20th International Conference on Renewable Energies and Power Quality (ICREPQ'22)
Vigo (Spain), 27th to 29th July 2022
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, Volume No.20, September 2022

A nested decision tree for event detection in smart grids

J. Turanzas[1], M. Alonso[1], H. Amaris[1], J. Gutierrez[1] and S. Pastrana[1]

[1] Electrical Engineering Department
[2] Computer Science and Engineering Department
Universidad Carlos III de Madrid
Avda. Universidad, 30 28911 Leganés. Madrid (Spain) Phone:+ 0034 916 248333, e-mail:
jaime.turanzas@alumnos.uc3m.es, monica.alonso@uc3m.es, hortensia.amaris@uc3m.es,
josue.gutierrez@alumnos.uc3m.es, spastran@inf.uc3m.es

Abstract.

Digitalization process experienced by traditional power networks towards smart grids extend the challenges faced by power grid operators to the field of cybersecurity**.** False data injection attacks, one of the most common cyberattacks in smart grids, could lead the power grid to sabotage itself. In this paper, an event detection algorithm for cyberattack in smart grids is developed based on a decision tree. In order to find the most accurate algorithm, two different decision trees with two different goals have been trained: one classifies the status of the network, corresponding to an event, and the other will classify the location where the event is detected. To train the decision trees, a dataset made by co-simulating a power network and a communication network has been used. The decision trees are going to be compared in different settings by changing the division criteria, the dataset used to train them and the misclassification cost. After looking at their performance independently, the best way to combine them into a single algorithm is presented.

Key words. Cyberattack, Smart Grid, Machine Learning, FDI, Event Detection.

1. Introduction

Power Systems are critical infrastructures in modern and developed countries since the beginning of the 20th century. If a power failure is long enough, it could lead to complete interruption of key services and the economy.

Unlike more traditional power systems, smart grids have greater interactions and communications with intelligent devices and have a more complex context. All the interactions and communications made between the network and the devices involved make smart grids a more open environment and highly dependent on the quality and reliability of the data. This open environment makes the smart grids more vulnerable to intruders and cyberattacks that may be critical to the proper functioning of the network.

Main detection techniques used in smart grids focus on the development of intrusion detection systems (IDS) [1-3]. The massive incorporation of monitoring and control elements in the smart grids generates a huge amount of data and information about the state of the network. Modifying the information recorded by sensors or smart meters, or altering the operating instructions, is part of the cyberattacks on smart grids known as FDI (False Data Injection). The consequences of these attacks are very broad, from electricity theft by altering the demand of low voltage consumers, to the disconnection of some generators due to erroneous control signals.

Traditionally, techniques based on network models have been used to detect cyberattacks on smart grids. However, both traditional methods based on state estimation, as well as methods of detecting erroneous data, fail in the context of Smarts Grids [4], as well as in the real-time detection of alterations of measures or operation set points, as is the case of FDI type cyberattacks. Currently, techniques based on the use of data emerge as an alternative in the detection of attacks thanks to the massive availability of data in smart grids. This category of tools to detect attacks is the most innovative and the one that presents greater complexity due to the management of the data to be carried out, which is why it is common to use statistical methods and Artificial Intelligence techniques, such as Machine Learning (ML) for the detection of the attack.

In this paper, we present a ML algorithm, based on a decision tree, for FDI cyberattacks. The nested algorithm is able to detect twice: the false data and the device attacked in a cyberattack. A co-simulation between OMNET++ and Simulink is used to create a dataset of FDI attacks to the IEEE 14 buses test system. The dataset is used to train the algorithm. The results show a good performance of the developed algorithm to deal with FDI attack detection.

2. Problem Approach

A. State of Art

The high degree of automated communications between the new intelligent devices and the smart grid, creates a new fissure on the security systems. Today it is easier to attack a power system from anywhere, through internet, anonymously, using a weakness on the communications infrastructure and with a low budget. This new cyberattacks

lead to new types of attack like the false data injection (FDI) situations in which the attacker injects information that replaces all or part of the information being monitored, potentially causing the system to sabotage itself.

Event classification methods employ the historical data recorded in smart grids (information recorded by data acquisition systems, states, and operating instructions) to establish the complex nonlinear relationships between the available data representing network behavior. Several works have applied supervised ML techniques for intrusion detection in smart Grids, such as decision trees [5], Support Vector Machines (SVM) [6], Random Forest [7] and Naïve Bayes [8].

