2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Online neural-detection of false data injection attacks on financial time series

1st Alma Y. Alanis
CUCEI, University of Guadalajara
Guadalajara, Mexico
alma.alanis@academicos.udg.mx

2nd Oscar D. Sanchez*
CUCEI, University of Guadalajara
Guadalajara, Mexico
didier.sanchez@academicos.udg.mx

3rd Alejandra Ibarra
CUCEI, University of Guadalajara
Guadalajara, Mexico
alejandra.ibarra3564@alumnos.udg.mx

4th Eduardo Mendez
CUCEI, University of Guadalajara
Guadalajara, Mexico
eduardo.mendez@academicos.udg.mx

5th Jorge D. Sanchez
CUCEI, University of Guadalajara
Guadalajara, Mexico
jorge.sanchez7045@alumnos.udg.mx

6th Jorge Galvez
CUCEI, University of Guadalajara
Guadalajara, Mexico
jorge.galvez@academicos.udg.mx

***Abstract*—False data injection detection is a topic of interest because systems are prone to cyberattacks which can manipulate the state estimation process by injecting malicious data into the measurements, bypassing the detection of the security system. Causing the results of the state estimation to deviate from the safe values. This work proposes a false data injection detection methodology based on deep neural networks using sliding windows to generate online error vectors in order to detect and classify malicious data from measurement data. Two multilayer perceptron deep neural networks and the convolutional neural network were used in this work. In order to verify the feasibility of the proposed methodology, it is tested on data daily closing prices of the S&P 500 Index, pulled from Yahoo Finance for the years 2013–2022 to which false data were injected via software. The results show that the convolutional neural network presents the best results, with an accuracy above 93% and an F1-score of 0.91. It is shown that deep neural networks are a powerful tool in the detection of false data in data obtained through measurements.**

***Index Terms*—deep neural networks, false data injection, detection, real data**

I. Introduction

Financial indexes play a critical role in financial analysis by providing quantitative measures of a company's financial health and performance. These tools allow evaluating the liquidity, profitability, indebtedness and efficiency of a company, providing key information for making informed decisions. There are several financial indices considered the most important due to their relevance to evaluate the financial performance of a company: liquidity, profitability, indebtedness and efficiency index. On the other hand, the injection of false data into financial indices is an illegal practice, which, although it is rare, should not be ruled out in the financial field. Therefore, it is essential to have effective mechanisms to detect the injection of false data into financial indexes, the manipulation of financial data can lead to distortions in the evaluation of the performance and financial health of a company, which in turn can negatively affect investment decisions and the value of the stocks. Then its early detection can prevent harmful consequences for investors and the market in general [1]–[4]

The detection of false data injection (FDI) by cyberattacks can be classified into two categories, on the one hand, model-based detection methods and, on the other, model-free or data-driven detection algorithms. The main disadvantage of using model-based methods is that it is not always possible to have one. In response to the situation, new ways to detect attacks based on the inconsistency of historical data have been proposed [5].

Methods for the detection of FDI mainly use differences in the probability distributions of historical and current. However, in the data obtained by measurements it may not be applicable, for example, assuming that the attack vector is a trapezoidal attack or that the injected spurious data does not deviate significantly from the historical trend [6]–[8]. This makes it easy to falsely detect when in real applications, such as sudden changes.

Due to this, there are other model-free methods such as in [9] where a method for the detection of false data injection was proposed using the difference in the probability distribution between the historical measured data and that of the current measurement data, which presents a good detection performance against real data. Regarding model-based detection methods, we can mention the work presented in [10] he proposed a scheme to detect data prior to state estimation by using a vector autoregression model.

On the other hand, machine learning, unlike detection algorithms based on models for false data injection, are tools that depend on historical data of the system under study. In the work presented in [11], an FDI detection method was proposed. When there is a correlation of spatial and temporal data. By using wavelet transforms and deep neural networks to analyze the estimated states in continuous time. In [12] an extreme machine learning of a class and a network was proposed to detect FDI. The identification layer subnet uses the end machine learning algorithm to classify false data and

This research received external funding from CONACYT FOPl6-2021-01 number 319608.

normal data. On the other hand, [13] proposes a method to detect FDI using moving average, correlation and machine learning algorithms.

