Intrusion Detection in Smart Grid Distribution Domain Using Deep Ruptures Detection

Sarah Chahine
Faculty of Engineering
University of Balamand
Balamand, El Koura, Lebanon
e-mail: sarah.chahine@std.balamand.edu.lb

Chafic Mokbel
Faculty of Engineering
University of Balamand
Balamand, El Koura, Lebanon
e-mail: chafic.mokbel@balamand.edu.lb

Abstract— **Smart grids brought huge added value to classical power grids in terms of advanced monitoring, metering control, trustworthiness and efficiency. This comes with some challenges, a major one being assuring the security of the grid against cyber-attacks. Obviously, such concerns are serious because of the impact and risks on electrical energy provisioning. To prevent and react to possible attacks, intrusion detection appears as a critical component. Previous literature work shows that an intrusion onto the grid translates into a small glitch that a phasor may help in identifying. In this work, we suggest to detect the glitches directly from electrical signals (current, voltage, frequency and power). We suggest using the detection of changes in the signals properties as an indicator of intrusion. To this end, classical approaches in ruptures detection have been experimented. A new approach based on deep Long Short-Term Memory (LSTM) filtering is proposed. The main focus of our work is on intrusions occurring in the distribution domain. In order to conduct experiments for the validation of the techniques, simulated data have been produced. The built simulator is also described in the paper. Benchmark results permit to confirm that our newly proposed deep nonlinear LSTM-based method is a viable solution to consider for intrusion detection for the distribution domain in a smart grid.**

Keywords-Cyber security; Smart Grid; Intrusion Detection; Ruptures Detection; Deep Filters; LSTM.

I. INTRODUCTION

The smart grid is the result of connecting the grid components using a communication network and extending the functionality through advanced monitoring, metering control and actuation devices. This permits to achieve better productivity, trustworthiness and efficiency [1]. Such achievement does not come without a major cost. In this particular case, concerns are raised about the security of the smart grid and the serious consequences of any cyber-attack [2]. With cyber facilities added, the grid does not only face power failures such as single line-to-ground, line-to-line, and two-phase-to-earth, but it also faces attacks on the security of the communication. If the network is compromised, an intruder can cause faults and more critical threats that endanger the power generation, transmission and distribution. A hacker getting control of the system may induce a big shut down, a change in the loads, a change in the pricing of the kilowatt and many more. The induced faults might be cascaded throughout the grid where a failure in one component can affect many others around it [3].

Cyber-attack on the smart grid can occur at different levels and in various domains. In this work, we are particularly interested in the attacks within the distribution domain. Jamei et al. [4] propose to build resilient Cyber-Physical Systems (CPS) by using tools that monitor and analyze data collected from Micro-Phasor Measurement Units (μPMU). The authors show that an intrusion onto the grid translates into a small glitch that a phasor may help in identifying.

In the present paper, we assume that an intrusion leads to glitches in the electrical signals. Thus, the detection of glitches alerts about possible intrusion. We suggest to collect normal electrical signals (current, voltage, frequency and power) from the distribution domain of a smart grid and to apply ruptures detection algorithms in order to localize glitches and, thereby, possible intrusions.

Detecting changes in stochastic signals has been a major domain of research in the past few decades [5]-[8]. Applications exist in several sectors. In order to detect an abrupt change in signal characteristics, two components are needed: i) a model providing a cost function and ii) a search strategy. The model generally compares the properties of local parts of the signals to background properties of the same signals. Parametric and statistical models or statistical hypothesis testing may be used. Once the model is identified and trained, a search strategy is needed in order to select and slide the local and background windows of the observed signals across time. In this paper, we use the ruptures open source to detect the glitches [8]. We suggest using two deep neuronal nonlinear predictive filters in order to estimate the costs of an abrupt change, i.e., the presence of glitches. An interface between our nonlinear filters and the ruptures is also being developed in order to apply the same search strategies.

The paper is organized as follows. Section II provides a short description of smart grids and cyber security. Section III describes the approaches used to perform the detection of intruders. Section IV presents the simulator that has been specifically developed to obtain data for the experiments. The experiments conducted in order to validate the approach and benchmark the different technologies are detailed in Section V. Finally, the paper ends with the major conclusions and several perspectives.

 ISBN: 978-1-61208-788-7

II. SMART GRIDS AND CYBER SECURITY

Kim et al. [9] define the smart grid as an electrical grid that couples the power system with an Information Technology (IT) system. This offers several advantages in terms of advanced monitoring, control and efficiency, but introduces new risks, a major one being the threats related to security.