Yueyu Deng et al. look for reduce the prediction times [9]. To detect the event, they look at deviations between a load forecasting method and the actual load with real-time measurements. They used Support Vector Regression (SVR) to get the load forecast and then Support Vector Machine (SVM) to classify the scenarios as normal or FDIA. One of the weak points in this method is that an unpredictable event can happen, and the algorithm would predict that the system is being attacked. A forecasting problem can trigger the predictor and the system cannot be under an attack and its behavior is normal within this irregularity.

Neural Networks (NN) has been used in many ways to detect FDI attacks. Recurrent Neural Networks (RNN) [10] is compared with other methods and performance better than other methods as SVM. RNN are becoming popular due to their ability to temporally contextualise each instance. Other NN as Deep Belief Networks (DBN) [11] have been compared with SVM in different models and the results indicates that each model performance different depending on the power system model (SVM performed better in IEEE-9,14 and 30 and DBM in IEEE-118 and 300).

Mario R. Camana Acosta et al. using randomized trees and scoring good accuracy results [12]. However, the algorithm cannot give more information about where the problem is, that is, where the system is being attacked.

Comparing all the papers mentioned, the better results are not always offered by the same machine learning method, it depends on the system, on the variables measured, the dataset and the way in which the method is trained.

B. Our Proposal

In this article we propose a cyberattack detection algorithm. The algorithm developed is able to detect not only the data modified, but also the attacked device in the grid. Decision trees are used for FDI attack detection algorithm.

For the training process, a dataset is developed based on a co-simulation between OMNET++ and Simulink under normal conditions and simulated FDI attacks. Each scenario is labelled with the status of the power grid and the location of the attack. The simulation envisaged different attacks' type and location.

With both labels, the proposed FDI detection algorithm look for the combination of two models that better perform:

1) *Location*: To detect attacked device of an abnormal situation.
2) *Status*: FDI attack on some electrical feature.

This method is the first to allocate the threat on the smart grid and the first trained with data result of co-simulation of attacks. The type of attacks proposed are also a new feature of our dataset. The final algorithm is the combination of these two different models in order to have an only FDI attack detection algorithm.

3. Attack Detection: Location and Status

A decision tree is a classifier method which classifies the instances using recursive partitioning of data in a tree-like model. The algorithm will stop dividing the groups according to the criteria we define. The criterion can be a minimum number of instances in each group, a maximum or minimum depth on the tree or a minimum gain of information on each partition. Starting with the complete dataset, it will be split depending on the value of some features. The features used as a division criterion on each partition is chosen according to the internal homogeneity in the groups resulting from the partition and the heterogeneity between groups. Depending on the value in these features, instances will be directed to one node or another. These nodes are groups labelled to optimize the accuracy of the model. The criterion used to separate the instances (that measures the inner homogeneity and external heterogeneity) and how to measure the accuracy of the model are hyperparameters.

The settings variation tests are:

1) *Training Dataset:* We will compare the results between the model trained with complete dataset and the dataset without line breakers features.
2) *Division Criteria*: The comparison is made between Giny Impurity (eq.1) and Information Gain (eq.2).

$$I_G = \sum_{i=1}^{J}\left(p_i \sum_{k\neq i} p_k\right) = 1 - \sum_{i=1}^{J} p_i^2 \qquad (1)$$

Where p_i is the probability of being randomly classified as the class i.

$$IG = E_{before\ split} - \sum_{j}^{S} p_j E_{after\ split,j} \qquad (2)$$

Where p_i is the same as in equation 1 and E is Information Entropy:

$$E = -\sum_{i}^{J} p_i log_2 p_i \qquad (3)$$

3) Misclassification Cost Function: This function scores the accuracy of the model. Generally, all the classification errors have the same weight and all they are undesirable in the same way.

Sometimes, there are errors that are more important to avoid than others and the misclassification cost function need to adjust their weights.

We have trained different decision trees and compared their results to find the better combination between them, looking for one decision tree predicting before the other, discarding some cases or not, and studying the difference of performance removing some variables or not.

