Intrusion Attack Detection In Power System Network

Project Report Submitted for the partial fulfilment of the requirement for the Degree of Bachelor of Technology in Electrical Engineering

Submitted By

SHRUTAYU NASKAR
SANTU DINDA
ARPITA BISWAS
FAJLE SARUP HOSSAIN

Under the Supervision of

Prof. DEBASISH BISWAS

Department of Electrical Engineering

Techno International New Town

New Town, Rajarhat

Kolkata- 700156

Maulana Abul Kalam Azad University of Technology, West Bengal

May, 2023

ACKNOWLEDGEMENT

We take this opportunity to express our thanks to our project supervisor Prof. Debasish Biswas (Assistant professor), Dept. of Electrical Engineering, Techno International New Town, for his invaluable support, guidance, active supervision, judicial criticism, constant help and encouragement without which it would not have been possible for us to complete the project in this shape.

Special thanks reserved for Prof. (Dr.) Milan Basu, HOD, for his generous help.

Above all, we would like to thank all the faculty members, staff members, our friends and junior fellows for their generous and spontaneous help.

Submitted by –

SHRUTAYU NASKAR

FAJLE SARUP HOSSAIN

SANTU DINDA

ARPITA BISWAS

Department of Electrical Engineering

Techno International New Town

Dated: 25/05/2023

Place: Kolkata

TECHNO INTERNATIONAL NEW TOWN

CERTIFICATE

I, hereby forward the project report entitled "INTRUSION ATTACK DETECTION IN POWER SYSTEM NETWORK" Prepared by ……

Shrutayu Naskar	**Registration No:022792**	**Roll No: 18701619053**
Fajle Sarup Hossain	**Registration No:022794**	**Roll No:18701619051**
Santu Dinda	**Registration No:022637**	**Roll No:18701619067**
Arpita Biswas	**Registration No:022446**	**Roll No:18701619086**

Under my guidance and supervision in partial fulfilment of the requirements for the Degree of Bachelor of Engineering in Electrical Engineering.

…… ……

Prof. Debasish Biswas **External Examiner**

Department of Electrical Engineering

……

Prof. (Dr.) Milan Basu

(Head of Department)

Department of Electrical Engineering

INDEX

Sl. No.	Topic	Sub-Domain	Page
1	Chapter 1: Abstract		5
2	Chapter 2: Introduction		6
3	Chapter 3: Literature Review	3.1. Machine Learning 3.2. CNN 3.3. Project Ideas 3.4. Key Ideas	7-9
4	Chapter 4: CNN	4.1. Convolution Operation 4.2. Pattern Recognition 4.3. Gradient Descent Algorithm 4.4. Back Propagation Algorithm	10-12
5	Chapter 5: Methodology		13-15
6	Chapter 6: Area of Project Problem Formulation	6.1. Network Experiment 6.2. Types of Scenarios 6.3. Test Bed Architecture 6.4. Training the IDS 6.5. Evaluation 6.6: Implementation of CNN in IDS of Power System	16-31
7	Chapter 7: Proposed Solution		32-33
8	Chapter 8: Future Plan		34
9	Chapter 9: Conclusion		35
10	Chapter 10: Reference		36

Abstract

The "Intrusion Attack Detection In Power System Network" project focuses on addressing the growing cybersecurity risks in power system networks. With the increasing interconnection and digitization of power systems, they have become susceptible to intrusion attacks that can compromise their reliability and security. This project aims to develop effective intrusion detection techniques tailored specifically for power system networks. The research investigates various approaches, including anomaly detection, signature-based detection, and machine learning algorithms, to identify and mitigate intrusion attacks. The project involves designing and implementing an experimental setup using real-world or synthetic power system data to evaluate the performance of the proposed intrusion detection techniques. The evaluation considers metrics such as detection accuracy, false positive rate, false negative rate, detection time, and computational resources required. The results obtained from the project contribute to enhancing the security of power system networks by providing insights into effective intrusion detection methods. The conclusion highlights the significance of intrusion detection in power systems, the practical implications of the developed techniques, and potential avenues for future research and improvement in power system network security.

