the classifier trained with a $p = 100\%$ dataset. Specifically, for $\lambda = 1$ and $\lambda = 2$, improvements of 28% and 33%, respectively, were observed. With the proposed training method, the Transformer classifier achieved a 100% detection rate for $\lambda \geq 6$ while the same classifier trained with the $p = 100\%$ dataset underperformed for most of the values of λ.

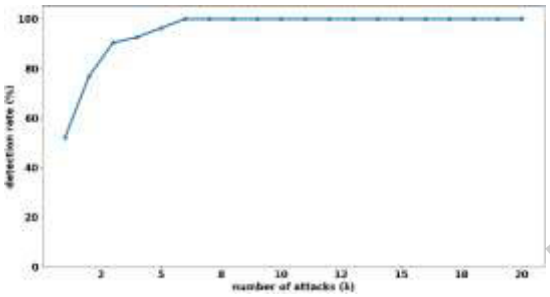


Figure 12: The number of attacks within the window $\boldsymbol{\lambda}$ vs the detection rate $\boldsymbol{R_\lambda}$ of the Transformer classifier with the proposed training method for weak attacks when $\boldsymbol{w = 20}$.

Tables 7 and 8 show the detailed results of the Transformer classifier trained using the proposed training method for the IEEE 14-bus and the IEEE 30-bus, respectively. Results were obtained for the three different window sizes considering four attack rates for the three attack magnitudes. It was observed that, in both the IEEE bus systems, for a given window size w, the detection rate, i.e., REC, decreased as the attack magnitude or the attack rate p decreased. Specifically, the classifier had a higher failure rate when weak attacks were injected at a low rate. The detection rate, REC, of the Transformer classifier for the weak attacks was 80.31% on the IEEE 14-bus and 90.65% on the IEEE 30-bus when the attack rate p was 25% and window size w was 10. The detection rate increased as the window size increased. However, this increment in the detection rate comes at the cost of diagnostic resolution. The PR and TNR of the Transformer classifier were above 99% with the proposed training method.

4.2 Time Series Prediction-based FDIA Detection

Four DNN model architectures shown in Table 2 were experimented with for the predictor model selection. Tables 9 and 10 show the FDIA detection results of different predictor model architectures for the IEEE 14-bus system and the IEEE 30-bus system, respectively. The performance of various DNN models was evaluated by considering different methods in $\boldsymbol{C_P}$, for the three attack magnitudes. In this experiment, predictors were trained with the configuration M described in Section 2.2, the attack rate p was 25%, and the detection window size w was 20. In both the IEEE bus systems, the TL model had the highest detection rate, i.e., REC, for a given implementation of the $\boldsymbol{C_P}$, except for the TL with *2-Norm* on the IEEE 30-bus where the detection rate was marginally lower than that of the LSTM. Among the four different classification methods of the $\boldsymbol{C_P}$, the *FCNN* had the highest REC and the *2-Norm* had the highest PR and TNR. However, the *FCNN* had a very low PR and TNR, while the *2-Norm* had the lowest REC. The *Thresh* approach provided a good balance among the REC, PR, and TNR, and it was chosen for the $\boldsymbol{C_P}$ of the

proposed predictor-based FDIA detector. Furthermore, we observed that the TL model had the lowest prediction Root Mean Squared Error (RMSE) and the Transformer had the highest prediction RMSE in most of the measurements. A lower RMSE on benign samples and a higher RMSE on attacked samples are favorable for achieving high accuracy. However, it was observed that in practice the TL model, which had the lowest RMSE on benign, provides a higher detection rate, REC, with a high PR and TNR.

Table 7: FDIA detection results of the Transformer classifier trained using the proposed training method on the IEEE 14-bus system.

w	p	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
10	25%	80.31%	99.70%	99.77%	96.30%	99.36%	99.41%	99.86%	100.00%	100.00%
	50%	96.80%	99.76%	99.77%	99.68%	99.41%	99.41%	100.00%	100.00%	100.00%
	75%	99.50%	99.77%	99.77%	100.00%	99.41%	99.41%	100.00%	100.00%	100.00%
	100%	100.00%	99.77%	99.77%	100.00%	99.41%	99.41%	100.00%	100.00%	100.00%
20	25%	92.24%	99.70%	99.73%	99.59%	99.45%	99.45%	99.95%	100.00%	100.00%
	50%	100.00%	99.73%	99.73%	100.00%	99.46%	99.45%	100.00%	100.00%	100.00%
	75%	100.00%	99.73%	99.73%	100.00%	99.46%	99.45%	100.00%	100.00%	100.00%
	100%	100.00%	99.73%	99.73%	100.00%	99.46%	99.45%	100.00%	100.00%	100.00%
30	25%	96.07%	99.86%	99.86%	100.00%	99.77%	99.77%	100.00%	100.00%	100.00%
	50%	100.00%	99.86%	99.86%	100.00%	99.77%	99.77%	100.00%	100.00%	100.00%
	75%	100.00%	99.86%	99.86%	100.00%	99.77%	99.77%	100.00%	100.00%	100.00%
	100%	100.00%	99.86%	99.86%	100.00%	99.77%	99.77%	100.00%	100.00%	100.00%

Table 8: FDIA detection results of the Transformer classifier trained using the proposed training method on the IEEE 30-bus system.