A. Smart Grids

National Institute of Standards and Technology (NIST) identifies seven domains in a smart grid [10]: Generation, Transmission, Distribution, Customer, Operations, Markets, and Service Provider. The electrical power is generated at different generation stations in the generation domain. The transmission domain is where the energy produced is being transmitted to the consumers. The distribution domain is where the test feeder is situated, and it is the domain the customers are connected to. It is the place where the high voltages are lowered and regulated for common use. The customer domain is where the energy is finally consumed. Customers are the end users. These four domains: generation, transmission, distribution and customer, form the physical system of a grid. Electrical power lines connect the different components of these four domains. In this work, we are particularly interested in the distribution domain.

Besides the physical system, three other domains exist in a smart grid. Service providers are just like any type of service providers in a different sector. To illustrate this, one can compare them to the Internet service providers where they are the direct contact with the customers. The operations domain is where the controller of the grid resides. The controller receives measurement and other monitoring information about the grid through the communication network. Based on the received information, the controller takes informed decision and sends commands to the different controlled units in the grid. Finally, the markets domain is where the marketing issues related to power production and consumption are treated.

Secure network communication links interconnect all the components of the seven domains. This network allows devices, systems or programs to exchange necessary information and interact for executing advanced applications within the smart grid.

B. Cyber Security

The introduction of the cyber system to the smart grid did not just bring technological advancements, it also introduced new and critical problems. Now, the grid not only faces power failures, such a single line-to-ground, line-to-line, and two-phase-to-earth, but it also faces attacks on the security of the communication. This can let a hacker in, and that can introduce faults and more critical threats that endanger the power generation, transmission and distribution. A hacker getting control of the system may induce a big shut down, a change in the loads, a change in the pricing of the kilowatt and many more. The induced faults might be cascaded throughout the grid where a failure in one component can affect many others around it [3]. Generally, the concerns are about two major classes: power grid safety and data safety. Actually, a cyber-attack can occur on different nodes in any part of the grid and in any domain. It can be a distributed denial of service, false injection of data, gaining access and control over the system, tapping the system and eavesdropping on the data passing, spreading a malware and many more. In this work, the objective is to build an intrusion detection system to cope with attacks that might occur in the distribution domain in order to assure better power grid safety.

C. Intrusion Detection

The methodologies of intrusion detection are categorized as “Anomaly-Based Detection” (AD), “Signature-Based Detection” (SD), and “Stateful Protocol Analysis” (SPA) [12]. A “Signature-Based Detection” is a detection where the attack is known and, thus, the method to solve it is also known. It can be described as Knowledge-Based detection. The “Anomaly-Based Detection” identifies deviations from the expected behavior. The normal behavior is picked up from studying the background data for a while. It can be described as Behavior-based Detection. The “Stateful Protocol Analysis” may look like the “Anomaly-based Detection”, however, it is based on knowing and tracing the protocol. SPA is also known as Specification-based detection.

Several studies have been conducted to build an Intrusion Detection System (IDS) for smart grids. Sedjelmaci and Senouci studied a combination of both distributed and centralized IDS with a focus on attacks such as Denial of Service (DoS), faulty data injection and resource injection [13]. They focused on the use of machine learning along with rule-based detection. They discovered that using rule-based alone consumed more energy, while the combination of machine learning and rule-based detection led to lowering the use of energy needed in detection. In addition, the detection turned out to be improved when combining both. Yu et al. presented an anomaly-based and watermarking-based IDS to counter false data injection [14]. Using watermarking, they insert hidden data to be able to verify the authenticity of the exchanged information and to detect any malicious injection.

Jamei et al. used microphasors to detect intruders [4]. Phasors are usually used in the transmission domain and much less in the distribution domain. The authors introduced microphasors in the distribution domain for cyber-attack detection. They compared the results using those microphasors to those from a distributed SCADA system.

Ozay et al. [15] use machine learning algorithms to detect complications in the smart grid system. These algorithms are also used to detect attacks and be able to differentiate faults from attacks. The fewer the False Positives and False Negatives, the higher the accuracy of the machine learning algorithm. The lack of data to train these machines presents a serious challenge. The use of machine learning algorithms is found extensively in a lot of researches to monitor and control systems [15]. It is also used now to detect attacks and be able to differentiate faults from attacks.