Keywords:

1.Intrusion attack detection, 2. Power system network, 3. Cyber-security, 4. Interconnection, 5. Digitization, 6. Reliability, 7. Security, 8. Anomaly detection, 9. Signature-based detection, 10. Machine learning algorithms, 11. Experimental setup, 12. Real-world data, 13. Synthetic data, 14. Evaluation metrics, 15. Detection accuracy, 16. False positive rate, 17. False negative rate, 18. Detection time, 19. Computational resources, 20. Power system network security

Introduction

The core mission of power systems is resilience - continued delivery of electricity to the customer. These systems have been designed with the redundancy and fault tolerance mechanisms to perform this mission, but at a time when computer security was not a design driver. As formerly physically isolated power systems were joined to the Internet for centralized control and management, it created a greater potential for unauthorized access and exposed these systems to the same vulnerabilities that plague traditional computer systems and networks. Industrial control systems, such as those used in the Smart Electric Grid, are becoming more complex in their architecture and design. The Supervisory Control and Data Acquisition (SCADA) systems that are used are more interconnected and span multiple communication protocols and physical interfaces. The methods by which data are collected from remote locations, as well as commercially available SCADA software developed for physically isolated systems, lead to more potential flaws in the hardware and software and provide a much larger attack surface to threat agents. Every asset of the Smart Grid, from home gateways to smart meters to substations to control rooms, is a potential target for a cyberattack. Modern power systems are now connected to the Internet and computer security is a new threat to resilience. Power companies must now engineer security into their systems in arrears of the system design, or rely exclusively on traditional computer network defenses to prevent unauthorized access. Power system operators who monitor, assess, and react to disturbances must now consider the new possibility that the system is under a cyber-attack. This question is particularly challenging for a human to answer because, unlike natural disturbances or faults, a cyber-attack is designed to deceive. In this work, we explore the suitability of machine learning methods as a means of discriminating power system disturbances. We theorize that the machine learning algorithms will leverage non-linear complex relationships between power system measurements and that these will be sufficient to discriminate between malicious, non-malicious and natural disturbances. Cyber-attacks can have the same effects as natural events and so differentiating between malicious and non-malicious in a large and interconnected system can be overwhelming if not infeasible for a human. The intent of this work is to determine an optimal algorithm that is accurate in its classification such that it can provide reliable decision support to a power system operator, and thus relieve that operator of the burden of determining whether a disturbance is an intentional act. We evaluate the classification performance of various machine learning methods and discuss the implications for fielding machine learning systems and any associated operational constraints.

Literature Reviews

1. Machine Learning:

Machine learning is a field of study and practice in computer science that focuses on creating algorithms and models capable of learning and making predictions or decisions without being explicitly programmed. It involves developing and applying statistical techniques and computational algorithms to enable computers to analyze and interpret large amounts of data, identify patterns, and make accurate predictions or take informed actions.

In machine learning, instead of being explicitly programmed with specific instructions, computers are trained to learn from data and improve their performance over time. This is achieved by feeding the machine learning algorithms with labeled examples or training data, allowing them to identify underlying patterns and relationships within the data. These algorithms then use this knowledge to make predictions or decisions when faced with new, unseen data.

Machine learning has numerous applications across various fields, including image and speech recognition, natural language processing, recommendation systems, fraud detection, medical diagnosis, autonomous vehicles, and many more. It plays a crucial role in enabling computers to perform complex tasks and make intelligent decisions based on data-driven insights.

2. Convolution Neural Network:

Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly effective at analyzing visual data. It is inspired by the structure and functioning of the human visual system. CNNs are widely used for tasks such as image classification, object detection, and image recognition.