This article proposes a method of detection of false data injection by online cyberattacks of economic data based on deep neural networks. The attacks add an unknown injection to a system's measurement data intermittently and within normal data ranges. The presence of false data generates an error with respect to the historical data which serves as input to neural networks for detection. We use multilayer perceptron neural networks (MLP) and convolutional neural network (CNN). The main contributions of this work are shown below:

1) Serial data detection is performed by deep neural networks online.
2) False injected data are within the normal ranges of the measured data.
3) The difference of the historical data in a time window is used as input to the neural networks.

II. DEEP NEURAL NETWORKS FOR FALSE DATA DETECTION

The anomaly detection in time series in the real world is not an easy task, especially when it comes to data obtained in real time. In general, in the detection of anomalies in time series data, the order and causality between the data observed over time must be analyzed together. In the case of FDI, there is a need for diagnostic methods that have the capacity to analyze large amounts of information that detect the injection of false data from sensors accurately and quickly for their implementation in real time.

For now, most of the methods used in FDI are based on models that accurately describe the system, this being its main disadvantage, since it is not always possible against a model, in addition to the fact that the system can be time-varying susceptibility to noise or disturbances not considered [14]. Both disturbances and uncertainties can cause false alarms making the approach ineffective.

A promising methodology is deep learning which has already proven to be effective in various real-world applications such as image classification, time series processing, language and speech modeling among others [15], [16]. For this work, we propose deep neural networks for online detection of anomalies in time series of economic data. The proposed neural network is the convolutional neural network and is compared with the perceptron neural network. The proposed neural networks are described below.

A. Time Series Classification

In this paper, a time series is defined as a vector $X = [x(0), x(1), ..., x(n)]$, $x(i)$ are real data or values obtained by sensors and t represents the size of the vector.

The data set D contains time series $X(i)$ and its respective class label vector $Y(i)$.

$$D = \{(X(0), Y(0)), (X(1), Y(1)), ..., ((X(N), Y(N)))\} \tag{1}$$

where q represents the number of classes that $Y(i)$ contains such that each element $j \in [1, q]$ is 1 if it is a class of $X(i)$ and 0 otherwise.

B. Deep Neural Networks

In this subsection, we present deep neural networks for the detection of false data injected by cyberattacks.

1) Multilayer Networks: The multilayer perceptron neural network (MLP) is an architecture that has been used in various problems [17], [18], being one of the most popular neural networks due to its robustness, efficiency and flexibility [18]. The structure of the MLP is simple, since it mainly consists of three types of layers: input layer, one or more hidden layers and output layer. Layers are made up of nodes, and these are connected to other nodes in contiguous layers by means of weights.

The output of the network y_o can be calculated by means of the sum of the n nodes that are in the hidden layers multiplied by the input neurons x_i and the weights W as shown below.

$$y_o = f(\sum W_{nm} f(\sum W_{nn} x_i)) \tag{2}$$

where f represents the activation function, W_{nn} are weights of the first layer, W_{nm} are weights of the next layer and m represents the number of neurons.

C. Convolutional neural network

Another deep neural network in classification applications we can mention the convolutional neural network (CNN), whose applications are varied and have proven to be very efficient such as image classification [19], object recognition [20], classification of time series [21] and fault diagnosis [22], [23].

The convolutional neural network has been used mainly in the processing of 2D images, or in information that can be represented in 2D. Currently, 1D convolution layer architectures have been chosen for time series processing, which instead of using 2D filters on the input signal. For this work, a 2×1 convolution kernel is implemented to detect false data injection online.

The convolutional neural network is made up of three layers, a filter bank layer, a non-linearity layer and a feature grouping layer [24].

