Effective detection of cyber attack in a cyber-physical power grid system.

OTOKWALA, U., PETROVSKI, A. and KALUTARAGE, H.

2021

Effective Detection of Cyber Attack In A Cyber-Physical Power Grid System

[1] Uneneibotejit Otokwala, [2] Andrei Petrovski, [2] Harsha Kalutarage
[1]u.otokwala@rgu.ac.uk; [2]a.petrovski@rgu.ac.uk; [2]h.kalutarage.ac.uk
School of Computing, Robert Gordon University, Aberdeen - UK

Abstract.

Advancement in technology and the adoption of smart devices in the operation of power grid systems have made it imperative to ensure adequate protection for the cyber-physical power grid system against cyber-attacks. This is because, contemporary cyber-attack landscapes have made devices' first line of defense (i.e. authentication and authorization) hardly enough to withstand the attacks. To detect these attacks, this paper proposes a detection methodology based on Machine Learning techniques. The dataset used in this experiment was obtained from the synchrophasor measurements of data logs from snort, simulated control panels and relays of a smart power grid transmission system. After the preprocessing of the dataset, it was then scaled and analyzed before the fitting of - Random Forest, Support Vector Machine, Linear Discriminant Analysis and K-Nearest Neighbor algorithms. The fitting of the different classifiers was done in order to find the algorithm with the best output. Upon the completion of the experiment, the results of classifiers were tabulated and the result of the Random Forest model was the most effective with an accuracy of 92% and a significantly low rate of misclassification. The Random Forest model also shows a high percentage of the true positive rate that is critical to the security issue.

.

Keywords: Cyber-attack Detection, True positive rate and Smart Grid system.

1 Introduction

The Purdue model for Industrial Control System (ICS) has bridged the gap between Information Technology (IT) and Operation Technology (OT) through the deployment of Wireless Sensor Network (WSN) and robots. As a result, the cyber-physical power grid system which is also known as the smart grid system has witnessed a tremendous advancement as Intelligent Electronic Device (IED) and other internet enabled devices have been incorporated into its structure for effective monitoring and value addition in its operations [1]. In fact, Cedric et. al., [2] had proposed that "*next generation of electric power grid system and other critical infrastructures will rely mainly on advanced technologies such as: industrial automation control systems, error diagnostics, preventive maintenance, automatic safety switching, advance metering infrastructure, and synchrophasor systems*". These advancements however, have exposed the system to a new vista of cyber-attack landscape which are clearly intended

to undermine the smart grid system, cause system misuse and obviate it from the critical role it plays in the society.

Cyber-attacks on the smart grid system occur when an unauthorized user leverages on the flaws and vulnerabilities of the devices to gain access to the internet enabled device. Some of the vulnerabilities include: weak passwords, unpatched firmware, weak encryption, insecure web links, etc. [3]. According to Alasdair Gilchrist [3], hackers have in recent times resorted to looking for older firmware to perform their attacks especially for versions with known vulnerabilities. For example, the power grid infrastructure system which were isolated and only run on proprietary softwares are now running on Commercial-of-the-Shelf (COTS) components and according to reports [4] [5], several cyber-attacks have been targeted against it because the COTS are not resilient enough and because the built-in safeguards against cyber-attacks are not properly hardened, maintained or updated [6]. It is also noteworthy that before now, most cyber-attacks were restricted to the IT infrastructure of critical organizations; however, with the convergence of OT and the IT infrastructure, there has been a significant shift in cyber-attacks to OT infrastructures [7] and these breaches often results in: reset of the phasor parameters, system shutdown, and disruption of the power grid system [6]. Usually, the Operating System (OS) provides the abstraction and support mechanism for the protection of hardware and application from misuse [8]; however, the cyber-attacks and threats especially, from non-state actors have assumed some level of sophistication in recent times. This therefore, makes the effective detection and prevention of cyber-attacks on the smart power grid system very important [9] [10].

1.1 Structure of a smart grid system.

A typical structure of a cyber-physical power grid system is shown in Fig.1 with the components.

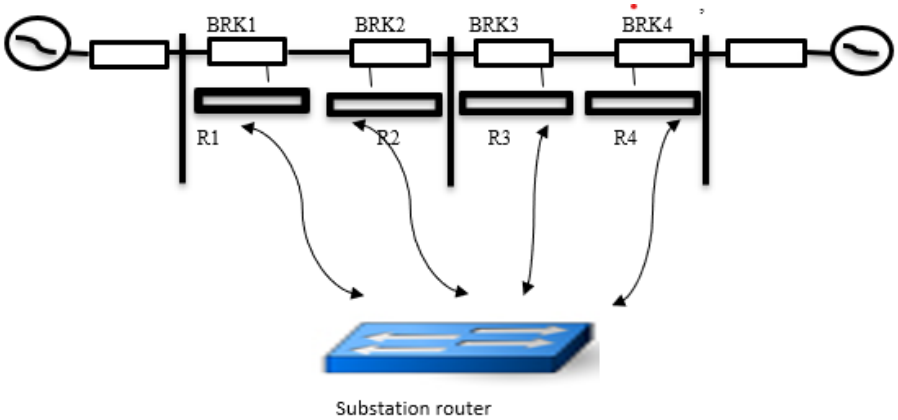

Fig. 1 Structure of a Cyber-physical Power Grid System [11]