w	p	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
10	25%	90.65%	99.48%	99.54%	96.25%	99.02%	99.09%	98.96%	99.81%	99.82%
	50%	98.08%	99.54%	99.54%	99.86%	99.09%	99.09%	99.95%	99.82%	99.82%
	75%	99.95%	99.55%	99.54%	100.00%	99.10%	99.09%	100.00%	99.82%	99.82%
	100%	100.00%	99.55%	99.54%	100.00%	99.10%	99.09%	100.00%	99.82%	99.82%
20	25%	99.36%	99.09%	99.09%	99.09%	99.09%	99.09%	99.73%	100.00%	100.00%
	50%	99.73%	99.09%	99.09%	100.00%	99.10%	99.09%	100.00%	100.00%	100.00%
	75%	100.00%	99.10%	99.09%	100.00%	99.10%	99.09%	100.00%	100.00%	100.00%
	100%	100.00%	99.10%	99.09%	100.00%	99.10%	99.09%	100.00%	100.00%	100.00%
30	25%	99.91%	99.86%	99.86%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50%	100.00%	99.86%	99.86%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	75%	100.00%	99.86%	99.86%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100%	100.00%	99.86%	99.86%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

The TL model with the *2-Norm* criterion corresponds to the method proposed in [28]. It was observed that the detection rate, REC, of the method proposed in [28] drops down to 11.74%, 41.69%, and 74.43% on the IEEE 14-bus, and to 6.67%, 15.62%, and 72.47% on the IEEE 30-bus, for weak, medium, and strong attacks, respectively, when the attack rate p was 25%. This indicates the vulnerability of the predictor-based FDIA detectors exist in the literature.

Table 9: FDIA detection results of L, T, TL, and LT time series prediction models with four different prediction error vector classification methods. IEEE 14-bus system, detection window size $w = 20$, and attack rate $p = 25\%$.

$\mathcal{P}$	$\mathcal{C}_{\mathcal{P}}$	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
L	*Thresh*	18.68%	91.91%	98.36%	34.29%	95.43%	98.36%	52.24%	96.95%	98.36%
	2-Norm	8.81%	100.00%	100.00%	20.78%	100.00%	100.00%	41.55%	100.00%	100.00%
	FCNN	42.97%	68.79%	80.50%	52.92%	80.60%	87.26%	50.14%	91.46%	99.22%
	KNN	29.13%	76.50%	91.05%	38.81%	91.50%	96.39%	65.07%	98.48%	93.93%
T	*Thresh*	18.68%	78.96%	95.02%	37.44%	88.27%	95.02%	65.57%	92.94%	95.02%
	2-Norm	7.08%	100.00%	100.00%	19.27%	100.00%	100.00%	49.27%	100.00%	100.00%
	FCNN	37.53%	69.02%	83.15%	54.79%	80.59%	86.80%	76.39%	94.73%	95.75%
	KNN	25.11%	71.80%	90.14%	40.73%	91.30%	96.12%	66.53%	99.18%	99.45%
TL	*Thresh*	24.98%	93.83%	98.36%	56.26%	97.16%	98.36%	81.32%	98.02%	98.36%
	2-Norm	11.74%	100.00%	100.00%	41.69%	100.00%	100.00%	74.43%	100.00%	100.00%
	FCNN	47.35%	78.86%	86.85%	67.12%	94.03%	94.89%	77.72%	99.33%	99.91%
	KNN	42.01%	78.57%	88.54%	59.73%	97.03%	98.17%	80.87%	99.88%	99.04%
LT	*Thresh*	13.74%	82.02%	96.99%	22.79%	88.32%	96.99%	34.57%	91.98%	96.99%
	2-Norm	5.07%	100.00%	100.00%	8.40%	100.00%	100.00%	23.88%	100.00%	100.00%
	FCNN	35.75%	61.75%	77.85%	39.36%	76.15%	87.67%	52.42%	81.88%	88.40%
	KNN	24.20%	62.65%	85.57%	28.40%	76.23%	91.14%	40.64%	92.13%	96.53%

Table 10: FDIA detection results of L, T, TL, and LT time series prediction models with four different prediction error vector classification methods. IEEE 30-bus system, detection window size $w = 20$, and attack rate $p = 25\%$.

$\mathcal{P}$	$\mathcal{C}_{\mathcal{P}}$	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
L	*Thresh*	24.47%	90.85%	97.53%	34.75%	93.37%	97.53%	65.98%	96.40%	97.53%
	2-Norm	7.31%	96.97%	99.77%	13.70%	98.36%	99.77%	33.01%	99.31%	99.77%
	FCNN	40.32%	66.69%	83.93%	55.53%	79.09%	86.53%	76.39%	95.27%	97.44%
	KNN	22.47%	88.97%	97.21%	30.55%	96.68%	98.95%	55.30%	99.43%	99.68%
T	*Thresh*	29.59%	93.37%	97.90%	42.60%	95.30%	97.90%	76.94%	97.34%	97.90%
	2-Norm	6.21%	99.27%	99.95%	15.43%	99.71%	99.95%	41.74%	99.89%	99.95%
	FCNN	47.95%	76.14%	86.71%	60.23%	80.59%	88.95%	85.21%	97.33%	97.85%
	KNN	34.20%	89.81%	96.12%	46.44%	96.12%	98.13%	77.08%	99.18%	99.36%
TL	*Thresh*	43.29%	92.91%	96.89%	59.95%	95.08%	96.89%	97.63%	96.92%	96.89%
	2-Norm	6.67%	100.00%	100.00%	15.62%	100.00%	100.00%	72.47%	100.00%	100.00%
	FCNN	60.55%	79.43%	85.62%	68.49%	93.74%	93.97%	95.57%	99.48%	99.54%
	KNN	41.96%	90.62%	97.21%	48.22%	99.06%	99.54%	89.13%	99.90%	99.91%
LT	*Thresh*	22.51%	90.96%	97.76%	27.35%	92.44%	97.76%	43.79%	95.14%	97.76%
	2-Norm	5.66%	98.41%	99.91%	11.64%	99.22%	99.91%	23.52%	99.61%	99.91%
	FCNN	46.67%	58.74%	68.58%	37.26%	68.48%	88.22%	68.77%	89.59%	86.67%
	KNN	25.39%	76.37%	92.15%	32.83%	76.49%	89.91%	50.82%	96.28%	98.04%