In the convolution or filter bank layer you use several filters that slide through the input data, the convolution of the filter and the receptive field generate as a result an element that is placed in the next layer, then the filter slides to the next area and the operation is repeated [25].

In the nonlinearity layer, various nonlinear activation functions are applied to limit or cut off the output. Commonly the CNN uses the Rectified Linear Unit (ReLU) activation function as shown below:

$$ReLU = \begin{cases} 0, & \text{if } x < 0, \\ x, & \text{if } x \geq 0. \end{cases} \tag{3}$$

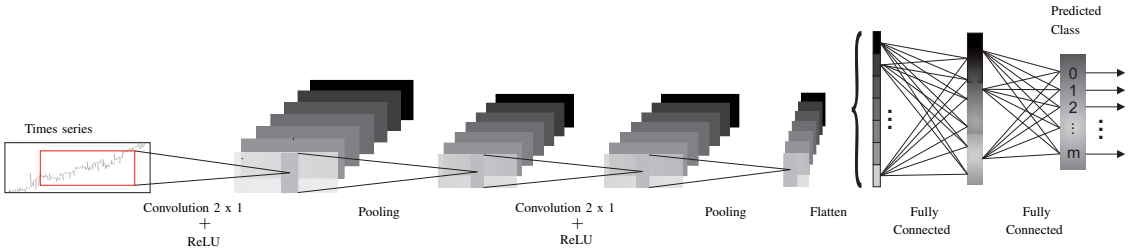

Fig. 1: CNN architecture for FDI, where m is the number of classes.

Finally, the feature extraction layer used to reduce the dimension of the data, the methods used by CNN are maximum pooling and mean pooling. The graphic representation of the three layers is described in the figure 1.

III. Detection and isolation scheme based on deep Neural Networks

As described above, the FDI problem can be separated into two stages, on the one hand, the timely detection of the fault and then, the isolation of the sensor where occurs the fault. The methodology used is described below.

A. Fault detection logic

In this paper we address the problem of false data injection in univariate series (data obtained from measurements of a single variable) which can be introduced at any time.

Considering univariate series has the drawback that the information is little since there is only one sensor signal. Context and information are extremely useful for neural networks, especially the so-called deep ones.

The task of incorporating context in the processing of univariate time series using neural networks can be through overlapping time windows or using recurrent connections to model the flow of time directly [26]. In the case of time windows, also called time delay embedding, they extract information from past measurements. Its use has been mainly in the study of dynamic systems to understand the nature of attractors, prove non-linearity and chaotic behavior [27].

In this paper an online sliding window is explored, with the purpose of detecting the injection of false data as they are observed, so we define X as a univariate time series defined as $\{x(0), x(1), ..., x(t)\}$. In this way, the sliding window $X(t)$ is a lag vector extracted from the time series and formed by the current sample t and past values $d-1$, that is:

$$X(t) = [x(t-(d-1)), ..., x(t-1), x(t)] \tag{4}$$

d represents the size of the array with sample lags $\{1, 2, ..., d-1\}$. From the lag vector an error vector is generated using the current sample $x(t)$ and the past samples $\{x(t-1), x(t-2), ..., x(t-d-1)\}$, as follows:

$$E(t) = [x(t)-x(t-(d-1)), ..., x(t)-x(t-2), x(t)-x(t-1)] \tag{5}$$

Then,

$$E(t) = [e(t-d), ..., e(t-1), e(t)] \tag{6}$$

This vector serves as input to the neural network for false data injection detection. The process is iterative, so the operation is repeated for each sample that is obtained. This process is shown in Fig. 2.