The key feature of CNNs is their ability to automatically learn hierarchical patterns and features from input data. They achieve this through the use of convolutional layers, which consist of filters or kernels that slide over the input data, performing convolution operations. These operations involve element-wise multiplication of the filter with the input data, followed by summation.

The main advantage of using convolutional layers is their ability to capture local spatial patterns, such as edges, corners, and textures, regardless of their location within the input data. This property makes CNNs well-suited for image processing tasks where the position of the features may vary.

After the convolutional layers, CNNs typically include pooling layers, which down-sample the output of the convolutional layers by selecting the most important features. This helps reduce the computational complexity and makes the network more robust to variations in the input.

Finally, CNNs often include fully connected layers, which connect every neuron from the previous layer to every neuron in the subsequent layer. These layers help in learning high-level representations and making predictions based on the extracted features.

By training a CNN on a large dataset of labeled examples, the network can learn to recognize and classify objects or patterns in new, unseen images. The training process involves adjusting the weights of the network to minimize the difference between predicted outputs and the true labels, using optimization algorithms like gradient descent.

Overall, CNNs have revolutionized the field of computer vision and have been instrumental in achieving state-of-the-art performance in tasks such as image classification, object detection, and image segmentation.

3. **Project ideas:**

Power system disturbances and cyber-attacks pose significant challenges to the reliable and secure operation of power grids. Traditional methods for detecting and mitigating these threats are often rule-based and rely on predefined thresholds or signatures. However, with the increasing complexity and sophistication of attacks, there is a need for more advanced techniques. Machine learning methods have emerged as promising tools for detecting power system disturbances and identifying cyber-attacks in real-time.

Machine learning algorithms can be trained on historical power system data to learn the normal operating behavior of the grid, including variations due to disturbances. These algorithms can then be applied to real-time data streams to detect anomalies that deviate from the learned patterns. By continuously monitoring the power system, machine learning models can identify abnormal conditions indicative of cyber-attacks or other disturbances, enabling timely intervention and response.

One practical implication of deploying machine learning systems in power system architectures is improved situational awareness. Machine learning algorithms can process and analyze large volumes of data from various sensors and sources, providing a comprehensive understanding of the power system's current state. This enhanced situational awareness enables operators to quickly identify and respond to abnormal events, improving the overall reliability and resilience of the grid.

Moreover, machine learning methods can assist in the early detection of cyber-attacks, which is crucial for minimizing their impact. By analyzing network traffic, system logs, and other relevant data, machine learning models can identify suspicious patterns or anomalies that may indicate a cyber-attack in progress. This early detection allows operators to take proactive measures to mitigate the attack, prevent further damage, and minimize the downtime of critical infrastructure.

However, deploying machine learning systems in power system architectures also presents challenges. One key challenge is the need for high-quality and diverse training data. The machine learning models must be trained on representative datasets that capture a wide range of normal and abnormal operating conditions. Obtaining such datasets can be challenging due to privacy concerns and limited availability of labeled data for rare events like cyber-attacks. Collaborative efforts between power system operators, researchers, and data providers are essential to address these challenges and ensure the effectiveness of machine learning systems.

Another consideration is the interpretability and explainability of machine learning models. Power system operators and regulators need to understand the reasoning behind the decisions made by these models. Efforts are being made to develop interpretable machine learning techniques that provide insights into the factors influencing the model's predictions, enabling operators to trust and validate the system's outputs.

In conclusion, machine learning methods offer valuable capabilities for detecting power system disturbances and identifying cyber-attacks in real-time. By leveraging historical data and advanced analytics, machine learning systems enhance situational awareness, enable early detection of threats, and support proactive response measures. However, careful attention must be given to data quality, interpretability, and collaboration to ensure the successful integration of machine learning into existing power system architectures, ultimately leading to a more secure and resilient power grid.