Next, we assessed the performance of the three predictor configurations M, MS, and S described in Section 2.2. We experimented with the L, T, TL, and LT architectures for all three predictor configurations using the *Thresh* method for the $\mathcal{C}_{\mathcal{P}}$. The TL model achieved the highest detection rate for multivariate predictor configurations M and MS, while the L model achieved the highest detection rate for the univariate predictor configuration S. FDIA detection results of the two IEEE bus systems are shown in

Tables 11 and 12 (The PR was excluded in the tables due to the space constraints). In both IEEE 14-bus and IEEE 30-bus, the univariate model had the highest detection rate, i.e., REC, because of its higher sensitivity to the FDIAs as explained in Section 3.2. For a given window size w, the detection rate decreased as the attack magnitude or the attack rate p decreased. A relatively higher detection rate, REC, was observed on the IEEE-30 bus compared to the IEEE-14 bus. This was because the number of measurements m in the IEEE 30-bus was higher than that of the IEEE 14-bus. The higher the number of measurements, the higher the chance of detecting attacks. However, for the same reason, the TNR of the IEEE 30-bus was relatively lower than that of the IEEE-14 bus. The PR of each experiment in Tables 11 and 12 was comparable to its TNR.

Table 11: Comparison of the FDIA detection results of multivariate and univariate predictors on the IEEE 14-bus system.

w	p	Multivariate Multi-prediction (M) - TL				Multivariate Single-prediction (MS) - TL				Univariate Single-prediction (S) - L			
		REC			TNR	REC			TNR	REC			TNR
		Weak	Medium	Strong		Weak	Medium	Strong		Weak	Medium	Strong	
10	25%	23.25%	47.07%	71.66%	99.00%	34.53%	58.69%	80.88%	98.95%	64.02%	66.47%	77.62%	99.50%
	50%	43.97%	69.70%	90.40%	99.00%	58.82%	80.85%	95.89%	98.95%	88.80%	86.06%	94.65%	99.50%
	75%	61.32%	88.36%	98.04%	99.00%	78.08%	95.25%	99.86%	98.95%	97.53%	97.81%	99.04%	99.50%
	100%	78.45%	97.35%	100.00%	99.00%	90.18%	99.27%	100.00%	98.95%	99.13%	99.45%	99.91%	99.50%
20	25%	24.98%	56.26%	81.32%	98.36%	47.31%	70.50%	88.81%	98.40%	82.28%	79.32%	83.65%	98.95%
	50%	48.72%	83.01%	95.21%	98.36%	76.99%	92.60%	98.54%	98.40%	98.72%	96.07%	97.58%	98.95%
	75%	64.52%	95.21%	99.54%	98.36%	90.96%	98.86%	99.95%	98.40%	100.00%	99.59%	99.86%	98.95%
	100%	80.82%	99.50%	100.00%	98.36%	96.99%	100.00%	100.00%	98.40%	100.00%	99.91%	100.00%	98.95%
30	25%	23.06%	63.38%	84.16%	99.27%	67.95%	85.34%	96.03%	98.40%	91.46%	89.32%	85.75%	98.40%
	50%	44.11%	86.03%	96.67%	99.27%	89.50%	97.35%	99.77%	98.40%	99.36%	99.50%	98.17%	98.40%
	75%	63.52%	96.16%	99.45%	99.27%	97.72%	99.95%	100.00%	98.40%	100.00%	99.86%	99.82%	98.40%
	100%	77.44%	99.18%	100.00%	99.27%	99.63%	100.00%	100.00%	98.40%	100.00%	100.00%	99.91%	98.40%

The TL model had the lowest RMSE for the multivariate predictor configurations M and MS, and the LSTM model had the lowest RMSE for the univariate predictor configuration S. However, the RMSE of the univariate LSTM model was comparatively higher than that of the multivariate TL models. Figure 13 compares the RMSE of the multivariate and the univariate predictors for benign data considering six measurements in the IEEE 14-bus system. The RMSE of the univariate LSTM was still within an acceptable range so the predictions were very close to the actual measurements for benign. Figure 14 shows the output of the univariate LSTM predictor for the real power injection of bus 4 and for the real power injected to line 7 at bus 4 on the IEEE 14-bus considering 100 timesteps. The actual measurements z_i^t is plotted with a solid black line, and the predicted measurement $\hat{z}_i^t$ is plotted with a dotted blue line. The figure shows that both the power injections and the power flow predictions are very close to the actual. Similarly, all the measurements were predicted with a small error. Detailed results of the different predictions are omitted due to space limitations.

In conclusion, our results showed that a higher detection rate can be achieved using a univariate predictor with the *Thresh* method in the $\mathcal{C}_{\mathcal{P}}$ module. However, it was observed that the detection rate

achieved by the predictor networks was inferior to the detection rate of the Transformer classifier network in Section 4.1. Especially a significant reduction in the detection rate was observed in medium and strong attacks for low attack rates while the classifier had a near 100% detection rate.