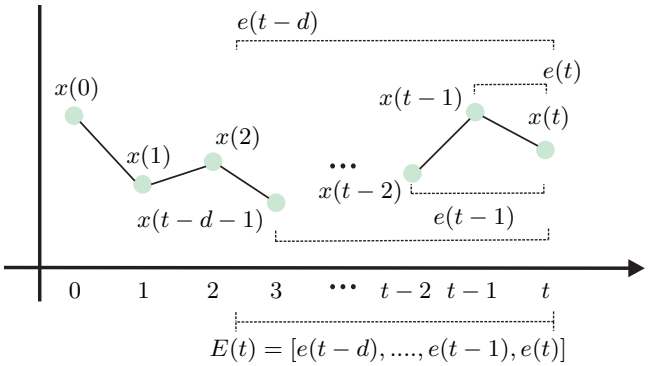

Fig. 2: The sliding window extracted from the time series, which is used to generate an error vector $E(t)$ that serves as input to the neural networks.

The appropriate dimensions of the lagged error matrix can be selected based on the complexity of the time series data and the false data injected. In terms of anomaly detection, different dimensions are selected for the error matrix in order to find the right dimension to improve the classification performance.

B. Architecture of the proposed online neural-detection of false data injection attacks

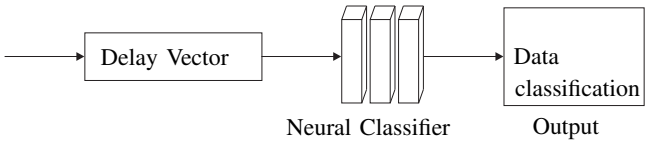

Fig. 3: Configuration of neural networks used to detect the injection of false data.

The data used described above contains false data injection throughout the time series. These injected data are generated from the mean $\pm$ the standard deviation of the normal data, which makes the detection of false data more complicated since these are in the normal damage ranges. Therefore, error

vectors were generated from past samples of dimension d as explained in the III-A subsection, which are used as input to deep neural networks. This approach allows the neural network to better differentiate false data from normal data due to the discrepancy that exists between the current and past samples, with a low computational cost, which allows an online implementation. Fig. 3 represents the proposed configuration for the FDI, using the proposed deep neural networks. The labels for each class are shown in table I.

Label	Fault class
0	Normal data
1	False data

TABLE I: Labels for detection of false data injection.

The architecture of the neural networks used to detect false time series data is described below:

- MLP has 17 neurons in the input layer, two hidden layers with 25 neurons in each layer, and a neuron at the output.
- In general, the architecture of the convolutional neural network is shown in Table II. The parameters are selected in such a way that the network is not too complex and also does not lose classification performance. The size of the convolution layer filters depends on the input vector that is generated by the delay error vector. Therefore, the appropriate size d of the vector $E(t)$ is considered.

Dimension of the delay vector	Convolution+ReLu + Pooling layer	Dense layer 1	Dense layer 2	Outputs
17	20	180	100	1

TABLE II: CNN architecture. The kernel size is 2×1 in the convolution layer.

The delay vectors were varied from dimensions 2 to 20, in order to select the most suitable vector sizes for online implementation. It was observed that very small sizes do not provide enough information for fault classification, however, increasing the amount of information provided to the neural classifier reduces the classification error but increases the computational cost. For this work, the results of vector sizes with the best results according to accuracy and classification time are $d = 17$.

IV. Data description

We analyze the adjusted daily closing prices of the S&P 500 Index, pulled from Yahoo Finance for the years 2013–2022. The financial literature establishes that the historical series of prices are not stationary, which could generate serious problems with the application of chaotic measures and tests. The data is noisy and has not been treated to reduce noise from the time series. The data is shown in Fig. 4.

V. Results

The results were obtained using the proposed neural networks with their respective architectures.

This paper presents economic data that contains injection of false data due to cyber attacks. The figure 5 shows the

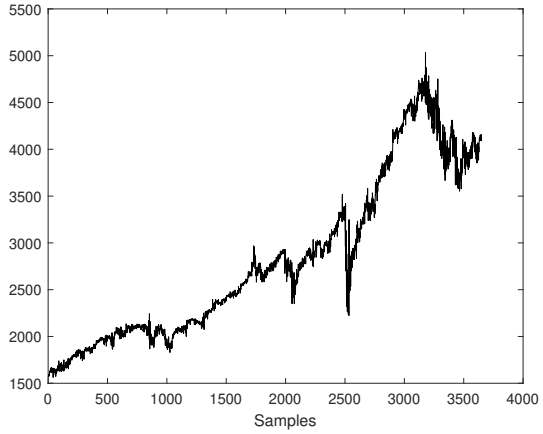

Fig. 4: S&P 500 data set.

economic data, it can also be observed intervals where the injection of false data occurs. For the training data, 80 % of the data and 20 % of the test were considered.