 ISBN: 978-1-61208-788-7

In the present work, we aim at building an anomaly-based IDS in the distribution domain using deep predicting filters that detect changes in measured electrical signals.

III. INTRUSION DETECTION BY DETECTING RUPTURES

As mentioned above, the proposed approach to intrusion detection relies on ruptures detection in electrical signals. Classical ruptures detection approaches are studied and experimented. We suggest performing rupture detection using a deep nonlinear filter.

The key idea in our intrusion detection is to catch a change in the electrical signals' properties, i.e., one or more of the measurements of voltage, current, frequency and power as a function of time.

Ruptures can be detected when changes in signal properties are observed. In order to achieve such detection, several approaches can be used. One approach consists in building a model representing the signals' background characteristics and to decide if such a model fits with each short window of the signals [7]. Another approach compares statistical properties or performs statistical tests in order to verify if the signals in different windows are the results of a unique process [8].

In this paper, we adopt the same formulation as in [8]. Let $\underline{y} = \{\underline{y}_1, \ldots, \underline{y}_T\}$ be a Multivariate non-stationary random process where $y_t \in \mathbb{R}^d$. It is supposed that $\underline{y}$ is piecewise stationary, i.e., there exist K unknown instants of ruptures $t_1^*, \ldots, t_K^*$ where some characteristics of $\underline{y}$ change. In order to determine K as well as the instants of ruptures, a criteria function is defined as:

$$V = \sum_{k=1}^{K-1} c\left(\underline{y}_{t_k}, \ldots, \underline{y}_{t_{k+1}}\right) \tag{1}$$

In (1), $c(.)$ is a cost function which measures goodness-of-fit of the signal segment to a specific model, as defined in [8]. If K is unknown, the cost is compared to a threshold in order to decide on the ruptures. The model represents the background information while local signals are to be tested against the background model in order to compute the cost. Once the model-cost function is defined, a search algorithm must be adopted. This algorithm defines how the local and background windows are set and used to explore the signals.

The models-cost functions are categorized into parametric and non-parametric [8]. Some parametric cost functions are: Maximum Likelihood Estimation, Multiple Linear Model, and Mahalanobis-type Metric. Some non-parametric cost functions are: Non-Parametric Maximum Likelihood Estimation, Rank-Based Detection, and Kernel-Based Detection.

In [8], search methods are classified into three different categories:

- Window-Sliding
- Binary Segmentation
- Bottom-Up Segmentation

Binary segmentation is "greedy sequential algorithm" [8]. The Bottom-up segmentation of a signal is used to perform fast segmentation of a signal. It works just the opposite of Binary segmentation. It starts first with multiple points of change and then decreases them by taking out the less important ones (the ones with least inconsistency) until there remains only the one that actually represents the correct number of changes. This is done by splitting the signal into various small sub-signals and then merging these parts sequentially until there remains the number of change points only. The window-sliding method computes the difference between two adjacent windows. The discrepancy can be described by the function below:

$$d(y_{a..t}, y_{t..b}) = c(y_{a..b}) - c(y_{a..t}) - c(y_{t..b}) \quad \text{[8] (2)}$$

where $1 \le a < t < b \le T$

This function identifies how this approximation method measures the difference between one window and the one just after it. The distance is higher if the cost of the concatenation of the two adjacent windows is higher than the sum of the costs relative to each window taken separately.

For identifying the number of change points, certain constraints are taken based on whether the points of change are known or not. The accuracy of any detection method is the ability to estimate correctly the place of change points.

A. Classical Models

In the present work, the following models have been experimented for cost calculation:

- Autoregressive (AR)
- Least Absolute Deviation
- Least Squared Deviation
- Linear Model Change

An autoregressive model of order p computes an estimate of the present sample of a signal as a linear combination of the p previous samples. In case of scalar signals, this can be written as:

$$\tilde{y}_t = \sum_{i=1}^{p} a_i y_{t-i} \tag{3}$$

The autoregressive combination coefficients are determined in a way to minimize the mean square prediction error.

$$\underline{\hat{a}} = \underset{\underline{a}}{\operatorname{argmin}}\, Err\left(y_i \ldots y_f\right) = \underset{\underline{a}}{\operatorname{argmin}} \sum_{t=i+p}^{f} \|y_t - \tilde{y}_t\|^2 \tag{4}$$

The linear model change minimizes the mean square prediction error. Let $0 < t_1 < t_2 < \ldots < n$ be the unknown points of change. The linear regression model is described as:

$$y_t = \underline{z}_t^T \underline{\delta}_j + \varepsilon_t, \ \forall\ t = t_j, \ldots, t_{j-1} - 1 \text{ for } j > 1 \tag{5}$$

where y_t is the observed dependent variable, $\underline{z}_t$ is the covariate vector and $\underline{\delta}_j$ is the prediction coefficients vector.