Table 12: Comparison of the FDIA detection results of multivariate and univariate predictors on the IEEE 30-bus system.

w	p	Multivariate Multi-prediction (M) - TL				Multivariate Single-prediction (MS) - TL				Univariate Single-prediction (S) - L			
		REC			TNR	REC			TNR	REC			TNR
		Weak	Medium	Strong		Weak	Medium	Strong		Weak	Medium	Strong	
10	25%	32.89%	51.35%	86.71%	97.21%	66.07%	84.81%	97.06%	95.07%	91.55%	92.41%	95.02%	95.25%
	50%	56.29%	77.99%	97.35%	97.21%	88.47%	97.35%	99.73%	95.07%	98.44%	98.99%	99.82%	95.25%
	75%	74.70%	92.19%	99.86%	97.21%	97.72%	99.86%	100.00%	95.07%	99.95%	99.95%	99.95%	95.25%
	100%	90.00%	99.54%	100.00%	97.21%	99.77%	100.00%	100.00%	95.07%	100.00%	100.00%	100.00%	95.25%
20	25%	43.29%	59.95%	97.63%	96.89%	79.09%	95.16%	99.50%	95.34%	98.68%	98.13%	99.54%	95.11%
	50%	71.05%	88.36%	99.95%	96.89%	96.62%	99.82%	100.00%	95.34%	99.95%	100.00%	100.00%	95.11%
	75%	85.62%	97.12%	100.00%	96.89%	99.22%	100.00%	100.00%	95.34%	100.00%	100.00%	100.00%	95.11%
	100%	94.89%	99.95%	100.00%	96.89%	99.95%	100.00%	100.00%	95.34%	100.00%	100.00%	100.00%	95.11%
30	25%	51.87%	76.89%	99.77%	97.26%	88.77%	99.00%	100.00%	95.80%	99.91%	100.00%	100.00%	95.57%
	50%	73.79%	94.16%	100.00%	97.26%	99.32%	100.00%	100.00%	95.80%	100.00%	100.00%	100.00%	95.57%
	75%	86.07%	97.40%	100.00%	97.26%	99.91%	100.00%	100.00%	95.80%	100.00%	100.00%	100.00%	95.57%
	100%	95.39%	99.91%	100.00%	97.26%	100.00%	100.00%	100.00%	95.80%	100.00%	100.00%	100.00%	95.57%

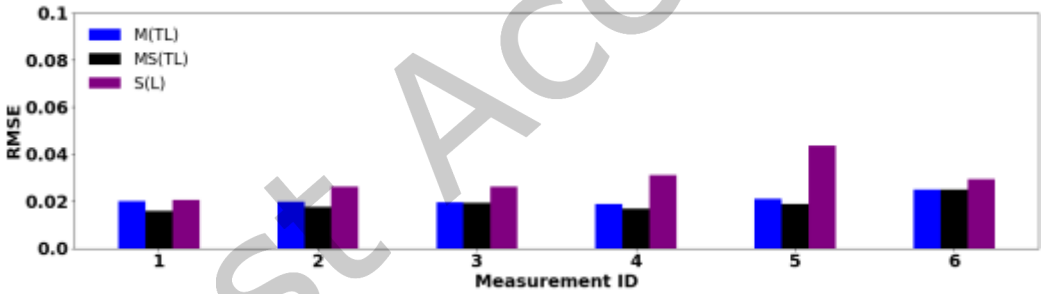
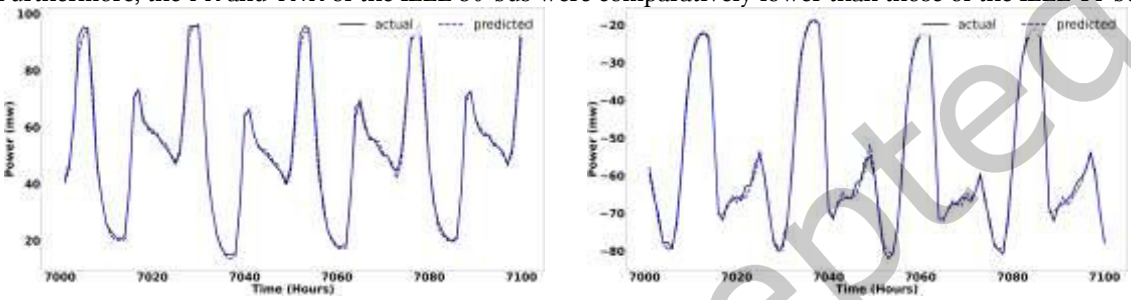


Figure 13: Comparison of the RMSE of the normalized predictions of the multivariate and the univariate predictors for six measurements in the IEEE 14-bus system.

4.3 The Proposed Enhanced FDIA Detection Framework Combining a Classifier and Predictors

The FDIA detection results of the proposed classifier and predictor combined framework were assessed by considering different detection window sizes, attack magnitudes, and attack rates. Results are shown in Tables 13 and 14 for the IEEE 14-bus system and the IEEE 30-bus system respectively. High REC, PR, and TNR were observed on both the IEEE 14-bus and IEEE 30-bus. It was observed that the detection rate, i.e., REC, of the hard-to-detect sparse FDIAs improved by combining the proposed time series classifier-based detector with the times series predictor-based detector. The values in bold indicate the improved detection

rates, REC values, by combining the two detection methods when compared to using only one detection method. A significant improvement was achieved in the detection rate of small-magnitude attacks injected at low rates on both IEEE bus systems. Since the classifier and predictor are operating independently, the number of true positives of the classifier and predictor combined approach is always no less than that of the individual methods. Therefore, the detection rate, REC, of the proposed framework was always equal or higher compared to using only one detection method. Similarly, the number of false positives of the classifier and predictor combined approach was always no less than that of the individual methods. Consequently, the TNR of the proposed framework was always equal or lower compared to using only one detection method. The PR of the classifier and predictor combined method was lower than the PR of the classifier-based detector, and marginally higher or lower than the PR of the predictor-based detector. Furthermore, the PR and TNR of the IEEE 30-bus were comparatively lower than those of the IEEE 14-bus.



