Attack Detection in IEC 61850 Protocol using Artifical Neural Network.

Ketulkumar Polara, Jayesh Soni, Himanshu Upashyay
CNT 6150: Advanced Sensor & IoT Data Analysis With Deep Learning
Florida International University
Miami, Florida, USA

***Abstract*— Industries across the globe are going through new Industrial revolution, power utilities and Smart Grid are not an exception, where it is backboned by technologies like Internet of Things (IoT), and Artificial Intelligence (AI). With adaptation and rapid growth of IoT devices in Smart grid has led to increase in frequency and intensity of cyber-attacks. In this paper I am proposing Deep learning approach to detect cyber-attacks on IEC 61850 communication protocol, which is the new international standard for communication of Industrial Communication Systems (ICSs) in smart grids. In the paper I will start by describing working of IEC 61850 protocol, so to gain the domain knowledge aspect for AI driven solutions, and then move on towards the Deep learning methodology for attack detection. During the last decade, Artificial neural networks (ANN) has proven its potential to accurately classify labels, and its applications for every domain. Considering ANN's performance I will be training the network data collected on IEC 61850 to classify attack. The proposed approached has given me the accuracy of 99.98% for detecting the attacks.**

Keywords*— *IEC 61850, Internet of Things, Cyber-attacks, Artificial Neural Network, Machine Learning, Principal Component Analysis.

I. INTRODUCTION (*HEADING 1*)

IEC 61850 communication protocol is the new international standard because of its capability to handle the next generation modern Intelligent Electronics Devices (IEDs), protocols like Modbus and DNP3 can't quite handle the deployment flexibilities, Plus, IEC 61850 is ethernet-based and developed on object-oriented data model which makes the IEC 61850 an optimal choice for IEDs and substation communication. It has been agreed LAN connection are secure due to their nature and so far, known most of the IEC 61850 networks are based on LAN connection [1], but IEC 61850 also provides some mapping for web applications through client/server communication which widens the scope for cyber-attacks on the smart grids. For future substation automation, it is expected to be having a deep dependence on IEDs, and with this deep dependence it introduces security vulnerabilities in smart grids which can led to power outages and hardware failure/damages which ultimately increases the financial costs.

In this paper, the deep learning-based ANN model learns the difference between normal communication and attacks by utilizing the normal and attack network data collected on IEC 61850 by Petr Matoušek, Ondřej Ryšavý, and Peter Grofčík at Brno University of Technology, Czech Republic [4].

The rest of the paper follows the structure: Section II presents the background and inner working of IEC 61850 protocol. Section III presents the Deep Learning methodology to classify attacks. Section IV presents the results of the methodology and Section V concludes the analysis with some future goals.

II. IEC 61850 BACKGROUND

IEC 61850 was established in 2003 by Electro-Technical Commission's (IEC) Technical Committee 57 (TC57). IEC 61850 abstract data model can be mapped to number of protocols, but currently standard IEC 61850 is mapped to MMS (Manufacturing Message Specification), GOOSE (Generic Object-Oriented Substation Event), and SVM (Sampled Measured Values) where MMS protocol supports client/server communication over IP, GOOSE protocol utilizes Ethernet based multicast communication, and SVM, as name suggests Sampled Measured Values it carries power line and voltage values.

In this paper, my focus in on MMS (Manufacturing Message Specification) since it is based on client server architecture, and it widens the cyber vulnerabilities.

A. MMS (Manufacturing Message Specification)

is a messaging system for modeling real devices and functions and for exchanging information about the real device and exchanging process data – under real-time conditions – and supervisory control information between networked devices and/or computer applications. [3]

MSS communication adopts standards form ISO/IEC 9506-1 which provides service specification (Virtual Manufacturing Devices, services, and messages), and ISO/ IEC 9506-2 which provides protocol specifications (Rules, message sequence, format, encoding, and Interaction of MMS layers with other communication layers). [6]

As mentioned above MMS communication is based on client/server architecture model with object-oriented approach with object classes, instances, and methods. Where client can be any application or network device which ask server for action or data. A server hosts the Virtual manufacturing device (VMD) and its objects that client want to access. VMD is an object that contain all other objects. Where client requests a service which server responses accordingly.