The cost function over an interval I is the minimum mean square of the prediction error ε_t.

$$c\left(\underline{y}\right) = \sum_{t \in I} \|\varepsilon_t\|^2 = \sum_{t \in I} \left\|y_t - \underline{z}_t^T \underline{\delta}_j\right\|^2 \tag{6}$$

B. Deep Prediction Model

In addition to the previous models, two deep learning nonlinear models were also introduced with a customized cost function [8]. These are the Long Short-Term Memory

(LSTM) and the Multivariate machine learning. These machines have been used as nonlinear predictors of the current values of the electrical signals. It is assumed that, if the machines are well trained and if the signals do not change their properties, i.e., no glitch is present, then the prediction error shall be relatively small. In contrast, a large prediction error shall be observed when a glitch is present.

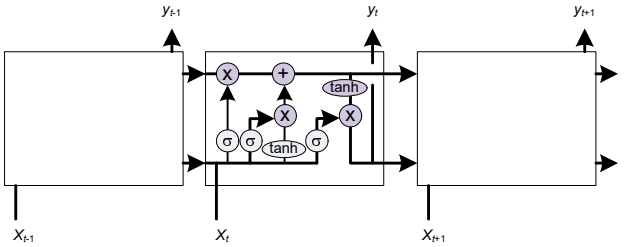

Figure 1. LSTM predictive filter (Reproduced from [16]).

The LSTM is a recurrent neural network. Its architecture is shown in Figure 1. Its name shows that it keeps information for a short term from past inputs. Because of its continuous learning process capability, it increases its resistance to noise and overcomes technical problems that might arise. The two practical complications overcome by LSTM are exploding and vanishing gradients, both linked to how the network is trained.

Multivariate predictive filter is a recurrent neural network as well [17]. Just like the LSTM, the Multivariate is resistant to noise and has a learning capability to be able to detect changes such as abrupt changes in the system. Figure 2 shows the relation of the different layers to the output layer. The purpose of this bidirectional neural network is to train the system both forward and backward while having both the forward and backward layers connected to the output.

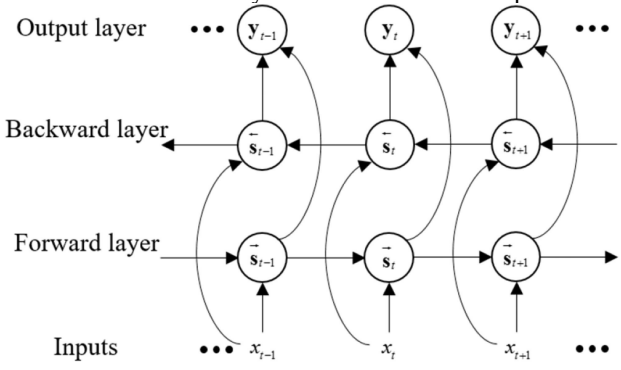

Figure 2. Multivariate prediction filter (Reproduced from [17]).

The LSTM and Multivariate are best used since they can take Multivariate signals, i.e., multiple electrical measurements. Convolutional layers may be added at their input in order to make the filter deeper. The predicted signal at their output serves to compute a cost function that is the mean square of the prediction error.

$$c(\underline{y}) = \sum_{t \in I} \|\underline{\varepsilon}_t\|^2 = \sum_{t \in I} \left\|\underline{y}_t - \underline{\tilde{y}}_t\right\|^2 \quad (7)$$

where $\underline{\tilde{y}}_t$ is the predicted signal by the neural filter.

The mean square prediction error cost function is being used as for non-neural algorithms in order to detect the presence of ruptures in the electrical signals, which identifies possible intrusion.

IV. SIMULATOR

A simulator has been developed in order to generate the data that served in our experimentations. We focus on intrusions occurring in the distribution domain. However, a glitch resulting from an intrusion will be carried back to the main center where appropriate action shall be taken. Thus, the simulation must cover the distribution grid, the transmission grid and the control center. This helps narrow down the section where the problem is occurring. For this reason, the simulator system used here considers the three domains: transmission, distribution, and network.