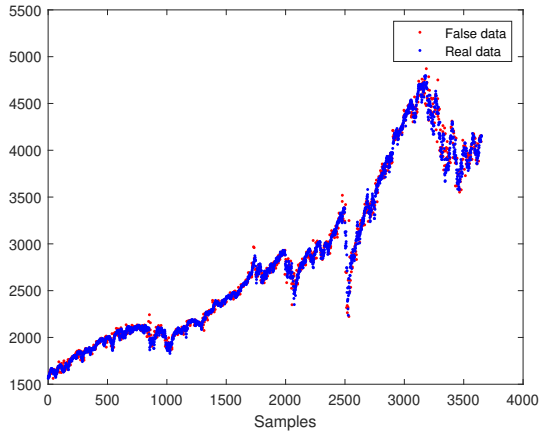

Fig. 5: S&P 500 data set with injection of false data scattered throughout the time series.

It should be noted that the fake data is introduced by means of offline software simulating a fake attack to test the applicability of the proposed classifiers. It can be seen that the injected data is within the normal values of the normal data.

The performance of the neural networks was evaluated by accuracy, confusion matrix and ROC curve (receiver operating characteristic curve). The ROC curve compares the rates of true positive TP and false positive FP that vary between cut-off points. Therefore, the higher the results are above this line, the better performance has the classifier [28].

Classification accuracy (CA) indicates the relationship between the number of correct predictions concerning the total number of samples.

$$CA = \frac{TP + FP}{TP + TN + FP + FN} \tag{7}$$

Precision (P) shows the classification performance of the true positives with respect to the false positives.

$$P = \frac{TP}{TP + FP} \tag{8}$$

The *Recall* (R) returns results on the number of true positives correctly identified.

$$R = \frac{TP}{TP + FN} \tag{9}$$

The $F1$ score is useful when the class distribution is unbalanced. Obtained by combining the measures of *Precision* and *Recall* into a single value.

$$F1 = 2 * \frac{P * R}{P + R} \tag{10}$$

Figs. 6, 8 show the ROC curves of the neural models. Fig. 7, 9 show the confusion matrix of the MLP and CNN neural network respectively. The accuracy, precision, recall and F1-score that each neural network obtained to classify is shown in Table III.

Model	AUC	CA	P(%)	R(%)	F1-Score
MLP	0.8752	0.8008	0.7701	0.7949	0.7823
CNN	0.9797	0.9307	0.9373	0.8998	0.9182

TABLE III: Results obtained with all the proposed neural networks.

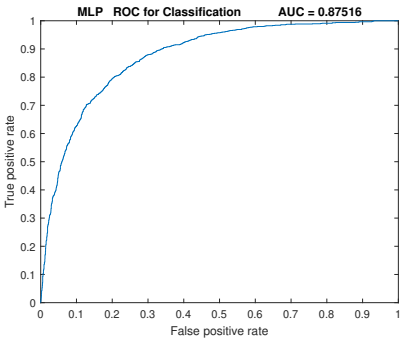

Fig. 6: Classification performance of the MLP neural classifier.

As the dimension d of the retarded error vector was increased, the classification results improved. However, increasing the excess dimension does not significantly reduce the classification error or might even increase it, due to the nonlinearity of the data. Due to this, in this work only the results obtained with a dimension $d = 17$ are presented.