A typical structure of a power grid system has power generators on both ends to supply electricity to the grid. The devices labeled R1, R2, R3 and R4 are Intelligent Electronic Devices (IEDs) which are connected to each circuit breaker, BRK1, BRK2 through BRK4. The role of the IED is to monitor events on the grid and to switch on or off the circuit breakers. According to the authors [11], there are two events that can cause the circuit breakers to trip and the events are: (a) an alert within the line (L1 and L2) that could initiate the IEDs to cause the breakers to trip (b) the operators manually issuing a command to the IED to break the circuit. In both instances, the intelligent devices, use a distance protection algorithm which enables the circuit breakers to trip irrespective of the cause of the command, i.e. whether it is a valid or invalid cause. Below is a list of events scenarios from the 2 mentioned above that can result in line tripping-

(a) Short-circuit fault – this is when there is a short circuit between two lines (two or more lines touching each other). This often results in very high voltage that could lead to massive damages.
(b) Line maintenance – this is when the line is intentionally disabled to allow for line maintenance.
(c) Remote tripping command – this is a possible attack in which an attacker breaches the device's defense and sends a command to a relay thereby causing a breaker to open.
(d) Relay setting change – this is another form of attack in which the attacker upon penetrating the device's defense, reconfigures the relay's setting and disables the relay function such that the relay will not trip even for a valid fault or a valid command.
(e) Data Injection – this is another form of attack in which the attacker upon entry, initiates a seeming valid fault by changing the phasor values of current, voltage, and other parameters just to ensure that the line trips.

It is apparent that from the scenarios highlighted above, successful attacks against the power system has the propensity to obliterate and render the power grid system incapable of providing efficient power. With these inadequacies and the insufficient scalability of the smart power grid system to mitigate the cyber challenges [12], there is a need to identify the cyber-attacks and secure the power grid system infrastructure.

1.2 Objective, Contribution and structure of paper

The objective of this paper is to find an effective cyber-attack detection model by fitting different machine learning classifiers on a simulated smart power grid system dataset. The results will then be compared and the most effective of the models will be tested for effective performance using different metrics. The effectiveness of the performance of our model will therefore be our contribution for effective intrusion detection of cyber-attacks in the smart power grid system. The rest of this paper is

organized as follows. Section II related literatures. Section III discusses data analysis. Section IV model fitting and performance evaluation. Section V is conclusion.

2 Related literatures

While a lot of research papers have been writing on the subject of intrusion detection in the smart grid system, a number of them appears static in their approach to intrusion detection especially in looking out for particular anomalous deviant behaviors. Considering that contemporary attacks on smart grid system have become dynamic, it therefore, requires that approaches should be dynamic and holistic such that detection could be effective even in multiclass situations. For example, cyber-attacks such as Relay Setting Change are common in smart grid system and they are often subtle and obfuscated in order to anonymize the attack. This kind of attack may likely not display an anomalous deviant behaviour to enable some of the proposed IDS detect. Here are some of the literatures -

2.1 Wide Area Monitoring System (WAMS).

This system was adapted through a concerted effort by several organizations to widely monitor power grids system in real-time within a "neighboring grids cluster". Basically, WAMS monitors the cyber-physical system parameters such as phasors of voltage, current, and the status of the IEDs, relays, circuit breakers etc. [2]. The real-time data so generated from the multiple remote points are then synchronized by the WAMS and then transmitted for measurements by the Phasor Measurement Unit (PMU). The PMU is a device used to estimate the magnitude and phase angle of the phasor parameters (voltage, current, etc.) in the electricity grid. The monitoring and synchronization is done in order to ensure accuracy whilst looking out for deviations and malicious values that could lead to down time resulting from attacks[13] [14].

2.2 Specification-based Intrusion Detection System.

Unlike the signature-based and anomaly-based Intrusion Detection Systems, the Specification-based IDS is a behavior-rule specification-based technique for intrusion detection that was introduced by Ko in 1996. It has its application mostly in medical cyber-physical systems, electrical cyber-physical grid system, software engineering and in network protocol of some critical infrastructures [2] [15]. In this IDS, the rules work by representing the system behavior of the state machine at every instance of time. According to Pan et al. [2], the state machine behaviors are represented by a sequence of states according to the policies specified. The devices are then monitored and tracked for intrusion, changes and anomalous behaviors that could drive the system state from safe to unsafe state. Any noticeable sequence of behaviors that are outside the predefined specifications are flagged as intrusion. In a nutshell, the authors averred that the Specification-based IDS can be likened to a complement of the anomaly-based IDS.

2.3 **Semi-Supervised Anomaly IDS**.

This is another form of behavior-based IDS which was proposed by Park et al [16]. Though this model was targeted at the Medical Cyber-Physical devices (MCPD) for assisted living environments, it could as well be adopted to detect anomalous behavior in the power grid system. Basically, this Semi-Supervised Anomaly IDS audits a series of events called, episodes. These episodes are sensor ID, start time and duration of events. In using the Hidden Markov Model (HMM) technique, a comparative analysis is then done to determine the current state of events and what happens thereafter. Based on the noticed behaviour, classification is then done by classifying the behaviour as low-level state or high-level state in order to be able to infer whether it is an abnormal or normal behavior.