(a) (b)

Figure 14: Prediction of (a) the power injection of bus 4 and (b) the real power injected to line 7 at bus 4 in the IEEE 14-bus system using the univariate LSTM.

Table 13: Detection results of the proposed enhanced FDIA detection framework that consists of a time series classifier and multiple time series predictors on the IEEE 14-bus system.

w	p	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
10	25%	**82.04%**	99.07%	99.27%	**96.59%**	98.82%	98.90%	99.86%	99.47%	99.50%
	50%	**97.03%**	99.25%	99.27%	**99.77%**	98.91%	98.90%	100.00%	99.50%	99.50%
	75%	**99.73%**	99.27%	99.27%	100.00%	98.92%	98.90%	100.00%	99.50%	99.50%
	100%	100.00%	99.27%	99.27%	100.00%	98.92%	98.90%	100.00%	99.50%	99.50%
20	25%	**93.01%**	98.60%	98.68%	99.59%	98.42%	98.40%	99.95%	98.96%	98.95%
	50%	100.00%	98.69%	98.68%	100.00%	98.43%	98.40%	100.00%	98.96%	98.95%
	75%	100.00%	98.69%	98.68%	100.00%	98.43%	98.40%	100.00%	98.96%	98.95%
	100%	100.00%	98.69%	98.68%	100.00%	98.43%	98.40%	100.00%	98.96%	98.95%
30	25%	**96.62%**	98.24%	98.26%	100.00%	98.21%	98.17%	100.00%	98.43%	98.40%
	50%	100.00%	98.29%	98.26%	100.00%	98.21%	98.17%	100.00%	98.43%	98.40%
	75%	100.00%	98.29%	98.26%	100.00%	98.21%	98.17%	100.00%	98.43%	98.40%
	100%	100.00%	98.29%	98.26%	100.00%	98.21%	98.17%	100.00%	98.43%	98.40%

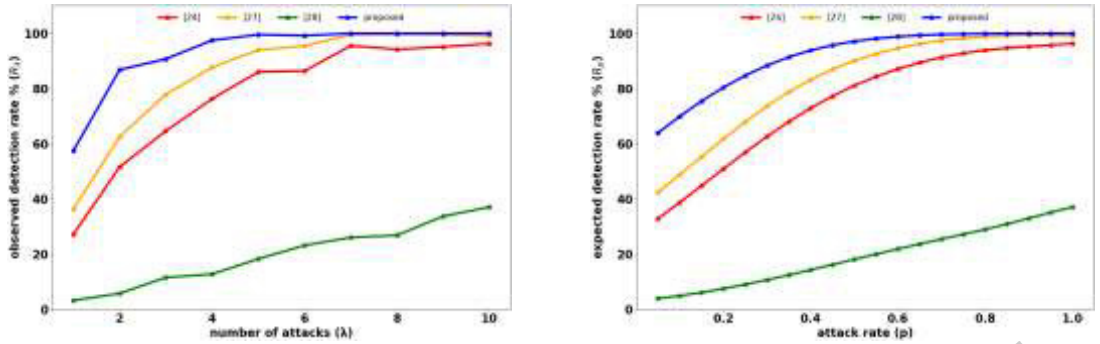
Table 14: Detection results of the proposed enhanced FDIA detection framework that consists of a time series classifier and multiple time series predictors on the IEEE 30-bus system.

w	p	Weak			Medium			Strong		
		REC	PR	TNR	REC	PR	TNR	REC	PR	TNR
10	25%	**93.17%**	94.65%	94.93%	**97.34%**	94.60%	94.66%	**99.34%**	95.27%	95.25%
	50%	**98.86%**	95.11%	94.93%	**99.91%**	94.91%	94.66%	**100.00%**	95.46%	95.25%
	75%	**99.95%**	95.17%	94.93%	100.00%	94.93%	94.66%	100.00%	95.47%	95.25%
	100%	100.00%	95.18%	94.93%	100.00%	94.93%	94.66%	100.00%	95.47%	95.25%
20	25%	**99.50%**	94.66%	94.38%	**99.36%**	94.65%	94.38%	**99.77%**	95.33%	95.11%
	50%	**99.95%**	94.68%	94.38%	100.00%	94.68%	94.38%	100.00%	95.34%	95.11%
	75%	100.00%	94.68%	94.38%	100.00%	94.68%	94.38%	100.00%	95.34%	95.11%
	100%	100.00%	94.68%	94.38%	100.00%	94.68%	94.38%	100.00%	95.34%	95.11%
30	25%	99.91%	95.63%	95.43%	100.00%	95.76%	95.57%	100.00%	95.76%	95.57%
	50%	100.00%	95.63%	95.43%	100.00%	95.76%	95.57%	100.00%	95.76%	95.57%
	75%	100.00%	95.63%	95.43%	100.00%	95.76%	95.57%	100.00%	95.76%	95.57%
	100%	100.00%	95.63%	95.43%	100.00%	95.76%	95.57%	100.00%	95.76%	95.57%