3.1. Key ideas:

I. Power system disturbances: *A. Complex nature B. Wide range of sources (natural and man-made events)*

II. Role of power system operators: *A. Heavy reliance on operators for decision-making B. Determining causes of disturbances C. Deciding appropriate response actions*

III. Challenges with cyber-attacks: *A. Human judgment less certain due to deceptive tactics B. Attempts to disguise the attack and deceive operators C. Difficulty in discerning the true state of the system*

IV. Machine learning for discrimination and detection: *A. Viability of machine learning as a solution B. Discriminating between types of power system disturbances C. Focus on detecting cyber-attacks with deception*

V. Evaluation of machine learning methods: *A. Assessing machine learning as disturbance discriminators B. Analyzing the effectiveness of machine learning in identifying and differentiating cyber-attacks*

VI. Practical implications for deploying machine learning systems: *A. Enhancing existing power system architectures B. Improved detection and discrimination of disturbances C. Enhanced situational awareness D. Proactive response measures E. Data quality and diversity challenges F. Interpreting machine learning outputs G. Collaborative efforts between operators, researchers, and data providers.*

4. CONVOLUTION NEURAL NETWORK (CNN)

CNNs, or ConvNets, are quite similar to regular neural networks. They are still made up of neurons with weights that can be learned from data. Each neuron receives some inputs and performs a dot product. They still have a loss function on the last fully connected layer. They can still use a nonlinearity function.

A regular neural-network receives input data as a single vector and passes through a series of hidden-layers. Every hidden layer consists of a set of neurons, wherein every neuron is fully connected to all the other neurons in the previous layer. Within a single layer, each neuron is completely independent and they do not share any connections. The last fully connected layer, also called the output layer, contains class scores in the case of an image classification problem.

Generally, there are three main layers in a simple ConvNet. They are the convolution layer, the pooling layer, and the fully connected layer.

CNN = Input layer + hidden layer + fully connected layer

We can see a simple neural network in the following image:

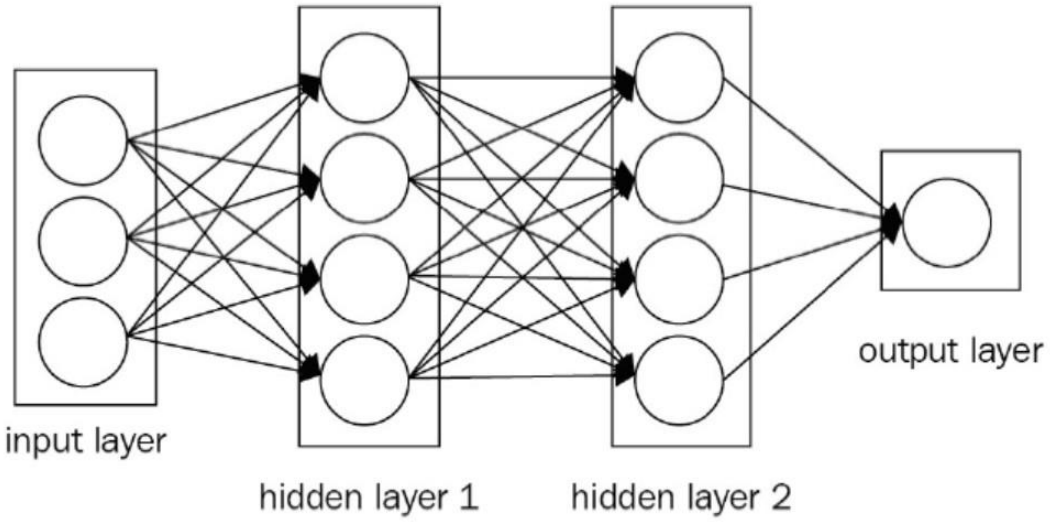

Fig 4.1: *A regular three-layer neural network*

4.1.Convolution Operation:

- We have some image or matrix (10*10) and feature detector (3*3)
- Feature detector = Kernel = Filter
- Feature detector detects features like edges or convex shapes.
- Feature map = Conv (input image, feature detector). Element wise multiplication of matrix.
- Feature map = Convolved feature
- Stride = navigate in input image
- We reduce the size of image. This is important because code runs faster.
- We create multiple feature maps because we use multiple feature detectors (Filter).