2.4 **Data-Stream-Based Mining IDS**.

Faisal et. al. [17] proposed the use of Data-Stream-Based Mining IDS for the monitoring of intrusion in smart grid Advanced Metering Infrastructure (AMI). The structure of this IDS is similar to the anomaly-based IDS, but it selects a stream of data as against the conventional static mining techniques often observed in the anomaly technique. This proposal is, however, very limited in application to the smart meter. Therefore, this model is not suitable for intrusion detection of cyber-attacks in the cyber-physical smart power grid system.

3 Data Analysis

3.1 Description of dataset

The dataset used in this paper is the power system dataset [2][18]. It is made of 129 variables (128 predictors and 1 response variable of 3-classes). The dataset contains the measurement of electric transmission on a smart power grid system. These measurements were done using 4 synchrophasors which measures 29 features of the events in each Phasor Measurement Unit (PMU) totaling 116 features. The PMU uses a common time source to synchronize the various measurements and the features so measured were classified as attacks and benign data. The benign data is consisted of Normal traffic and NoEvents. These measurements were obtained using: snort, a simulated control panel, and relays. The parameters measured are: the voltage phase angle (PA1:VH – PA3:VH), the voltage phase magnitude (PM1:V – PM3:V), the current phase angle (PA4:IH – PA6:IH) and the current phase magnitude (PM4:I – PM6:I). Others are: the zero voltage phase angle (PA7:VH – PA9:VH), zero voltage phase magnitude (PM7:V – PM9:V), the zero current phase angle (PA10:VH – PA12:VH) as well as the zero current phase magnitude (PM10:V – PM12:V). In addition, there were also other parameters that were measured, and they are: frequency for relay (F), frequency delta (DF), appearance impedance for relays (PA:Z),

appearance impedance angle for relays (PA:ZH) and status flag for relays (S). Other descriptions in the dataset are fault location, line maintenance and load condition. The entire setup was aimed at measuring both the normal traffic transmission in the grid as well as the attacks (cyber intrusion) that could impact the power grid system.

3.2 **Dataset Class distribution**.

To enable us visualise the distribution of the instant classes in the response variable of our dataset, using RStudio Integrated Development Environment (IDE), we plotted a barplot of the values. See the plot in Fig. 2 and the R code in Appendix A.

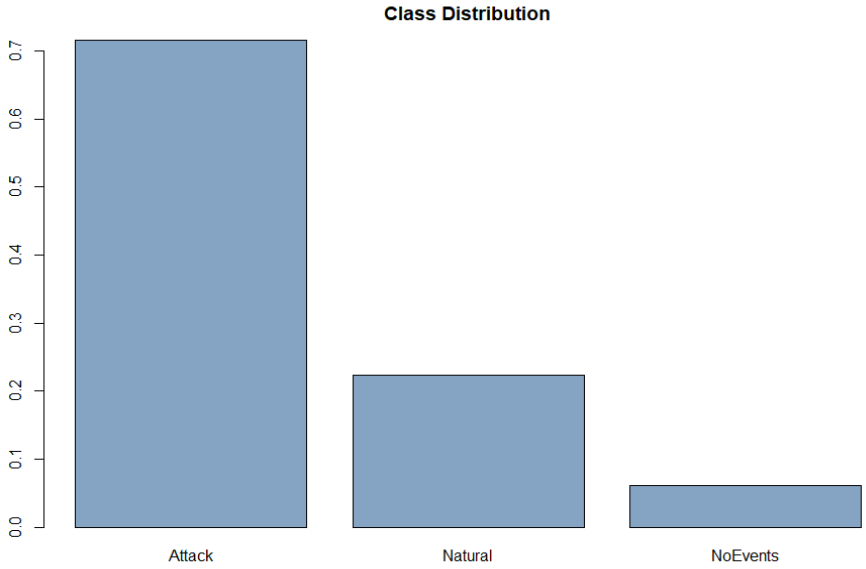

Fig. 2 Barplot showing the distribution of the instances of the response variable

A code snippet of the proportional representation of the classes in the response variable are as shown in Appendix B. Though the class representation and barplot shows the Attack class as the majority class over the benign class, the dataset does not fit into the description of an imbalanced dataset in cybersecurity considering the ratio between the classes. If we consider the dataset as binary (attacks and benign) then the ratio is 1 : 2. For attack to Natural it is 1 : 3 and for attack to NoEvent it is 1 : 11. In typical intrusion dataset, a ratio of 1 : 10 and above for a majority to minority class is expected before a dataset could be classified as an imbalanced dataset. More importantly, since our target class is the attack class, and it is a majority class, we elected to proceed with the dataset but with a view to ensuring that a higher recall rate is achieved and that the Area Under the Curve (AUC**)** for the ROC curves is high.

3.3 **Data pre-processing.**

Data cleaning and pre-processing is a way of preparing the dataset for eventual use and to also ensure that all the data points contribute to the model without bias. It involves outlier removal, feature selection and data normalisation. However, in our experiment, we only performed outlier removal and data normalisation using the scaling function.