A. Discussion

In this work, the problem of false data injection by cyber attacks that can cause wrong decision making or uncertainty in the veracity of the data was introduced. Typically, the injection of false data is within normal ranges, so it can go unnoticed, in addition to the fact that it can occur at any time, so strategies that monitor the observed data online are

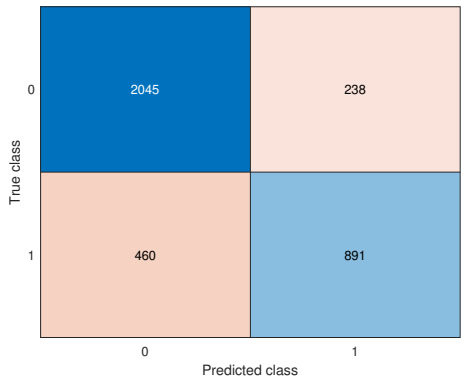

Fig. 7: Confusion plot for MLP neuronal network. True Positive: Neuronal classifier predicts correctly normal samples. True Negative: Neuronal classifier predicts correctly false data. False Positive: Neuronal classifier predicts incorrectly samples as normal samples. False Negative: predictions made incorrectly as false data.

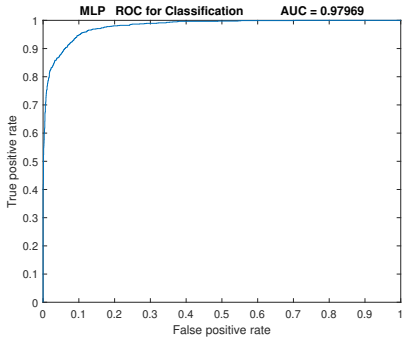

Fig. 8: Classification performance of the CNN neural classifier.

extremely important for timely detection. of the attacks. false inputs are around 30 % of the data and are found throughout the time series. The results obtained from the confusion matrix in Fig. 9 it can be seen that the CNN neural network has a good classification performance for true positive classes as well as false positive classes. According to the roc curve of the CNN neural classifier, they show a curve with true positives close to 1 and false positives around 0, in addition to an AUC of 97.9 %, which indicates a good classification performance. It can be concluded that the CNN Neural Network works well for false data injection detection.

On the other hand, the MLP neural network presents an AUC of 87.52 % and the confusion matrix presents poor classification performance, which indicates that the neural classifier using the MLP has problems detecting the injection of false data, so the best option to carry out this task is the CNN neural network.

VI. Conclusion

Different neural networks such as MLP and CNN were tested. In addition, context was incorporated into the neural

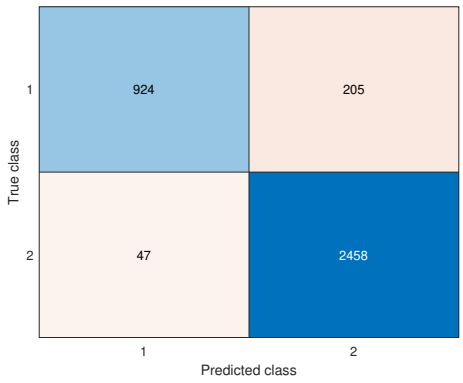

Fig. 9: Confusion plot for CNN neuronal network. True Positive: Neuronal classifier predicts correctly normal samples. True Negative: Neuronal classifier predicts correctly false data. False Positive: Neuronal classifier predicts incorrectly samples as normal samples. False Negative: predictions made incorrectly as false data.

detectors through the generation of sliding windows to generate an error vector.

The classification tests were carried out with real economic data obtained from Yahoo Finance for the years 2013-2022, obtaining a good detection performance by deep neural networks, which motivates us to work with neural networks in the problem of detection online series. of time. In addition, it is intended to continue exploring the detection of false data injection of online time series, for data obtained by sensors.

In future work, we intend to use more complex neural networks for the detection of anomalies, such as the neural network with long and short-term memory which is very useful when dealing with temporal data.

ACKNOWLEDGMENTS

Authors thank the University of Guadalajara for giving us the support to develop this research. The research received external funding from CONACYT FOPl6-2021-01 number 319608.