Table 1 shows some of the MMS objects and their services

MMS Object	MMS Service
Application Process VMD	Initiate
	Conclude
	Abort
	Reject
	Cancel
	Identify
Named Variable Objects	Read
	Write
	InformationReport
	GetVariableAccessAttribute
	GetNameList
	GetNamedVariableListAttributes

Named Variable List Objects	GetNameList
	DefineNamedVariableList
	DeleteNamedVariableList
	Read
	Write
	InformationReport
Journal Objects	ReadJournal
	InitializeJournal
	GetNameList
Domain Objects	GetNameList
	GetDomainAttributes
	StoreDomainContents
Files	FileOpen
	FileRead
	ObtainFile
	FileClose
	FileDirectory
	FileDelete

Table 1: MMS Objects and services [3]

MMS PDU	TLV identifier	Tag Number
confirmed-RequestPDU	0xa0	0
confirmed-ResponsePDU	0xa1	1
confirmed-ErrorPDU	0xa2	2
unconfirmed-PDU	0xa3	3
rejectPDU	0xa4	4
cancel-RequestPDU	0xa85	5
cancel-ResponsePDU	0xa86	6
cancel-ErrorPDU	0xa7	7
initiate-RequestPDU	0xa8	8
initiate-ResponsePDU	0xa9	9
initiate-ErrorPDU	0xaa	10
conclude-RequestPDU	0x8b	11
conclude-ResponsePDU	0x8c	12
conclude-ErrorPDU	0xad	13

Table 2: MMS types and Tags [3]