Outlier removal. While summarising and visualizing the dataset, we observed that the dataset was fraught with outliers that needs to be removed. However, further introspection into the dataset shows that the anomaly was caused by fault of 10% - 19% on the relay of Line 1 which results in "Inf" values. The same outliers were found in Line 2, and relays number 3 and 4 of the power line. In addition, our observation also gave credence to the fact that these outliers may have been as a result of either the disabling of a single relay for line maintenance, remote tripping command of a single relay or a fault. In all the cases, the percentage of the disabling function lies between 10 – 29%. In view of these and the need to visualise the data points that clearly deviate from the others, we decided to use boxplot package of RStudio to visualise the outliers in the dataset [19]. From the plot (Fig. 3), data points that were discovered to significantly deviate from the rest of the points were identified and removed. As could be seen in the figure, the outliers are "Inf" and they were found in the following variables: "R1.PA.Z", "R2.PA.Z", "R3.PA.Z", and "R4.PA.Z".

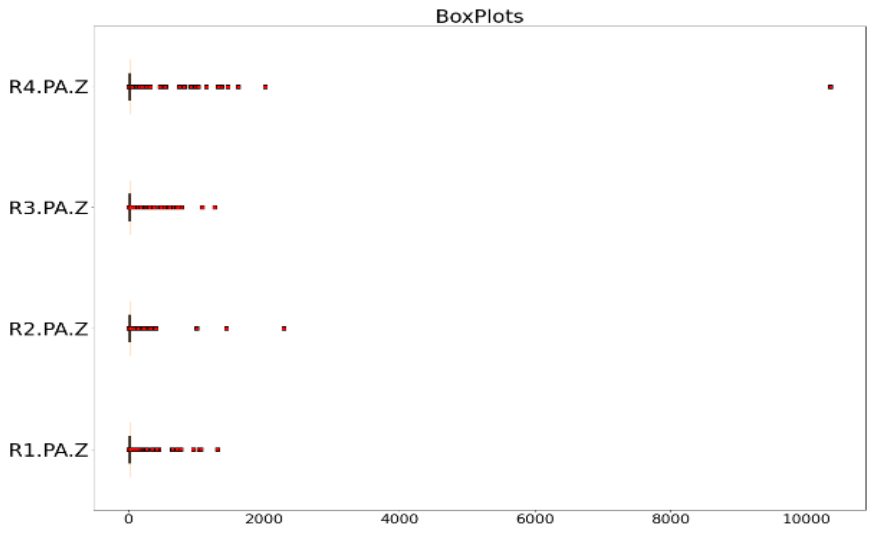

Fig. 3. Boxplot representation of values and outliers

Data Normalization. Data normalization during multivariate analysis is to enable each variable to contribute equally to the analysis. Therefore, the normalization method we

used was scaling, and we scaled from the first to the 128th variable leaving out the response variable which is a factor variable. Upon completion of the scaling, we then appended the response variable before we commenced the application of the classifier for modeling. See Appendix C for the code snippet on data scaling.

4. Model Fitting and Performance Evaluation

At this stage of our experiment, using a Windows 10 computing machine with intel core i5 processors and RStudio IDE, we applied some machine learning algorithms on the dataset. The essence was to fit several models and then compare the results of the models in order to determine which of them has the best accuracy, sensitivity and specificity. Also, our reason for using both linear and non-linear classifiers to fit the models was because, we observed that a few of the classifiers are highly likely to be biased toward the majority class in their output. However, before we applied the classifiers, we ensured that the dataset was clean of all factors that might affect our output. At this point, the total number of observations and variables after data pre-processing was, 52,885 – observations, and 129 - variables. We then partitioned the dataset into training and testing data and assigned 37,000 of the observations which constitute about 70% to training of the classifier. The remainder of the dataset which constitutes 30% of the observations was then used for validation. After the splitting, we went on to fit the model using the different classifiers.

4.1 Linear Discriminant Analysis (LDA).

The LDA [20] was the first classifier we used. It is a linear classifier that is robust and good at performing dimension reduction in the course of its application on datasets. It mostly works by dividing the data space into N number of disjoint regions such that probability densities are calculated with the assumption that the data is Gaussian with each attribute having same variance close to the mean. This classifier produced an accuracy of 71% with a high percentage of misclassification rate. Table 1 contains the values of the sensitivity and specificity of this classifier. Also find the R-code snippet for the model in Appendix D.

4.2 Support Vector Machine (SVM).

Support Vector Machine (SVM) [21] is a non-linear classifier that is used for both regression and classification problems. SVM produces significant accuracy with less computation power. To maximize the output and margin, SVM uses decision boundaries to classify data points that are closer to the hyperplane. These data points then influence the number of data points closed to the hyperplane, position and the orientation of the hyperplane. Our accuracy while using this classifier to fit our model was 72%. This model also showed a high percentage of misclassification rate hence our

desire to tune the kernel parameters in order to ensure improved performance. See Appendix E for the R-code and Table 1 for the value of sensitivity and specificity.

SVM Tuning.