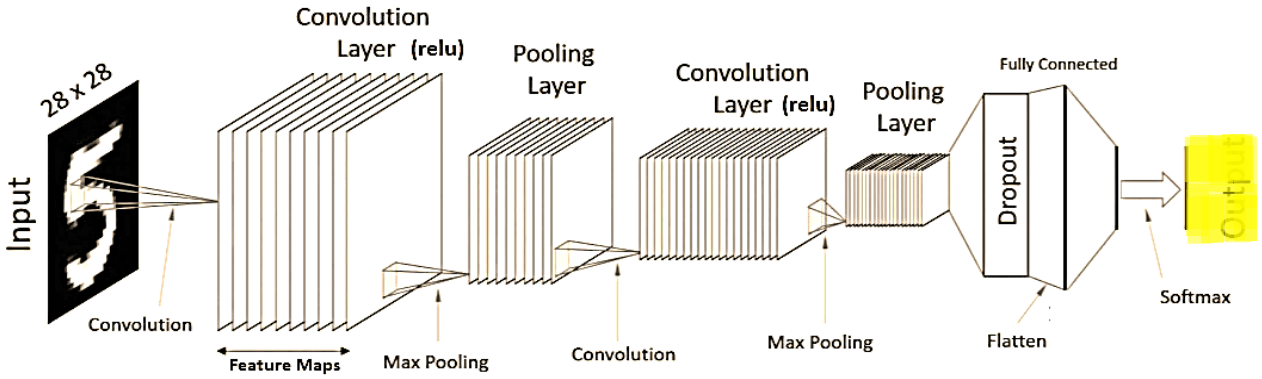

Fig 4.2: *Convolution neural network working steps*

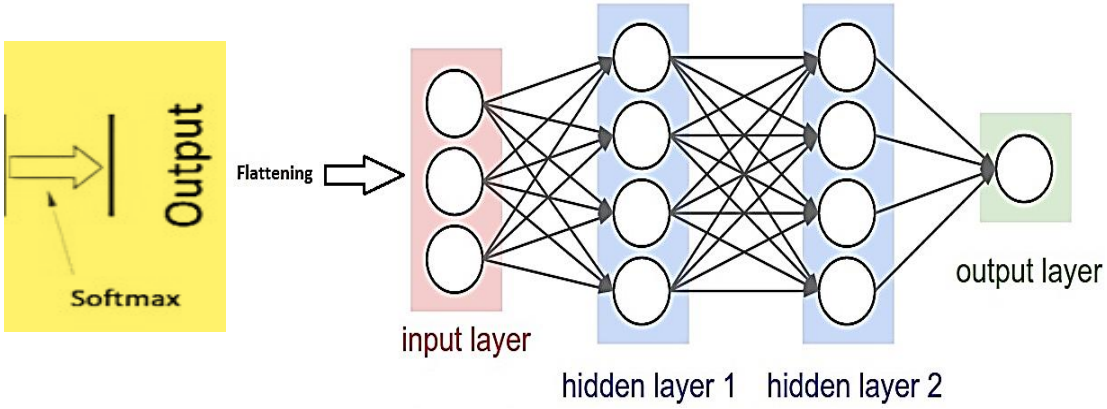

Fig 4.3: *Adjoint with fig 4.2, working steps of regular neural network*

4.2. Pattern recognition:

The task of feature extraction is performed by the computational units in the hidden layer(s) of the network.

The machine is designed as a feedforward network using a supervised learning algorithm.