4.4 Evaluation of the FDIA Detection Rate Under Different Attack Rates

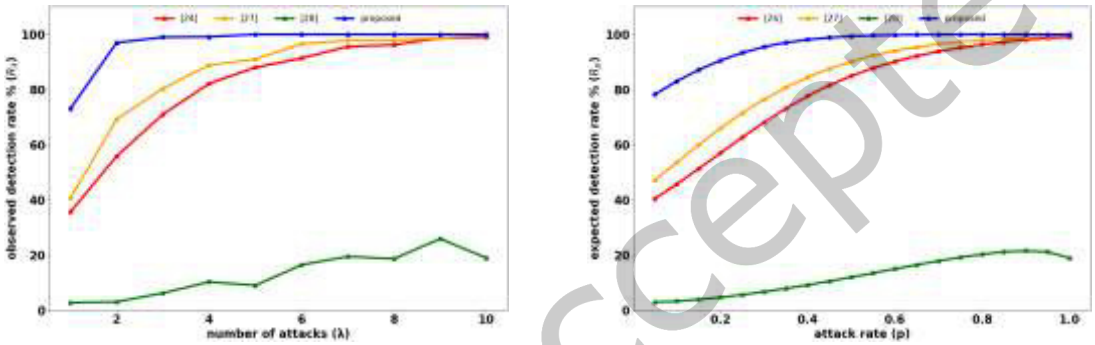
To illustrate the detection rate improvement by the proposed framework, the expected detection rate, i.e., REC, of the proposed FDIA detection framework and the methods in [24], [27], and [28] were assessed using the evaluation criterion in Section 3.4. Weak attacks with a window size w of 10 was considered in this evaluation. A testing dataset was created such that the number of samples for each λ was uniformly distributed, and then the observed detection rate R_λ was obtained for each λ. Figures 15(a) and 16(a) show the R_λ at different λ values for the two IEEE bus systems. Thereafter, the expected detection rates R_p were computed for twenty different attack rates p, where $5\% \leq p \leq 100\%$, using Equation (11). Figures 15(b) and 16(b) show the R_p at different p values for the two IEEE bus systems. The expected detection rates R_p estimated using Equation (11) were very close to those experimentally observed. For instance, the estimated R_p of the proposed method for $p = 25\%$, $p = 50\%$, $p = 75\%$, and $p = 100\%$ were, 84.70%, 97.11%, 99.78%, and 100% on the IEEE 14-bus, and 93.32%, 99.30%, 99.98%, and 100% on the IEEE 30-bus, respectively. These estimated R_p values were comparable to the corresponding REC values reported in Tables 13 and 14. Figures 15(b) and 16(b) show that the expected detection rate R_p of the existing methods in the literature decreased drastically at low attack rates. The proposed method improved the FDIA detection rate at low attack rates significantly compared to the existing methods.

Figure 17 shows the increase in the expected detection rate R_p, i.e., REC, of the proposed framework compared to the method in [27], for weak attacks at different attack rates for the three window sizes considered in the experiments. A significant improvement in the detection rate was observed at low attack rates with the proposed framework for all three window sizes. The detection rate improved approximately by 22%, 31%, and 39% on the IEEE 14-bus, and by 31%, 45%, and 48% on the IEEE 30-bus, for the window sizes 10, 20, and 30, respectively, when the attack rate p was 5%. Our previously proposed method in [26] reported a 99.63% detection rate for $w = 30$ when $p = 20\%$ in the IEEE 14-bus, but the TNR was 89.10%. In comparison, the proposed method improved the false positive rate by 9.2% with a 3.5% reduction in the detection rate which is a desirable trade-off.



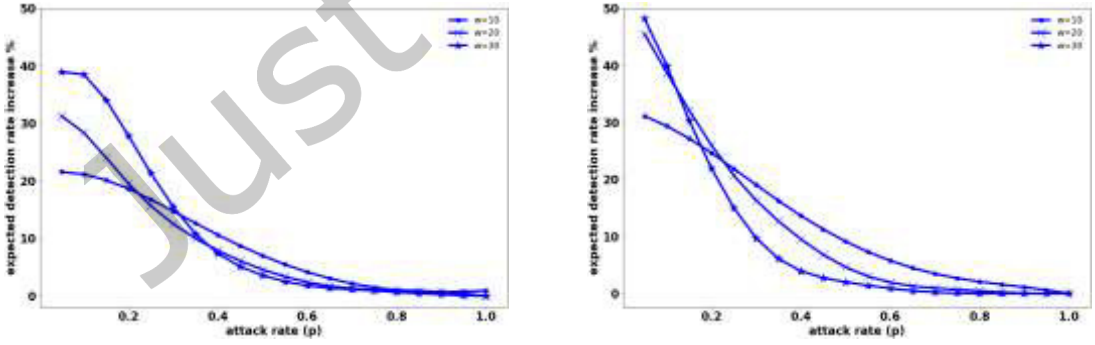
(a) (b)

Figure 15: (a) The change in the detection rate $\boldsymbol{R_\lambda}$ as the number of attacks within the detection window changes and (b) the change in the expected detection rate, $\boldsymbol{R_p}$, as the attack rate changes, for weak attacks in the IEEE 14-bus system when $\boldsymbol{w = 10}$.



(a) (b)

Figure 16: (a) The change in the detection rate $\boldsymbol{R_\lambda}$ as the number of attacks within the detection window changes and (b) the change in the expected detection rate, $\boldsymbol{R_p}$, as the attack rate changes, for weak attacks in the IEEE 30-bus system when $\boldsymbol{w = 10}$.