Since the accuracy of our SVM model was not very high especially considering the high rate of misclassification, we decided to tune our SVM kernel parameters in order to improve the accuracy as well as reduce the Cost Matrix [21]. Usually, the SVM kernels takes data points as inputs and outputs similarity score that affects the class boundaries. The measure of the closeness on both sides of the hyperplane is the similarity and the nearer the data points are to the hyperplane, the higher the similarity score. We knew that to achieve a better SVM classifier output, it would require a better measure of closeness which can only be achieved through the right values of the kernel parameters. At this point, we then proceeded to try several values for gamma and cost with a view to having an optimal value that will yield a better accuracy and recall rate. We also applied the different kinds of kernel: Radial kernel, Polynomial kernel, Sigmoid kernel and Linear kernel. In the end, we were able to obtain a gamma value of 0.1 and cost parameter value of 20 in the radial kernel. With these values, we were able to tune the kernel parameters and obtained a better accuracy and a little reduction in the misclassification rate. With this tuning, we were able to improve the accuracy from 72% to 77%. However, we observed that the misclassification rate was still high hence the need for us to further apply some other non-linear classifiers. The sensitivity and specificity values have been provided in Table 1 and the R-code snippets are in Appendix F.

4.3 K – Nearest Neighbour (KNN)

The K-Nearest Neighbor (KNN) [22] is another non-linear classifier that we also used to model our work. KNN uses Euclidean distance to measure the distance between one data point and its neighbor. Based on the size of our dataset, we calculated the value of K as 192 and 193 (nearest neighbour), we then fit in the model and computed the confusion Matrix. The accuracy of the KNN model when it was fitted was 71% with a very high misclassification rate as the sensitivity and specificity were very low. See Table 1 for the values and Appendix G for code.

4.4 Random Forest

Random Forest (RF) [23] uses decision trees that are randomly created from selected data samples to make its predictions on each tree and then selects the best solution by means of voting. Usually, the more trees the classifier can create, the more robust the forest is. Its method of data splitting is an ensemble approach based on divide and conquer method. Individual trees are usually generated by the classifier using an attribute selection indicator. The application of Random Forest classifier to fit the model improved the accuracy of the model to 92% at 95% CI. Also, the model detection rate of the true positives (sensitivity) and specificity also improved. The improved

accuracy makes the model quite relevant for the detection of instances of attacks in a multiclass dataset as the one we are using. Furthermore, the balanced accuracy across the three instances were also very high which is an indication of suitability of the classifier for our experiment. It is also worthwhile to add that with a Kappa value of 82%, the model could be said to have performed very well in the identification and detection of the attack classes. See Table 1 for more on the detected values and Appendix H for a snippet of the code.

4.5 Experimental result comparison

We computed the Confusion Matrix of each of the classifiers and tabulated the values of the classes in Table 1. For the purpose of this experiment, we restricted the values to the computed Accuracy, Sensitivity and Specificity.

Table 1. Outputs of the confusion matrix of each of the models

	Accuracy	Sensitivity			Specificity		
		Attack	Natural	NoEvent	Attack	Natural	NoEvent
LDA	71%	99%	1%	6%	3%	99%	99%
SVM	71%	99%	0	4%	1%	99%	99%
Tuned SVM	77%	94%	28%	70%	39%	65%	98%
KNN	71%	100%	0	0	0	100%	100%
RF	92%	98%	68%	91%	73%	98%	99%

4.6 Confusion Matrix of best model

From the comparison of the values in Table 1, the output of the Random Forest model gave the best result of all the classifiers. In addition, the RF model also gave the lowest misclassification rate of all the models hence the confusion matrix in Table 2. The numbers along the diagonal represent the correct decisions made, and the numbers on the left and right of the diagonal represent the errors otherwise known as misclassification of the various classes. The confusion matrix code is in Appendix I.

Table 2. The Confusion Matrix of the Random Forest classifier

PREDICTED VALUES	ACTUAL VALUES		
	Attacks	Natural	NoEvents
Attacks	11202	984	56
Natural	142	2592	6
NoEvents	9	4	890

Recall and Precision. The recall otherwise known as Hit Rate or sensitivity is one of the metrics of measurement of the performance of a model. It is the number or proportion of the correctly predicted positive values divided by the total number of positive values (TP / (TP + FP)). False positives are values that our model incorrectly classified as positives but are actually negative values.

Attack **-** from the confusion matrix, from first column / row is

$$\text{Recall} = \frac{11202}{11202+142+9} = 98\%$$

$$\text{Precision} = \frac{11202}{11202+984+56} = 92\%$$

Natural - from the confusion matrix, from second column / row is

$$\text{Recall} = \frac{2592}{2592+984+4} = 72\%$$

$$\text{Precision} = \frac{2592}{2592+142+6} = 95\%$$

NoEvent - from the confusion matrix, from third column / row is

$$\text{Recall} = \frac{890}{890+56+6} = 94\%$$

$$\text{Precision} = \frac{890}{890+9+4} = 99\%$$

To further explain the value of our recall and precision - given all the predicted labeled class called Attack, the number of instances that were correctly predicted has a **precision = 0.92** (92%). Also, a **recall = 0.98** shows that for all instances that should have label Attack, our model correctly captured 98%.