The final algorithm has two different decision trees, one of them classifies the location of the attack and the other the type of attack

4. Training dataset

The algorithm implemented has been trained with a dataset made from co-simulation of IEEE14 test system on Simulink receiving information from a communication network that send packages of false information simulated on OMNET++ (Figure 1).

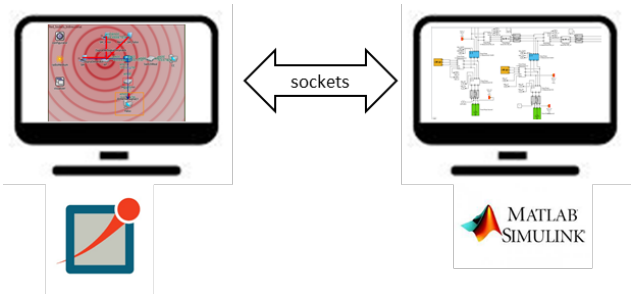

Fig.1. Co-simulation for training dataset creation

The dataset features are shown in Table I.

In the co-simulation false data package has been created randomly in OMNET++ and sending to Simulink IEEE 14 buses test system. These false data substitute original electrical features data and, on some occasions, has been combined with a false status of the line breakers. The last two features (status and bus labels) work as labels for supervised decision tree learning developed. Status labels are as follow:

1) *State 0*: Normal Situation
2) *State 2*: FDI attack on some electrical feature.
3) *State 3*: FDI attack on a line breaker status. A Line breaker can appear as closed without being closed.
4) *State 4*: FDI combined attack on line breakers and electrical feature.

Line labels stand for the line in which the false data appears. Label 0 represents a normal situation, the other labels (from 1 to 14) represent the line in which the attack has been detected.

The dataset has 8.758 instances corresponding to different generation/demand scenarios under normal operation conditions or attacks scenarios. The instances corresponding to attacks scenarios are balanced in number of attack cases of each status and bus breakers. For each instance, electrical and logical features have been recorded. We have used two datasets: the original one with 8758 instances, and a second dataset with reduced number of normal situations to balance the number of instances in the 4 classes. Comparing the results, it seems better to keep the original distribution to better recognize normal situations, which are going to happen most of the time.

Table I. – Dataset Features

Type of Features	Quantity	Total Amount
Electrical Features		
Active and Reactive Power on Generators	10	10
Active and Reactive Power on Buses	28	38
Angle and Module on Buses	28	66
Angle and Module of Loads	40	106
Logical Features		
Line Breakers Status	15	121
Status Label	1	122
Bus Label	1	123

5. Results

A. Training Dataset

In order to determine the minimum dataset size and number of features for the training process, 4 datasets have been used:

Dataset 1.a: The dataset has 8758 instances without line breakers status features.

Dataset 1.b: This dataset is a reduction of complete Dataset 1.a, and it contains only 3600 instances without line breakers status features.

Dataset 2.a: Complete dataset with breaker status features.

Dataset 2.b: Reduced dataset 1.a with 3600 instances and line breakers status features.

All datasets have been used to train the location and status algorithm to compare their performances and how the amount of data affects to the accuracy. As it can be seen in results shown in Fig. 2, without line breaker status feature, the detection algorithm global accuracy is 76.2%. However, it is very difficult to predict an attack of label 3 (FDI on breakers), only 72 instances have been correctly classified, that is the 8.4% of instances related to label 3. By the way, training status algorithm with dataset 2.a, the accuracy has been improved to 99.98%, as it can be seen in Fig. 3. State 3 cyberattack is completed detected with dataset 2.a.

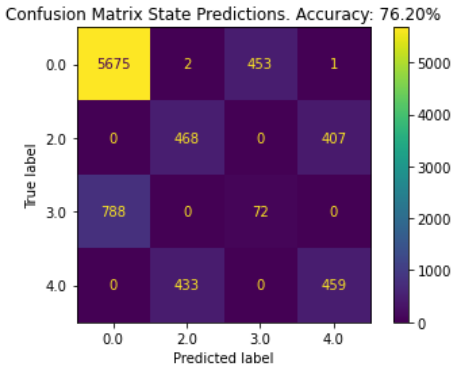

Fig.2. Confusion Matrix of the decision tree that classifies the state of the smart grid. Trained with dataset 1.a.