The simulation system is made up from open-source components: Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS), Network Simulator – 3 (NS-3), GridDyn, and Gridlab-d, in addition to MATLAB. The HELICS acts as the main connector between all the different parts [18]. The NS-3 is used as the network of the system that connects the various parts [19]. The GridDyn [20] is used for creating and simulating the transmission grid. The Gridlab-d [21] and the MATLAB [22] are used for the creation and the simulation of the distribution grid.

HELICS is used to combine GridDyn (Transmission Grid) with NS-3 (Communication Network) along with MATLAB (Distribution Grid with static changes) or Gridlab-d (Distribution Grid with dynamic changes). To set up the system, a server and a client need to be created. The server collects and stores all the data. The client side is where all the action will be taken. The attack is taken up on the distribution grid and, for that reason, the system is created in a way to focus and grab data from that point.

After the setup of the system is completed, several tests are run where a simulation is created as if the hacker is going into the system. The attack creates a glitch for a fraction of time. Attacks in various time frames, phases, and grid states are simulated. A breaker switches on and off for a fraction of time to recreate the same glitch. Data is collected for the current, voltage, power and frequency. An XML file is generated for each test and keeps track of each experiment's context and conditions. For Gridlab-d, the IEEE 13-bus test feeder is used.

Jamei et al. [4] show how the glitch occurs and it was replicated with the setup provided.

 ISBN: 978-1-61208-788-7

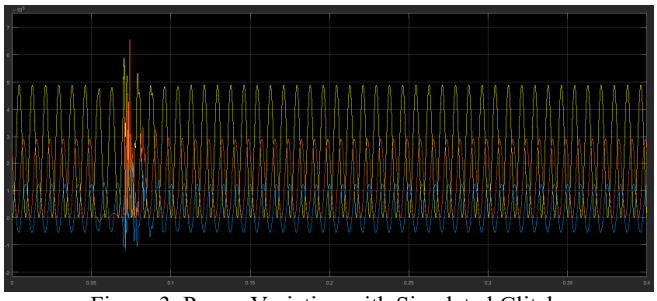

Figure 3. Power Variation with Simulated Glitch.

Figure 3 shows a sample of the simulation where the power signal is generated with a glitch in it. The x-axis represents the time axis while the y-axis shows the power value for phases A, B and C of the power grid.

V. EXPERIMENTS AND RESULTS

Using the simulator, forty different cases have been generated. For each simulation, the instance of the glitch is known. The electrical signals collected are voltage, current, frequency and power. A threshold value is specified on the cost in order to decide on a possible rupture. Ruptures detection algorithms may yield several consecutive detection points. A range of 500 points around the detected point is taken as the same point in order to smooth the detection results. Two types of errors can occur: a detection of rupture when it does not exist (False Acceptance) and a non detection of a rupture (False Rejection). False Acceptance (FA) and False Rejection (FR) rates are calculated based on the number of points detected and their location within the accepted range and based on the known reference points.

A. Classical Ruptures Detection

The forty simulations are fitted into five different models which are AR Order 4, Gaussian Process Change, Least Absolute Deviation, Least Squared Deviation and Linear Model Change. We have experimented several thresholds to which the cost functions are compared. In this paper, we report identical results on two different threshold values 5 and 0. The Gaussian Process Change was not taken into consideration when checking the FA and FR since most of the results did not give any break points, although they existed.

TABLE I. FA AND FR RATES FOR CLASSICAL RUPTURES DETECTION

		FA (%)	FR (%)
AR Order 4	Current	25557.30	0
	Power	20377.67	0
	Voltage	19819.06	0
Least Absolute Deviation	Current	8063.17	290.36
	Power	8861.98	0
	Voltage	11433.17	0
Least Squared Deviation	Current	6851.54	0
	Power	7695.77	0
	Voltage	8374.67	0
Linear Model Change	Current	9195.32	37.50
	Power	16865.10	0
	Voltage	21716.36	22.50