F – Measure. F- Measure also known as F-score or F1 is another metric for the measurement of the accuracy of a classifier especially a dataset whose distribution of classes in the dataset is slightly skewed towards the majority class. Our dataset fits into this category hence our desire to also compute the F-score of our model. It is described

as the harmonic mean of the precision and recall as it is the most common metric that is used on an uneven or imbalanced classification problem.

$$F = 2X\frac{Precision*Recall}{Precision+Recall}$$

$$F = 2X\frac{92*98}{92+98} = 2X47.45 = 94.9 \approx 95\%$$

An F-score value of 1 indicates that the variance among the class mean is exactly what is expected given the within-classes variance and not by chance. Therefore, with our model's F-score tending to 1 (F1≈1), we can infer that the RF model was able to classify and detect the attacks. Also, considering our Confidence Interval of 95% with a significance of 0.05, the value of our computed P-value (see Appendix I) is less than the significance level (0.05) therefore, we can also infer that the value is statistically significant and supports the adoption of the RF model as suitable for detection of attacks.

4.7 Cutoff value, Receiver Operating Characteristics (ROC) and AUC

Cut-off value - The ROC curve is used to determine the optimum cut-off value especially as it shows the trade-off between the true positives and the false positive at different cut-off marks. Basically, it evaluates the hit rate and false alarm rate at varying thresholds (Figure 4) [24]..

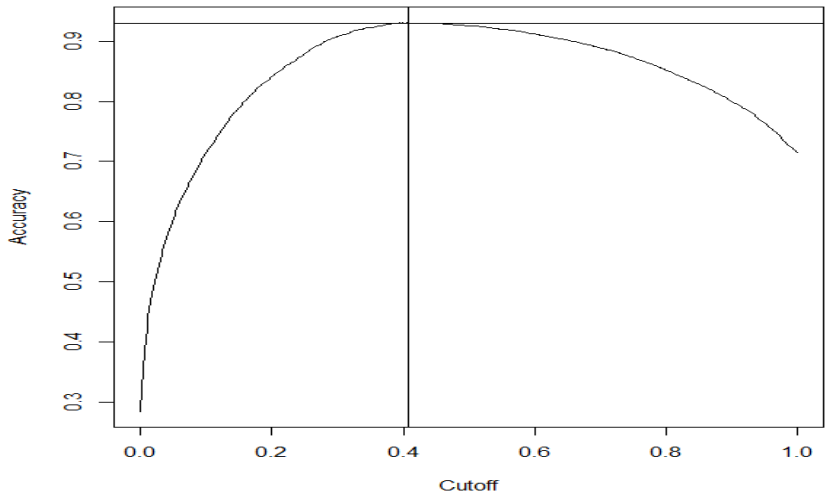

Figure 4. A plot showing the overall Accuracy values against several Cutoff values of the RF model.

From Figure 4, it can be observed that the accuracy of our model tends to increase with an increase in the cutoff values. However, at a maximum threshold value before the default cutoff (0.5), the model was able to achieve the maximum accuracy. The code snippet is in Appendix J.

ROC Curve – The ROC curve is a veritable tool for visualizing and evaluating classifiers performance accuracy and it is independent of the class distribution. ROC curve's ability to tend to the top-left corner of the graph indicates a better performance. Our RF model ROC from the graph (Figure 5) tends to the top left corner of the graph which is a pointer to the ability of the model to predict the true positive rates more correctly.

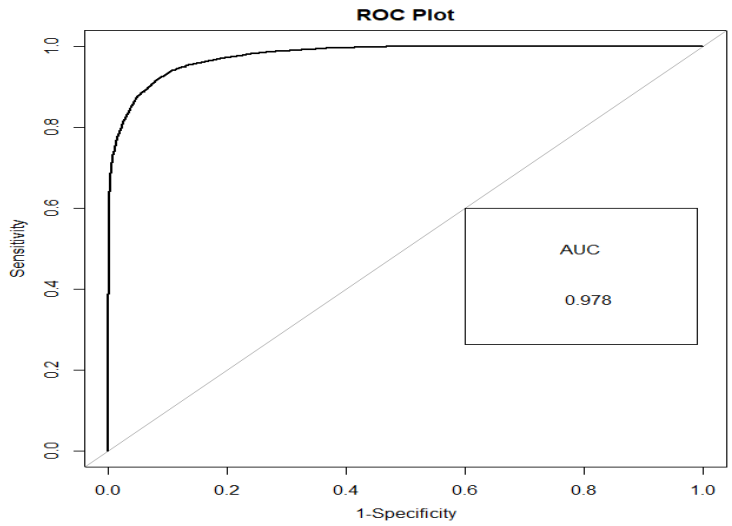

Figure 5: Showing ROC graph of sensitivity against 1-specificity and the area under the curve.

AUC - The area of ROC graph is 1 and its scale ranges from points 0.0 – 1.0. To measure the predictive accuracy of a model, the AUC of the curve needs to be computed as it is the probability that a given randomly chosen value is a positive instance of higher rank. An AUC of 0.5 indicates that the ROC curve lies on the baseline (the diagonal) where FPR = TPR which indicates that the predictive value of the AUC in the ROC curve is less accurate or at best the detection could only happen by chance. However, with our model AUC= 0.978, it is indicative that our RF model has a higher chance of detecting high positives.