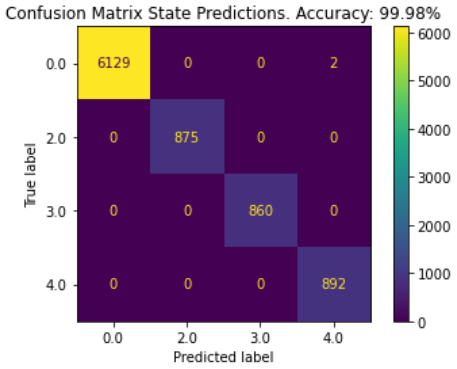

Fig.3. Confusion Matrix of the decision tree that classifies the state of the smart grid. Trained with dataset 2.a

The improvement on the accuracy in dataset 2.a with respect dataset 1.a happens in the same way for the location algorithm. Fig.4 and Fig. 5 show the confusion matrix for dataset 1.b and 2.b training process respectively. As in the previous case, incorporation of line breakers features improves the performance of detection algorithm from 79.39% to 100%.

The accuracy after the training with datasets 1.b and 2.b is shown in Table II. We can appreciate how the accuracy is close to 100% when the training data has the line breakers features, case similar to the Table III. Precision decreases dramatically when training is done without line breakers features in dataset 1.b, scoring less than 50% of accuracy.

As it can be seen, the performance of location algorithm is better than the status algorithm in terms of accuracy, 79.39% vs 76.2%, with a lower size of the training dataset. It can be concluded that status algorithm is more sensitive to breaker status feature than to location algorithm. This may be due to the number of classes.

The accuracy without breakers is not acceptable for a reliable algorithm, and the lack of these features failure to detect FDI attack on them. The dataset for training the algorithm will have all the instances (most of them, normal situations) and all the features shown in Table I.

Table II shows a summary of accuracy for status and location algorithm.

B. Division Criteria

All the decision trees trained previously have used Giny Impurity as the criteria to divide the instances group in each node into two new groups, but it is not the only criterion. We show in Table IV the results of the accuracy for a decision tree trained with a complete dataset but a different criterion. The criterion used was Entropy Information Gain whose main difference it has been the computation time. The results shown in Table IV show a slightly lower accuracy in most of the cases.

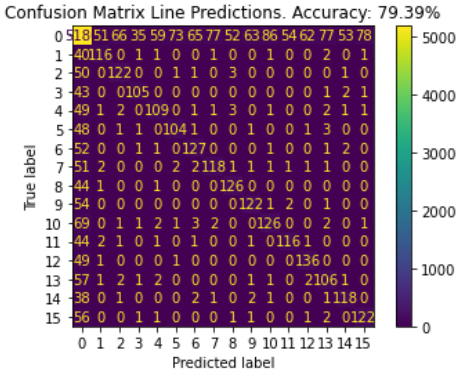

Fig.4. Confusion Matrix of the decision tree that classifies the location of the smart Grid. Trained with dataset 1.a

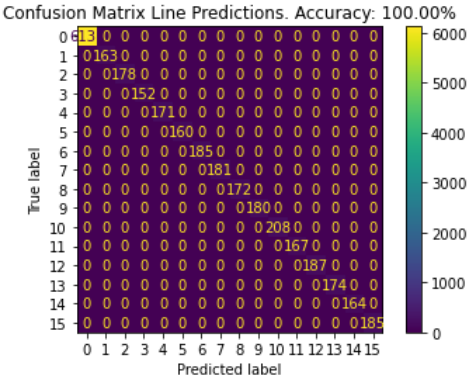

Fig.5. Confusion Matrix of the decision tree that classifies the location of the attack. Trained with dataset 2.a

Table II. – Accuracy with balanced normal situations

	With line breakers	Without line breakers
Status Prediction	99,63%	49,67%
Location Prediction	99,86%	63,39%

Table III. – Accuracy of individual decision trees

	With line breakers	Without line breakers
Status Prediction	99,98%	76,20%
Location Prediction	100%	79,39%

Fig.6 shows the confusion matrix of the decision tree trained with dataset 1.a. It could be seen that errors' distribution is quite similar to the one presented in Fig. 2. Comparing Giny Impurity and Entropy results we can conclude that there is no significant accuracy variation except in the status prediction without breakers, where decrease from a 76,20% accuracy to a 59,83%.