Table I shows the results of FA and FR for the different cost functions. For all the different cost functions, the number of falsely accepted points is high, which makes it far from acceptable for the real application. For the LSTM and Multivariate, the check gets updated every 200-point range. The error is calculated based on cost function the square root of the square sum. Below is the pseudo-code of the error() function.

```
def error(self, start, end):
    if end - start < self.min_size:
      raise NotEnoughPoints
    max = 0.0
    b = start
    s = 200
    while b <= end-s:
      sub=self.signal[b:b+s]
      y=np.sqrt(sum(np.square(sub)))/s
      if y[0] > max:
        max = y[0]
      b = b+s
    self.win_num = self.win_num+1
    return max

def sum_of_costs(self, bkps):
    epsilon=0.01
    for start, end in pairwise([0] + bkps):
      value=self.error(start, end)
      if value < epsilon:
        return value
    soc = max(self.error(start, end)
          for start, end in pairwise([0] + bkps))
    return soc
```

In the error function above, we define a start point and a window size. The error is calculated based on the square root of the square sum of this window. It is then compared with the threshold. If it surpasses the threshold, then this point will be considered as a glitch point.

TABLE II. LSTM PREDICTION ERROR SIGNALS USED AS INPUT OF RUPTURES DETECTION ALGORITHMS.

		FA (%)	FR (%)
AR Order 4	Current	263.11	3.40
	Power	189.23	5.46
	Voltage	190.82	6.98
Least Absolute Deviation	Current	234.61	1.03
	Power	297.20	3.05
	Voltage	256.15	0.31
Least	Current	391.13	0.67

 ISBN: 978-1-61208-788-7

Squared Deviation	Power	285.20	4.83
	Voltage	455.57	0.67
Custom Cost	Current	430.15	3.27
	Power	268.57	3.76
	Voltage	414.78	3.27
Normalized Custom Cost	Current	388.79	1.74
	Power	244.02	4.98
	Voltage	438.90	1.71

TABLE III. MULTIVARIATE PREDICTION ERROR SIGNALS USED AS INPUT OF RUPTURES DETECTION ALGORITHMS.

		FA(%)	FR(%)
AR Order 4	Current	234.17	4.74
	Power	**165.72**	6.23
	Voltage	267.07	5.50
Least Absolute Deviation	Current	217.39	4.07
	Power	263.01	4.12
	Voltage	328.85	4.12
Least Squared Deviation	Current	344.23	3.72
	Power	278.18	4.48
	Voltage	344.11	5.15
Custom Cost	Current	397.03	5.70
	Power	231.03	4.74
	Voltage	420.12	2.24
Normalized Custom Cost	Current	281.18	**2.38**
	Power	247.42	4.48
	Voltage	374.49	4.07

B. *Deep Nonlinear Ruptures Detection*

The LSTM and Multivariate predicted signals are being compared to the real signals and error signals are generated. These signals are fitted into different models which are AR Order 4, Gaussian Process Change, Least Absolute Deviation, Least Squared Deviation, Custom Cost (Square root of the square sum), and Normalized Custom Cost (Normalize the square root of the square sum). The rupture detection threshold is set to 0. The Gaussian Process Change in most of the cases does not deviate into giving a result and detecting any breakpoints. This makes it get ruled out when calculating the FA and FR afterwards.

The same experiments were conducted for Multivariate prediction. The Linear Model Change did not deviate at all and did not give results. The Gaussian Process Change reacted the same way as it did when it was applied to LSTM. For that reason, both the Linear Model Change and the Gaussian Process Change got removed when calculating the FA and FR later.

Tables 2 and 3 present the results in terms of FA and FR rates when the LSTM and Multivariate methods are first applied and then the generated prediction error signal is entered into the various cost functions, i.e., detecting changes of properties in the prediction error signal. Significant improvements are observed. However, the high FA makes it unacceptable as a precision for the electrical sector.

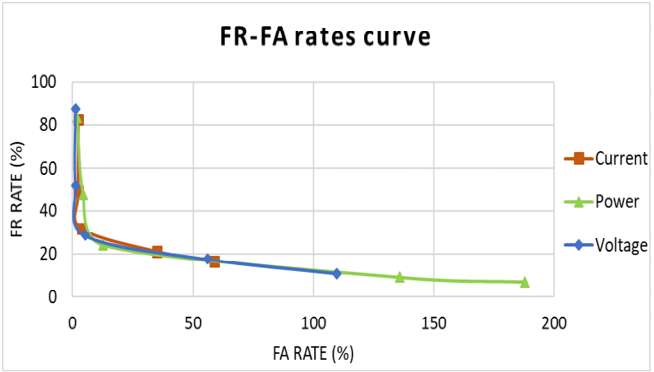

Figure 4. FR function of FA rates for LSTM with normalized cost.