(a) IEEE 14-bus (b) IEEE 30-bus

Figure 17: The improvement of the expected detection rate $\boldsymbol{R_p}$ of the proposed framework compared to the method in [27] under different attack rates and different detection window sizes for weak attacks.

5 CONCLUSIONS

This study investigated the detection of the FDIAs on the power grid state estimation using time series analysis DNNs under different attack rates and addressed various associated challenges. It was shown that existing time series analysis DNNs are highly vulnerable to FDIAs executed at low attack rates. Various alternative implementations are presented for time series classifiers and time series predictors to improve the FDIA detection rate. A novel method is proposed to train time series classification neural networks to detect FDIAs of any attack rate with high efficiency. We employed the proposed time series analysis DNN-based approaches to design an enhanced FDIA detection framework that included a time series classifier and multiple predictors. Furthermore, an analytical criterion is derived to estimate the FDIA detection rate of time series analysis DNNs under any attack rate. Experimental evaluations were done using state-of-the-art DNN architectures considering the IEEE 14-bus system and the IEEE 30-bus system. Several new findings were presented. It was demonstrated that sparse attacks injected at low rates are much more difficult to detect with existing time series analysis DNNs. Among different time series classifier DNNs, a Transformer-based model had the best detection performance. It was shown that the univariate predictors have a significantly higher detection rate than the multivariate predictors. Among the time series predictor DNNs, a univariate LSTM model had the best detection performance. Furthermore, time series classification DNNs had comparatively higher detection rates than prediction-based DNNs. The proposed enhanced FDIA detection framework achieved a significantly higher detection rate on FDIAs of low attack rates when compared to the existing methods. The proposed framework detected attacks injected at any rate with high efficiency.

Even though the proposed method significantly improved the FDIA detection rate, experimental results show that the detection rate still needs improvement. For instance, when the attack rate was below 40%, a considerable reduction in the expected detection rate was observed. Enhanced FDIA detection methods may need to be developed to further improve for such low attack rates.

REFERENCES

[1] Kesler, Brent. "The vulnerability of nuclear facilities to cyber attack; strategic insights: Spring 2010." *Strategic Insights, Spring 2011* (2011).

[2] Case, Defense Use. "Analysis of the cyber attack on the Ukrainian power grid." *Electricity Information Sharing and Analysis Center (E-ISAC)* 388 (2016): 1-29.

[3] J. Condliffe, Ukraine's Power Grid Gets Hacked Again, a Worrying Sign for Infrastructure Attacks, MIT Technol. Rev., Cambridge, MA, USA, 2016.

[4] Krause, Tim, Raphael Ernst, Benedikt Klaer, Immanuel Hacker, and Martin Henze. "Cybersecurity in power grids: Challenges and opportunities." *Sensors* 21, no. 18 (2021): 6225.

[5] Eder-Neuhauser, Peter, Tanja Zseby, Joachim Fabini, and Gernot Vormayr. "Cyber attack models for smart grid environments." *Sustainable Energy, Grids and Networks* 12 (2017): 10-29.

[6] Liu, Yao, Peng Ning, and Michael K. Reiter. "False data injection attacks against state estimation in electric power grids." *ACM Transactions on Information and System Security (TISSEC)* 14 (renamed to *ACM Transactions on Privacy and Security (TOPS)* in 2016), no. 1 (2011): 1-33.

[7] Huseinović, Alvin, Saša Mrdović, Kemal Bicakci, and Suleyman Uludag. "A survey of denial-of-service attacks and solutions in the smart grid." *IEEE Access* 8 (2020): 177447-177470.

[8] Wlazlo, Patrick, Abhijeet Sahu, Zeyu Mao, Hao Huang, Ana Goulart, Katherine Davis, and Saman Zonouz. "Man-in-the-middle attacks and defence in a power system cyber-physical testbed." *IET Cyber-Physical Systems: Theory & Applications* 6, no. 3 (2021): 164-177.

[9] Alanazi, Manar, Abdun Mahmood, and Mohammad Jabed Morshed Chowdhury. "SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues." *Computers & security* 125 (2023): 103028.

[10] Wood, Allen J., Bruce F. Wollenberg, and Gerald B. Sheblé. *Power generation, operation, and control*. John Wiley & Sons, 2013.

[11] Ghiasi, Mohammad, Taher Niknam, Zhanle Wang, Mehran Mehrandezh, Moslem Dehghani, and Noradin Ghadimi. "A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future." *Electric Power Systems Research* 215 (2023): 108975.

[12] Musleh, Ahmed S., Guo Chen, and Zhao Yang Dong. "A survey on the detection algorithms for false data injection attacks in smart grids." *IEEE Transactions on Smart Grid* 11, no. 3 (2019): 2218-2234.

[13] G. Liang, J. Zhao, F. Luo, S. R. Weller and Z. Y. Dong, "A Review of False Data Injection Attacks Against Modern Power Systems," in *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1630-1638, July 2017.

[14] Y. Ding and J. Liu, "Real-time false data injection attack detection in energy internet using online robust principal component analysis," *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, China, 2017.

[15] M. Zanetti, E. Jamhour, M. Pellenz, M. Penna, V. Zambenedetti and I. Chueiri, "A Tunable Fraud Detection System for Advanced Metering Infrastructure Using Short-Lived Patterns," in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 830-840, Jan. 2019.

[16] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar and S. Mishra, "Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid," in *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005-1016, June 2016.