Table IV. – Accuracy of individual decision trees using Entropy Information Gain as division criterion. Model trained with the full dataset.

	With line breakers	Without line breakers
Status Prediction	99,98%	59,83%
Location Prediction	99,95%	78,83%

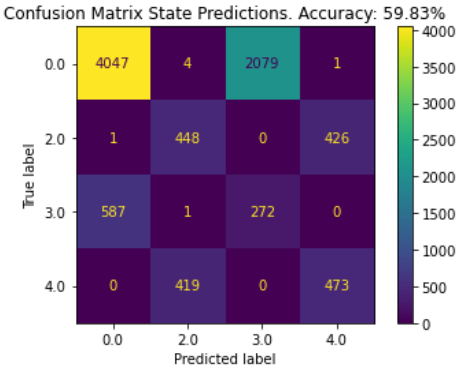

Fig.6. Confusion Matrix of the decision tree that classifies the state of the smart grid. Trained with dataset 1.a

B. *Weights*

The misclassification function is the function that gives the accuracy score of each model. Initially, this function understand that all misclassification errors are undesirable in the same way, but in the case of study it is not true.

In case of a cyberattack, it is better to predict a case of attack and being a false alarm rather than ignore an intruder. That is the reason why we tested training decision trees with different weights for each class of error. The weights have been configured in such a way that an error that causes an attack to be ignored is penalised more heavily than an error that confuses types of attack.

Results presented in Fig. 7 show confusion matrix of a decision tree that predict the state of the power system. It has been trained with dataset 1.a.

The global accuracy is similar in all cases as we can see in Table V with a slight improvement in the results, but the main difference is on the distribution of the errors.

Table V. – Accuracy of individual decision trees using unbalanced weights

	With line breakers	Without line breakers
Status Prediction	99,93%	72,04%
Location Prediction	99,99%	80,34%

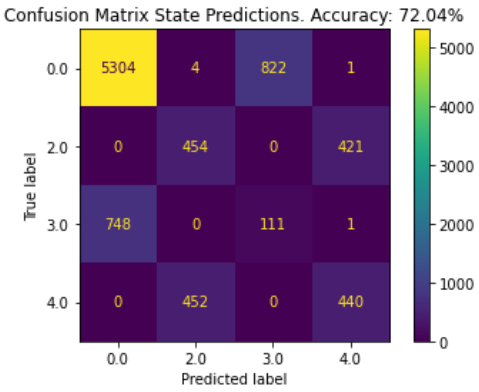

Fig.7. Confusion Matrix of the decision tree that classifies the state of the smart grid. Trained with dataset 1.a and with Entropy Information Gain division criteria

D. *Nested Algorithm*

From the results above, apart from the differences in performance in the configurations we can obtain the following conclusions:

1) *Dataset without Breakers:* The dataset with breakers instances is pretty accurate, there is barely any room for improvement.
2) *Location status ignores fewer attacks*: Looking at the first column of the confusion matrices we can appreciate that there are more instances classified as classes corresponding to attacks in status predictors rather than in location predictors. This may be due to the more classes in location predictors. The final model will use the location status to be the algorithm to classify if there is an attack or not.
3) *The division criteria and weights*: The differences are small enough to discard them in this first nested model, but can be interesting training the location predictor with the unbalanced misclassification cost function

Considering the previous conclusions, the nested final model first classifies the attack's location and then classifies the attack's types as soon as the data is received. In all cases, the location algorithm is more accurate than the state algorithm, that is the reason why the algorithm has this workflow.

Fig. 9 show the results of the nested FDI algorithm. As it can be seen, the status algorithm accuracy of the nested algorithm trained with dataset 1.b is improved compared to individual status algorithm. Moreover, misclassification attacks have been reduced in a 50%.

Table VI. – Accuracy of individual decision trees using the nested Model compared with the results of Table II.