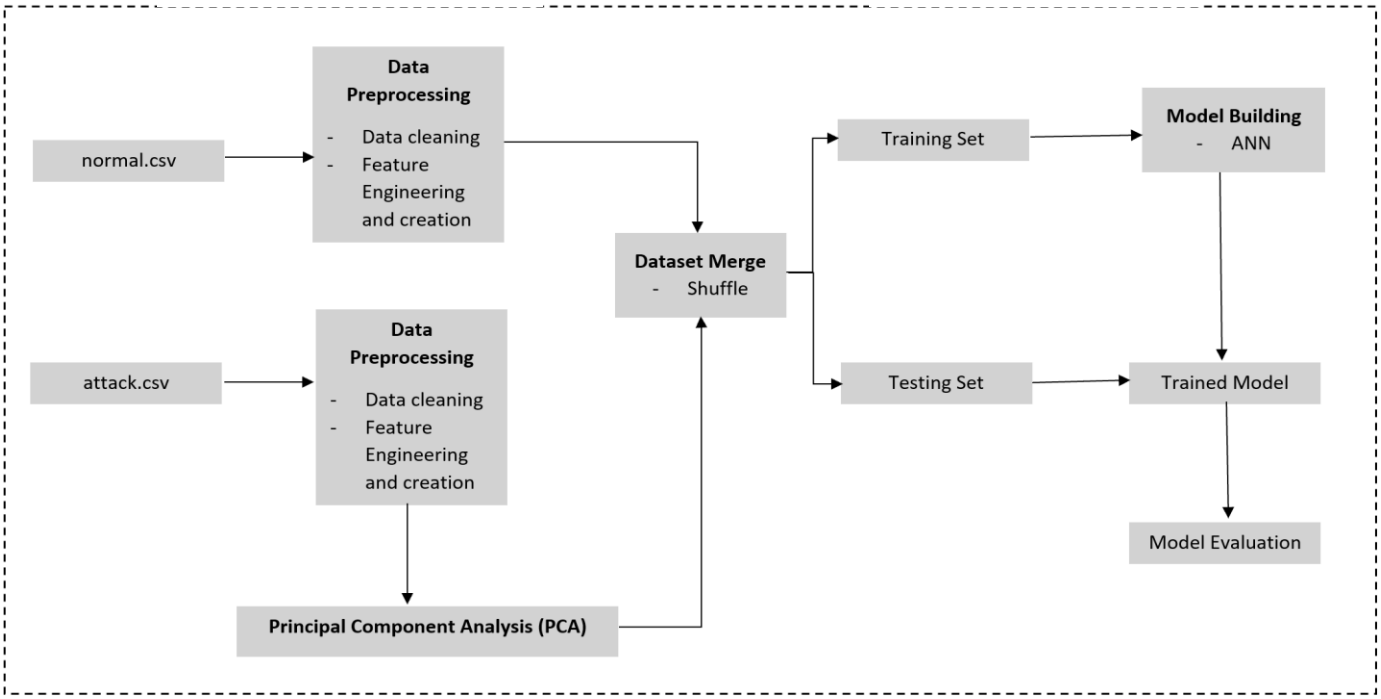

Figure 1: Proposed Methodology

III. PROPOSED METHODOLOGY

This paper was written using bottom down approach such that problem statement was formed based on the data available. The proposed methodology includes two major steps after data acquisition: data preparation, and model building for attack recognition. So, once we got the data, it must preprocess to make it ready for the next steps, data preprocessing steps contain another two steps of cleaning the data and performing dimension reduction process using Principal Component Analysis to prepare the data into right shape. After data preparation the final step is to train and test ANN model on the data prepared, hence detecting attacks. The whole process is shown in Figure 1.

A. Data Source and Description

Data used in the experiment was acquired from IEEE DataPort and was generated and uploaded by Petr Matoušek, Ondřej Ryšavý, and Peter Grofčík at Brno University of Technology, Czech Republic [4].

Original dataset DIR

Ics-dataset-for-smart-grids
--- but-iec104-i
--- but-iec104-ii
--- encs-iec104
--- encs-mms
---- inside_station_normal.csv
---- inside_substation_attack3.csv
--- gics-mms
--- rts-iec104
--- vrt-iec104

For this experiment, I will be focusing on the encs-mms folder where we have data on normal communication between only getNameList requests and responses which is inside the file inside_station_normal.csv whereas the file inside_substation_attack3.csv represents attack data where scanning attack was performed at 15:46:49 to read out all the IEC 61850 variables defined and so to get its values. plus, at 15:47:20 another attack was performed called write request where attacker started sending multiple write requests. Both .csv files contains 13 different features describing the network and MMS properties: Time Stamp (absolute time), Relative time (in sec, form start of capturing), Source IP address, Destination IP address, Source Port, destination Port, IP length (from IP header), MMS type (Represents MMS type shown in the table 2), MMS Service (Represents MMS Service shown in the table 1), Invoke Id (Represents truncation identifier), Domain Id (This feature is variable specific, relevant for write requests only), Item Id (This feature is also variable specific, relevant for write requests only), Object class (This feature is relevant for GetNameList MMS service request). For more information on MMS Petr Matoušek wrote a technical report [number].

B. Data Preprocessing

After acquiring data, next step is to prepare the data, generally depending on the dataset preprocessing step consists of Data cleaning, feature extraction, feature creation, feature engineering, etc. In this experiment, during this stage I performed two steps: data cleaning and feature engineering (and feature creation).

1. Data cleaning

The acquired network data consist of 13 different features representing network properties of IEC 61850 protocol. Since we have framed the problem as a classification problem, Removing the timestamp columns will not affect our analysis. Plus, redundant features like srcIP and dstIP were also removed from the dataset reason being this feature contributed same information as srcPort and dstPort for the final analysis. Additionally, I analyzed the whole dataset for missing (NaN) values, and it was found that Domain Id, Item Id and Object Class features contained missing values for both .csv files. The NaN values were handled by replacing it with string value of "no return" reason being Domain Id, Item Id, and Object Class registries values only when MMS service was "write request". I also dropped rows which contained missing values for Invoke Id feature for both the .csv files. The amount of NaN value rows for Invoke Id were very less compared to the whole dataset hence dropping it won't affect the analysis.

2. Feature Engineering and Feature Creation

After cleaning the data, it is important to prepare features into right datatype and format so to get best out of the model performance. As we have two .csv files representing two different categories normal and attack, and that's the classification problem we are trying to solve, I created target label column for both .csv file and assigned value of "0" for normal case and "1" for attack case. The acquired data contains 13 different features from which three features are discrete and with datatype of "object" which ANN's are not able to process it, so to handle this features I used One-Hot Encoding (Basically One Hot Encoding creates new column for each discrete value in the feature and assign 0 or 1 based on this occurrence in the observation(row)) which led us in creation of 3 more columns for normal.csv and 111 new columns for attack.csv. Next step in this stage is to merge normal and attack datasets into one, so to do, the number of features in both datasets has to match. To handle mismatch in shape of datasets I used Principal Component Analysis (PCA). Principal Component Analysis (PCA) is unsupervised Machine Learning algorithm with many use cases, in this experiment I am utilizing PCA's use case of Dimension reduction. PCA reduces the dimensionality of the data by replacing several correlated variables with a new set of variables that are linear combinations of the original variables [5], such that all the information is preserve. Additionally, I also normalized data using (1), so to make sure values in all the features are in same scale.

$$X' = \frac{X - Min(X)}{Max(X) - Min(X)} \times 255 \quad (1)$$

3. Train Test Split

After preparing data and merging the datasets, the next step is to split the dataset into training and testing set. In this regard, I randomly selected 80 percent of data for training set and rest of the data for testing set.