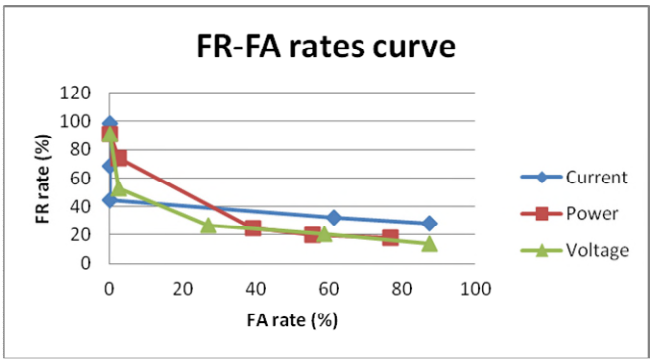

Figure 5. FR function of FA rates for Multivariate with normalized cost.

The prediction error signals obtained at the output of the deep nonlinear filters are now used directly to compute the cost. In this case, the cost becomes the mean square over the signal window (7). FA and FR rates are computed for different thresholds {0.01, 0.05, 0.07, 0.1, 0.2}.

Figure 4 presents the results of the LSTM method. The number of detected points is optimal when threshold is 0.07. A lower number of points is detected as the threshold increases over 0.07. A higher number of breakpoints are detected as the threshold decreases. Both FA and FR are low. The FR is the lowest which means that the system will rarely miss out any of the attackers getting into the system. This combination makes it the optimal thus far.

The same procedure is applied for the Multivariate method. The thresholds used are 0.01, 0.05, 0.07, 0.1, and 0.2. Figure 5 presents the data of the FA and FR rates obtained. The number of detected points is optimal when epsilon is 0.07. A lower number of points is detected as the threshold increases over 0.07. A higher number of breakpoints is detected as the threshold decreases. However, the performance is lower than for LSTM when the threshold=0.07.

By comparing all the above results, it can be concluded that the optimal method applied is when applying the LSTM, considering the cost as the mean square prediction error, and taking the threshold epsilon as 0.07. It gives the optimal

 ISBN: 978-1-61208-788-7

result with a low number of FA. It also yields the minimal FR. This shows that the newly introduced modified system gives the most accurate results.

VI. CONCLUSIONS AND PERSPECTIVES

In this paper, we proposed and studied an Anomaly-Based IDS in the distribution domain that uses deep predicting filters applied on electrical signals to detect ruptures. We assumed that an intrusion in the distribution domain yields a glitch in the electrical signals that might be identified when applying rupture detection techniques.

In order to perform the study, a simulator has been developed and a set of forty signals have been generated representing a variety of grids profiles and intrusions. The simulator covers the three domains: transmission, distribution and networking.

Classical ruptures detection algorithms have been first applied on the signals and provided poor performances. A new deep nonlinear rupture detection technique has been thus proposed. Two deep prediction filters have been developed, an LSTM and a Multivariate recurrent network. Satisfactory results were obtained when mean square prediction error was considered as the cost function. LSTM seems to provide the best performance.

As a perspective, we plan to study a combination of the experimented detectors.

REFERENCES

[1] K. Wang et al., “Wireless Big Data Computing in Smart Grid,” IEEE Wireless Communication, vol. 24, issue 2, pp. 58-64, April 2017, doi:10.1109/MWC.2017.1600256WC.

[2] M. B. Line, I. A. Tondel, and M. G. Jaatun, “Cyber Security Challenges in Smart Grids,” 2nd IEEE PES Int. Conf. and Exhibition on Innovative Smart Grid Technologies, Dec. 2011, pp. 1-8, doi: 10.1109/ISGTEurope.2011.6162695

[3] D. Liu, X. Zhang, and C. K. Tse, “A stochastic model for cascading failures in smart grid under cyber attack,” IEEE 3rd International Future Energy Electronics Conference, June 2017, pp. 783-788, doi: 10.1109/IFEEC.2017.7992139.

[4] M. Jamei et al., “Micro synchrophasor-based intrusion detection in automated distribution systems: Toward critical infrastructure security,” IEEE Internet Computing, vol. 20, issue 5, Sep-Oct 2016, pp. 18-27, doi: 10.1109/MIC.2016.102.