	Simple FDI Algorithm	Nested FDI Algorithm
Status Prediction	76,20%	80,59%
Location Prediction	79,39%	79,39%

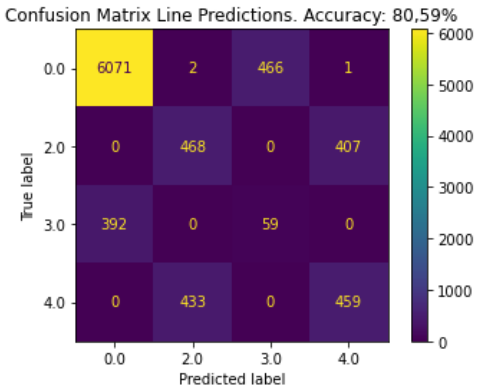

Fig.9. Confusion Matrix of the nested model that classifies the state of the smart grid. Trained with dataset 1.a

5. Conclusion

To deal with smart grids cyber challenges, in this paper an event detection algorithm is developed. The event detection algorithm objectives are twice: detect an event (status) and detect the location of the event (location), that is, the device attacked. Two decision trees have been developed to deal with individual objectives. The final event detection algorithm combined both individual decision trees in a nested algorithm in order to improve the accuracy of the event detector.

Looking at the results of individual decision trees performance, it can be observed that the algorithms using the features corresponding to the breakers status make almost none misclassification even when FDI attack takes place in the smart grid. The inconvenient of these algorithms is that depends on available information.

In order to reduce the number of features or data involved in the event detector, a nested algorithm has been proposed. The results of individual decision trees been shown that accuracy of location algorithm is higher than accuracy of status algorithm. For that reason, the nested event detector algorithm predicts first the attack location and second the status. The results of the nested algorithm demonstrate an improvement in the status algorithm accuracy.

Acknowledgement

This research was funded by Fundación Iberdrola España, within the 2020 research support scholarship program.

References

[1] S. Pastrana, J.E. Tapiador, A. Orfila and P. Peris-Lopez, '"DEFIDNET: A framework for optimal allocation of cyberdefenses in Intrusion Detection Networks," *Computer networks (Amsterdam, Netherlands : 1999)*, vol. 80, Apr 7, pp. 66-88.

[2] J. Hong and C. Liu, '"Intelligent Electronic Devices With Collaborative Intrusion Detection Systems," *TSG*, vol. 10, no. 1, Jan, pp. 271-281.

[3] M.S. Rahman, M.A. Mahmud, A.M.T. Oo and H.R. Pota, '"Multi-Agent Approach for Enhancing Security of Protection Schemes in Cyber-Physical Energy Systems," *TII*, vol. 13, no. 2, Apr, pp. 436-447.

[4] M.M.N. Aboelwafa, K.G. Seddik, M.H. Eldefrawy, Y. Gadallah and M. Gidlund, '"A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT," *JIoT*, vol. 7, no. 9, Sep, pp. 8462-8471.

[5] 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," *TII*, vol. 12, no. 3, Jun, pp. 1005-1016.

[6] Yi Wang, M.M. Amin, Jian Fu and H.B. Moussa, '"A Novel Data Analytical Approach for False Data Injection Cyber-Physical Attack Mitigation in Smart Grids," *Access*, vol. 5, pp. 26022-26033.

[7] S. Ahmed, Y. Lee, S. Hyun and I. Koo, '"Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest," *TIFS*, vol. 14, no. 10, Oct, pp. 2765-2777.

[8] da Silva, Paula Renatha Nunes, H.A. Gabbar, P. Vieira Junior and da Costa Junior, Carlos Tavares, '"A new methodology for multiple incipient fault diagnosis in transmission lines using QTA and Naïve Bayes classifier," *International journal of electrical power & energy systems*, vol. 103, Dec, pp. 326-346.

[9] W. HU, Z. LU, S. WU, W. ZHANG, Y. DONG, R. YU and B. LIU, '"Real-time transient stability assessment in power system based on improved SVM," *J MOD POWER SYST CLE*, vol. 7, no. 1, pp. 26-37.

[10] Y. Wang, D. Chen, C. Zhang, X. Chen, B. Huang and X. Cheng, '"Wide and Recurrent Neural Networks for Detection of False Data Injection in Smart Grids,", vol. 11604, pp. 335-345.

[11] L. Wei, D. Gao and C. Luo, '"False Data Injection Attacks Detection with Deep Belief Networks in Smart Grid," *CAC*, pp. 2621-2625.

[12] M. R. Camana Acosta, S. Ahmed, C. E. Garcia and I. Koo, '"Extremely Randomized Trees-Based Scheme for Stealthy Cyber-Attack Detection in Smart Grid Networks," *IEEE Access*, vol. 8, pp. 19921-19933.