C. Model Building

Once the data is cleaned, prepared for the model, and organized into training and test set, I pass the training data into ANN model so that the model can learn from the data to distinguish between attack and normal operation. For the analysis I used ANN model where I have input layer with input size of 13 representing the number of dimensions, followed by two fully connected hidden layers with Dropout value of 0.2 and 32 nodes each, followed by an output layer with 2 output classes representing normal and attack cases. Further, we also need to tune other parameters like optimizer learning rate, number of epochs, and batch size. I tune the ANN model with batch size of 128 and used Root Mean Squared Propagation optimizer with learning rate 0.0001 as the optimizer for the model and iterate over 10 epochs for classification. Once the model was trained next step is to evaluate the model using testing set. For that we pass the testing set to the trained model and let the model predict the label for each observation. Evaluation and testing results are shown in the next section.

IV. RESULTS

The model was Evaluated using four most common evaluation metrics which are Recall, Precision, Accuracy, and F1-score.

Recall – It is also known as true positive rate, it evaluates the performance of the model to correctly classifies attacks in the case of attack. Mathematically, it is defined as (2):

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Precision – Precision evaluates the performance of the model as a ratio of truly classified attacks upon all the classified attacks. Mathematically, it is defined as (3):

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Accuracy – Accuracy evaluates the performance of the model by its ability to correctly classifying attack as "attack" and normal as "normal". Mathematically, it is defined as (4):

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \times 100 \quad (4)$$

F1-Score – F1-Score is defined as harmonic mean of precision and recall. Mathematically, it is defined as (5):

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (5)$$

	precision	recall	f1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00

Figure 2: Evaluation Report

After evaluating the proposed methodology, I was able to achieve the accuracy of 99.98%.

V. Conclusion and Future Work

In conclusion the performed experiment to detect attacks on IEC 61850 protocol using deep learning algorithm ANN was successful. The model created was able to detect attacks like scanning and write requests. Mentioned methodology was applied with optimal hyperparameter tuning and optimal ANN architecture, I also used Principal Component Analysis algorithm for dimensionality reduction for attack dataset. Plus, I tested the model and its performance using testing set created during the analysis. The ANN model resulted with the accuracy of 99.98% in identifying attacks.

Future work may include adding more attack cases to the existing analysis and conduct similar analysis on another map of IEC 61850 which is GOOSE messages.

VI. References

[1] A. Elgargouri and M. Elmusrati, "Analysis of Cyber-Attacks on IEC 61850 Networks," 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), 2017, pp. 1-4, doi: 10.1109/ICAICT.2017.8686894.

[2] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, "IoT DoS and DDoS Attack Detection using ResNet," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318216.

[3] P. Matoušek, "Description of IEC 61850 Communication" Bron University of Technology, Technical Report no. FIT-TR-2018-01.

[4] Petr Matoušek, Ondřej Ryšavý, Peter Grofčík, March 16, 2022, "ICS Dataset for Smart Grid Anomaly Detection", IEEE Dataport, doi: https://dx.doi.org/10.21227/1trw-n685.

[5]https://www.mathworks.com/help/stats/dimensionality-reduction.html#:~:text=Principal%20Component%20Analysis%20reduces%20the,combinations%20of%20the%20original%20variables.&text=Perform%20a%20weighted%20principal%20components%20analysis%20and%20interpret%20the%20results.

[6] Zhechen Huang, Lei Gao, Yi Yang, Xiangping Kong, Jinjiao Lin, Chapter 2 - IEC 61850 Standards and Configuration Technology, Editor(s): Yubo Yuan, Yi Yang, IEC 61850-Based Smart Substations, Academic Press, 2019, Pages 25-62, ISBN 9780128151587, https://doi.org/10.1016/B978-0-12-815158-7.00002-0. (https://www.sciencedirect.com/science/article/pii/B9780128151587000020) Keywords: IEC 61850; SCL language; Configuration; Conformance test