5. Conclusion

In dealing with the growing integration and complexity of cyber-physical smart grid system, there is a necessity to explore an effective approach to detection, monitoring, optimizing, and more importantly, securing the smart power grid system. This paper has proposed an effective anomaly detection method against cyber-attacks in a smart grid system. Because the dataset we used has multiclass response variable, our focus was more on how to correctly classify and detect the true positive rate (Attacks) with a commensurate value of accuracy. The methodology we adopted to achieve this objective involved the application of several machine learning classifiers that will be able to provide a high accuracy as well as a high detection rate of the true positive rate. The classifiers we applied after necessary data cleaning and preparation were: Linear Discriminant Analysis, Support Vector Machine, K-Nearest Neighbor and Random Forest. Of all the classifiers, the Random forest model gave us the highest accuracy, a better detection rate of the true positives and also the specificity. We then went further to evaluate the performance of our model using metrics like precision, recall rate, F-score, ROC and Area Under the Curve. It is interesting to point out that all the metrics supported our model with very high probability for the detection of anomaly in a smart grid system.

6. Future work

The smart power grid system is experiencing a number of domain specific forms of cyber-attacks. These attacks include: data injection, remote tripping command injection, relay reset and others. Future works should look at identifying and classifying these forms of cyber-attack against the smart grid system infrastructure.

7. References

[1] C. Escudero, F. Sicard, and E. Zamai, “Process-Aware Model based IDSs for Industrial Control Systems Cybersecurity: Approaches, Limits and Further Research,” *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2018-Septe, pp. 605–612, 2018, doi: 10.1109/ETFA.2018.8502585.

[2] S. Pan, T. Morris, and U. Adhikari, “A specification-based intrusion detection framework for cyber-physical environment in electric power system,” *Int. J. Netw. Secur.*, vol. 17, no. 2, pp. 174–188, 2015.

[3] A. Gilchrist, *IoT security issues*. Walter de Gruyter GmbH & Co KG, 2017.

[4] G. Dondossola, J. Szanto, M. Masera, and I. N. Fovino, “Effects of intentional threats to power substation control systems,” *Int. J. Crit. Infrastructures*, vol. 4, no. 1–2, pp. 129–143, 2008, doi: 10.1504/IJCIS.2008.016096.

[5] T. Morris *et al.*, “Cybersecurity risk testing of substation phasor measurement units and phasor data concentrators,” *ACM Int. Conf. Proceeding Ser.*, 2011, doi: 10.1145/2179298.2179324.

[6] M. J. Haber and M. J. Haber, *Privileged Attack Vectors*. 2020.

[7] L. A. Maglaras *et al.*, “Cyber security of critical infrastructures.pdf,” *Elsevier*, vol. ICT Expres, pp. 42–45, 2018, doi: https://doi.org/10.1016/j.icte.2018.02.001.

[8] K. Mollus, D. Westhoff, and T. Markmann, “Curtailing privilege escalation attacks over asynchronous channels on Android,” *14th Int. Conf. Innov. Community Serv. “Technologies Everyone”, I4CS 2014 - Conf. Proc.*, pp. 87–94, 2014, doi: 10.1109/I4CS.2014.6860558.

[9] T. Wilhelm, “Chapter 10 - Privilege Escalation _ Elsevier Enhanced Reader.pdf,” in *Professional Penetration Testing*, Elsevier, 2013, pp. 271–306.

[10] D. N. Y. Conteh and M. D. Royer, “The Rise in Cybercrime and the Dynamics of Exploiting the Human Vulnerability Factor,” *Int. J. Comput.*, vol. 20, no. 1, pp. 1–12, 2016, [Online]. Available: http://www.ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/518/374.

[11] N. Events *et al.*, “Power System Attack Datasets - Mississippi State University and Oak Ridge National Laboratory - 4 / 15 / 2014,” no. 8, pp. 1–3, 2014.

[12] Y. Mo *et al.*, “Cyber-physical security of a smart grid infrastructure,” *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, 2012, doi: 10.1109/JPROC.2011.2161428.

[13] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, and G. C. Zweigle, “Smart generation and transmission with coherent, real-time data,” *Proc. IEEE*, vol. 99, no. 6, pp. 928–951, 2011, doi: 10.1109/JPROC.2011.2116110.

[14] W. Liu, Z. Lin, F. Wen, G. Ledwich, and S. Member, “A Wide Area Monitoring System Based Load Restoration Method,” *IEEE Xplore*, vol. 28, no. 2, pp. 2025–2034, 2013, doi: 10.1109/TPWRS.2013.2249595.

[15] R. Mitchell and I. R. Chen, “Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems,” *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 1, pp. 16–30, 2015, doi: 10.1109/TDSC.2014.2312327.

[16] K. Park, Y. Lin, V. Metsis, Z. Le, and F. Makedon, “Abnormal human behavioral pattern detection in assisted living environments,” *ACM Int. Conf. Proceeding Ser.*, 2010, doi: 10.1145/1839294.1839305.

[17] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, “Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study,” *IEEE Syst. J.*, vol. 9, no. 1, pp. 31–44, 2015, doi: 10.1109/JSYST.2013.2294120.

[18] S. Pan, T. Morris, and U. Adhikari, “Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems,” *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, 2015, doi: 10.1109/TSG.2015.2409775.

[19] C. C. Aggarwal, *Outlier analysis*, Second Edi., vol. 9781461463. Springer International Publishing, 2017.

[20] T. Gaber, A. Tharwat, A. Ibrahim, and A. Hassanien, “Linear Discriminant Analysis : A Detailed Tutorial,” *Univ. Salford, Manchester*, pp. 0–22, 2017, doi: http://dx.doi.org/10.3233/AIC-170729.