[5] M. Basseville and A. Benveniste (Eds), “Detection of Abrupt Changes in Signals and Dynamical Systems,” Lecture Notes in in Control and Information Sciences, LNCIS-77, Springer Verlag Berlin, Dec. 1985.

[6] R. Andre-Obrecht, “A New Statistical Approach for the Autoatic Segmentation of Continuous Speech Signals,” IEEE Trans. On Acoustics, Speech amd Signal Processing, vol. 36, issue 1, Jan. 1988, pp. 29-40, doi: 10.1109/29.1486.

[7] M. Basseville, “Detecting Changes in Signals and Systems – A Survey,” Automatica, vol. 24, issue 3, May 1988, pp. 309-326, doi: 10.1016/0005-1098(88)90073-8.

[8] C. Truong, L. Oudre, and N. Vayatis, “Selective Review of Offline Change Point Detection Methods,” Signal Processing, vol. 167, Feb 2020, pp. 1-20, doi: 10.1016/j.sigpro.2019.107299.

[9] H. Kim, K. Kim, S. Park, H. Kim, and H. Kim, “CoSimulating Communication Networks and Elecrical System for Performance Evaluation in Smart Grid,” Applied Sciences, vol. 8, issue 1, Jan. 2018, pp. 85-109, doi:10.3390/app8010085.

[10] “NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0,” NIST special publication 1108r3, NIST, Sep. 2014, doi: 10.6028/NIST.SP.1108r3.

[11] “Guidelines for Smart Grid Cybersecurity, Volume 1 – Smart Grid Cybersecurity Strategy, Architecture, and High Level Requirements,” NISTIR 7268 Revision 1, NIST, Sep. 2014, doi: 10.6028/NIST.IR.7628r1.

[12] H-J. Liao, C-H. R. Lin, Y-C. Lin, and K-Y. Tung, “Intrusion Detection System: A Comprehensive Review,” Journal of Network and Computer Applications, vol. 36, issue 1, Jan 2013, pp. 16-24, doi: 10.1016/j.jnca.2012.09.004.

[13] H. Sedjelmaci and S. M. Senouci, “Smart Grid Security: A New Approach to Detect Intruders in a Smart Grid Neighborhood Area Network,” Int. Conf. on Wireless Networks and Mobile Communications (WINCOM), 2016, pp. 6-11, doi: 10.1109/WINCOM.2016.7777182.

[14] W. Yu, D. Griffith, L. Ge, S. Bhattarai, and N. Golmie, “An integrated detection system against false data injection attacks in the smart grid,” Security and Communication Networks, Wiley and Sons, vol. 8, issue 2, Jan. 2015, pp. 91-109, doi: abs/10.1002/sec.957.

[15] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine Learning Methods for Attack Detection in the Smart Grid," IEEE Transactions on Neural Networks and Learning Systems, vol. 27, issue. 8, Aug. 2016, pp. 1773-1786, doi: 10.1109/TNNLS.2015.2404803.

[16] K. Smagulova, and A.P. James, "Overview of Long Short-Term Memory Neural Networks," In: James A. (eds) Deep Learning Classifiers with Memristive Networks. Modeling and Optimization in Science and Technologies, Springer, vol 14, Apr. 2020, pp139-153, doi:10.1007/978-3-030-14524-8_11.

[17] J. Toubeau, J. Bottieau, F. Vallée and Z. De Grève, "Deep Learning-Based Multivariate Probabilistic Forecasting for Short-Term Scheduling in Power Markets," in *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203-1215, March 2019, doi: 10.1109/TPWRS.2018.2870041.

[18] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily and J. Fuller, "Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework," Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2017, pp. 1-6, doi: 10.1109/MSCPES.2017.8064542.

[19] G. F. Riley and T. R. Henderson, “The ns-3 Network Simulator,” In: Wehrle K., Güneş M., Gross J. (eds) Modeling and Tools for Network Simulation. Springer, doi: 10.1007/978-3-642-12331-3_2.

[20] *GridLAB-D Simulation Software*. Available: https://www.gridlabd.org/[retrieved:May,2020].

[21] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," 2008 IEEE/PES Transmission and Distribution Conference and Exposition, Chicago, IL, 2008, pp. 1-5, doi: 10.1109/TDC.2008.4517260.

[22] *MATLAB - MathWorks - MATLAB & Simulink*. Available: https://www.mathworks.com/products/matlab.html/ [retrieved: May, 2020].

 ISBN: 978-1-61208-788-7