REFERENCES

[1] L. L. Albu, R. Lupu *et al.*, "Anomaly detection in stock market indices with neural networks," *Journal of Financial Studies*, vol. 9, no. 5, pp. 10–23, 2020.

[2] A. M. Costello, J. Granja, and J. Weber, "Do strict regulators increase the transparency of banks?" *Journal of accounting research*, vol. 57, no. 3, pp. 603–637, 2019.

[3] K. Golmohammadi, O. R. Zaiane, and D. Díaz, "Detecting stock market manipulation using supervised learning algorithms," in *2014 International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2014, pp. 435–441.

[4] Z. Zhao and T. Bai, "Financial fraud detection and prediction in listed companies using smote and machine learning algorithms," *Entropy*, vol. 24, no. 8, p. 1157, 2022.

[5] D. K. Molzahn and J. Wang, "Detection and characterization of intrusions to network parameter data in electric power systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3919–3928, 2018.

[6] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[7] S. K. Singh, K. Khanna, R. Bose, B. K. Panigrahi, and A. Joshi, "Joint-transformation-based detection of false data injection attacks in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 89–97, 2017.

[8] B. Li, T. Ding, C. Huang, J. Zhao, Y. Yang, and Y. Chen, "Detecting false data injection attacks against power system state estimation with fast go-decomposition approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2892–2904, 2018.

[9] G. Cheng, Y. Lin, J. Zhao, and J. Yan, "A highly discriminative detector against false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2318–2330, 2022.

[10] Y. Chen, K. Hayawi, Q. Zhao, J. Mou, L. Yang, J. Tang, Q. Li, and H. Wen, "Vector auto-regression-based false data injection attack detection method in edge computing environment," *Sensors*, vol. 22, no. 18, p. 6789, 2022.

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

[12] D. Xue, X. Jing, and H. Liu, "Detection of false data injection attacks in smart grid utilizing elm-based ocon framework," *IEEE Access*, vol. 7, pp. 31 762–31 773, 2019.

[13] S. Almasabi, T. Alsuwian, M. Awais, M. Irfan, M. Jalalah, B. Aljafari, and F. A. Harraz, "False data injection detection for phasor measurement units," *Sensors*, vol. 22, no. 9, p. 3146, 2022.

[14] J. Zhang, A. K. Swain, and S. K. Nguang, *Robust observer-based fault diagnosis for nonlinear systems using MATLAB®*. Springer, 2016.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[16] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex systems*, vol. 1, no. 1, pp. 145–168, 1987.

[17] X. Zhai, A. A. S. Ali, A. Amira, and F. Bensaali, "Mlp neural network based gas classification system on zynq soc," *IEEE Access*, vol. 4, pp. 8138–8146, 2016.

[18] T. Lim, M. Ratnam, and M. Khalid, "Automatic classification of weld defects using simulated data and an mlp neural network," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 49, no. 3, pp. 154–159, 2007.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[21] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International conference on web-age information management*. Springer, 2014, pp. 298–310.

[22] X. Pang, X. Xue, W. Jiang, and K. Lu, "An investigation into fault diagnosis of planetary gearboxes using a bispectrum convolutional neural network," *IEEE/ASME Transactions on Mechatronics*, 2020.

[23] S. Chen, Y. Meng, H. Tang, Y. Tian, N. He, and C. Shao, "Robust deep learning-based diagnosis of mixed faults in rotating machinery," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2167–2176, 2020.

[24] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE, 2010, pp. 253–256.

[25] S. Albawi, O. Bayat, S. Al-Azawi, and O. N. Ucan, "Social touch gesture recognition using convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[26] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[27] J. A. Perea and J. Harer, "Sliding windows and persistence: An application of topological methods to signal analysis," *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 799–838, 2015.

[28] J. Cerda and L. Cifuentes, "Uso de curvas roc en investigación clínica: Aspectos teórico-prácticos," *Revista chilena de infectología*, vol. 29, no. 2, pp. 138–141, 2012.