[17] E. M. Ferragut, J. Laska, M. M. Olama and O. Ozmen, "Real-Time Cyber-Physical False Data Attack Detection in Smart Grids Using Neural Networks," *2017 International Conf. on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2017, pp. 1-6.

[18] Wang, Defu, Xiaojuan Wang, Yong Zhang, and Lei Jin. "Detection of power grid disturbances and cyber-attacks based on machine learning." *Journal of information security and applications* 46 (2019): 42-52.

[19] O. Boyaci *et al.*, "Graph Neural Networks Based Detection of Stealth False Data Injection Attacks in Smart Grids," in *IEEE Systems Journal*, vol. 16, no. 2, pp. 2946-2957, June 2022.

[20] Li, Xueping, Yaokun Wang, and Zhigang Lu. "Graph-based detection for false data injection attacks in power grid." *Energy* 263 (2023): 125865.

[21] A. Ayad, M. Khalaf and E. El-Saadany, "Detection of False Data Injection Attacks in Automatic Generation Control Systems Considering System Nonlinearities," 2018 IEEE Electrical Power and Energy Conference (EPEC), Toronto, ON, Canada, 2018, pp. 1-6.

[22] Ayad, Abdelrahman, Hany EZ Farag, Amr Youssef, and Ehab F. El-Saadany. "Detection of false data injection attacks in smart grids using recurrent neural networks." In *2018 IEEE power & energy society innovative smart grid technologies conference (ISGT)*, pp. 1-5.

[23] James, J. Q., Yunhe Hou, and Victor OK Li. "Online false data injection attack detection with wavelet transform and deep neural networks." *IEEE Transactions on Industrial Informatics* 14, no. 7 (2018): 3271-3280.

[24] Zhang, Feiye, and Qingyu Yang. "False data injection attack detection in dynamic power grid: A recurrent neural network-based method." *Frontiers in Energy Research* 10 (2022): 1005660.

[25] Naderi, Ehsan, and Arash Asrari. "Toward detecting cyberattacks targeting modern power grids: a deep learning framework." In *2022 IEEE World AI IoT Congress (AIIoT)*, pp. 357-363. IEEE, 2022.

[26] D. Senarathna, S. Tragoudas, J. Wibbenmeyer and N. Khdeer, "Increasing Detection Rate of False Data Injection Attacks Using Measurement Predictors," *2023 IEEE 11th International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, Canada, 2023, pp. 148-152.

[27] Li, Yang, Xinhao Wei, Yuanzheng Li, Zhaoyang Dong, and Mohammad Shahidehpour. "Detection of false data injection attacks in smart grid: A secure federated deep learning approach." *IEEE Transactions on Smart Grid* 13, no. 6 (2022): 4862-4872.

[28] Baul, Anik, Gobinda Chandra Sarker, Pintu Kumar Sadhu, Venkata P. Yanambaka, and Ahmed Abdelgawad. "XTM: A Novel Transformer and LSTM-Based Model for Detection and Localization of Formally Verified FDI Attack in Smart Grid." *Electronics* 12, no. 4 (2023): 797.

[29] Yang, Hang, and Jiayi Jin. "A Transformer and Cnn-Based Hybrid Model for Localization Detection of False Data Injection Attacks in Smart Grids."

[30] Zu, Tong, and Fengyong Li. "Self-Attention Spatio-Temporal Deep Collaborative Network for Robust FDIA Detection in Smart Grids." *CMES-Computer Modeling in Engineering & Sciences* 141, no. 2 (2024).

[31] Li, Xueping, Linbo Hu, and Zhigang Lu. "Detection of false data injection attack in power grid based on spatial-temporal transformer network." *Expert Systems with Applications* 238 (2024): 121706.

[32] Aggarwal, Charu C. "Neural networks and deep learning." *Springer* 10, no. 978 (2018): 3.

[33] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6.

[34] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The Graph Neural Network Model," in *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61-80, Jan. 2009.

[35] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena* 404 (2020): 132306.

[36] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

[37] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997 Nov 15;9(8):1735-80.

[38] Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang; A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput* 2019; 31 (7): 1235–1270.

[39] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[40] Wen, Qingsong, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. "Transformers in time series: A survey." *arXiv preprint arXiv:2202.07125* (2022).

[41] K. Choi, J. Yi, C. Park and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," in *IEEE Access*, vol. 9, pp. 120043-120065, 2021.

[42] Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. "Convolutional sequence to sequence learning." In *International conference on machine learning*, pp. 1243-1252. PMLR, 2017.

[43] T. Cover and P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, January 1967.

[44] IEEE 14-Bus System: http://www.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm

[45] IEEE 30-Bus System: http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.htm

[46] Shahriar, Steven G.. "iDDAF: An Intelligent Deceptive Data Acquisition Framework for Secure Cyber-Physical Systems." . In Security and Privacy in Communication Networks (pp. 338–359). Springer International Publishing, 2021.

[47] Hebrail, Georges and Alice Berard. 2006. Individual Household Electric Power Consumption. UCI Machine Learning Repository.

[48] M. H. Shahriar, A. A. Khalil, M. A. Rahman, M. H. Manshaei and D. Chen, "iAttackGen: Generative Synthesis of False Data Injection Attacks in Cyber-physical Systems," *2021 IEEE Conference on Communications and Network Security (CNS)*, Tempe, AZ, USA, 2021, pp. 200-208.

[49] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

[50] Kingma, Diederik P. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

[51] Chollet, François, and others. "Keras." (2015). https://keras.io/api/.

[52] https://github.com/HBMDDS/fdia_detection.git

Received September 2024; revised December 2024; revised February 2025