[21] B. Schlkopf, A. J. Smola, and F. Bach, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2018.

[22] P. Thanh Noi and M. Kappas, “Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery,” *Sensors*, vol. 18, no. 1, p. 18, 2018.

[23] B. Van Essen, C. Macaraeg, M. Gokhale, and R. Prenger, “Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA?,” in *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, 2012, pp. 232–239.

[24] T. Fawcett, “An Introduction to ROC Graphs,” pp. 861–874, 2005, doi: 10.1016/j.patrec.2005.10.010.

Appendix

A.

```
> boxplot(new_Data)
> #ploting  the response variable
> barplot(prop.table(table(new_Data$marker)),
+         col = rainbow(2),
+         ylim = c(0,0.7),
+         main = "Class Distribution")
```

B.

```
> cbind(freq=table(new_Data$marker),
+       percentage=prop.table(table(new_Data$marker))*100)
          freq percentage
Attack   37851  71.572279
Natural  11809  22.329583
NoEvents  3225   6.098137
```

C.

```
> #scaling of dataframe
> Norm_Data <- as.data.frame(scale(new_Data[1:128]))
> Norm_Data$marker <-new_Data$marker
```

D.

```
> set.seed(222)
> part <- sample(1:52885, 37000, replace = F)
> LDAtraining <- Norm_Data[part,]
> LDAtesting <- Norm_Data[-part,]
> fit_LDA <- lda(marker~.,data = LDAtraining)
Warning message:
In lda.default(x, grouping, ...) : variables are collinear
> fit_LDA_predict <- predict(fit_LDA, LDAtesting)
> confusionMatrix(table(fit_LDA_predict$class, LDAtesting$marker))
```

E.

```
> #SUPPORT VECTOR MACHINE (SVM)
> fit_svm <- svm(marker~., data = training)
Warning message:
In svm.default(x, y, scale = scale, ..., na.action = na.action) :
  Variable(s) 'control_panel_log3' and 'control_panel_log4' constant. Cannot scale data.
> pred_svm <- predict(fit_svm,testing)
> confusionMatrix(table(pred_svm,testing$marker))
```

F.

```
> tuneSVM <- tune.svm(marker~., data = training, gamma = seq(0.1,1,by=0.2),
+                     cost =c(1,20, by=2))

There were 31 warnings (use warnings() to see them)
> newsvm <- svm(marker~., data = training, gamma=0.1, cost=20)
Warning message:
```

G.

```
> caret::confusionMatrix(table(fit_KNN192, testMarker))
Confusion Matrix and Statistics

          testMarker
fit_KNN192 Attack Natural NoEvents
  Attack    11359    3565      961
  Natural       0       0        0
  NoEvents      0       0        0

Overall Statistics

               Accuracy : 0.7151
                 95% CI : (0.708, 0.7221)
    No Information Rate : 0.7151
    P-Value [Acc > NIR] : 0.504

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Attack Class: Natural Class: NoEvents
Sensitivity                 1.0000         0.0000          0.0000
Specificity                 0.0000         1.0000          1.0000
Pos Pred Value              0.7151            NaN             NaN
Neg Pred Value                 NaN         0.7756          0.9395
Prevalence                  0.7151         0.2244          0.0605
Detection Rate              0.7151         0.0000          0.0000
Detection Prevalence        1.0000         0.0000          0.0000
Balanced Accuracy           0.5000         0.5000          0.5000
>
```

H.

```
# using RandomForest
fit_randforest <-randomForest(marker~., data = training)
pred_randforest <-predict(fit_randforest,testing)
confusionMatrix(pred_randforest, testing$marker)
```

I.

```
> # using RandomForest
> fit_randforest <-randomForest(marker~., data = training)
> pred_randforest <-predict(fit_randforest,testing)
> confusionMatrix(pred_randforest, testing$marker)
Confusion Matrix and Statistics

          Reference
Prediction Attack Natural NoEvents
  Attack    11202     984       56
  Natural     142    2592        6
  NoEvents      9       4      890

Overall Statistics

               Accuracy : 0.9244
                 95% CI : (0.9202, 0.9285)
    No Information Rate : 0.7147
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8142

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Attack Class: Natural Class: NoEvents
Sensitivity                 0.9867         0.7240         0.93487
Specificity                 0.7705         0.9880         0.99913
Pos Pred Value              0.9150         0.9460         0.98560
Neg Pred Value              0.9586         0.9248         0.99586
Prevalence                  0.7147         0.2254         0.05993
Detection Rate              0.7052         0.1632         0.05603
Detection Prevalence        0.7707         0.1725         0.05685
Balanced Accuracy           0.8786         0.8560         0.96700
```

J.

```
A performance instance
  'Cutoff' vs. 'Accuracy' (alpha: 'none')
  with 502 data points
> max <- which.max(slot(eval, "y.values")[[1]])
> max
[1] 298
> acc <- slot(eval, "y.values")[[1]][max]
> acc
[1] 0.9305005
> cut <- slot(eval, "x.values")[[1]][max]
> cut
57449
0.408
> print(c(Accuracy=acc, Cutoff=cut))
    Accuracy Cutoff.57449
   0.9305005    0.4080000
```