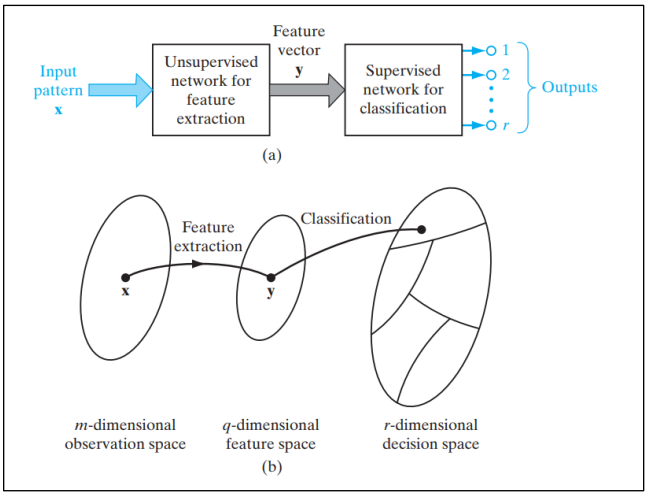

Fig 4.4: *Features extraction flowchart*

An artificial neuron is a function that takes an input and produces an output. The number of neurons that are used depends on the task at hand. It could be as low as two or as many as several thousands. There are numerous ways of connecting artificial neurons together to create a CNN. One such topology that is commonly used is known as a feed-forward network:

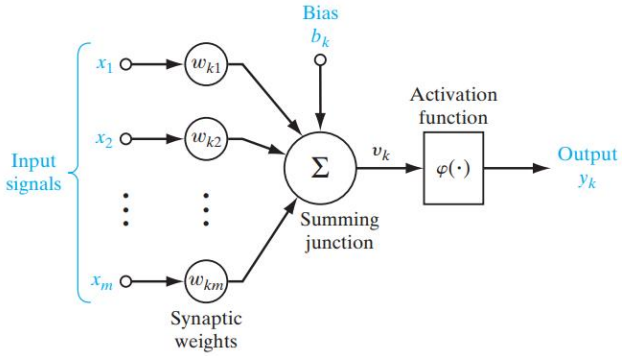

Fig 4.5: *neural network's summing junction operation*

Each neuron receives inputs from other neurons. The effect of each input line on the neuron is controlled by the weight. The weight can be positive or negative. The entire neural network learns to perform useful computations for recognizing objects by understanding the language. Now, we can connect those neurons into a network known as a feed-forward network. This means that the neurons in each layer feed their output forward to the next layer until we get a final output. This can be written as follows:

$$w_1x_1 + w_2x_2 + \ldots\ldots + w_nx_n \quad \ldots \text{(i)}$$

$$\sum_{i=1}^{i=n} w_ix_i = w_1x_1 + w_2x_2 + \ldots\ldots + w_nx_n \quad \ldots \text{(ii)}$$

4.3. Gradient Descent Algorithm

- This is the simplest training algorithm used in the case of a supervised training model. In case the actual output is different from the target output, the difference or error is find-out. The gradient descent algorithm changes the weights of the network in such a manner to minimize this mistake.

4.4. Back Propagation Algorithm

- It is an extension of the gradient-based delta learning rule. Here, after finding an error (the difference between desired and target), the error is propagated backward from the output layer to the input layer via the hidden layer. It is used in the case of Multi-layer Neural Network.

5. Methodology

Step 1: As we decide to make an Image for our project purpose which will be easily analyse by Machine Learning (CNN method). So, we need to make a matrix with the help of datasheet which form our desired image. Here, we arranged all the data in a (10*10) Input matrix and make our desired Image shape like "X".

Step 2: Now, from that matrix we identify and extract features which will help for convolution operation (i.e., multiply with each and every cell of input matrix and take their average value into another matrix), that increases data efficiency and decreases size of the matrix, called as Convolution layers.

Here, we extract 3 different types of (3*3) features matrix (RED, VIOLET, GREEN) from main (10*10) Input Matrix. After convolution operation we get 3 different types of (8*8) Convolution layers or matrixes for each and every feature map.

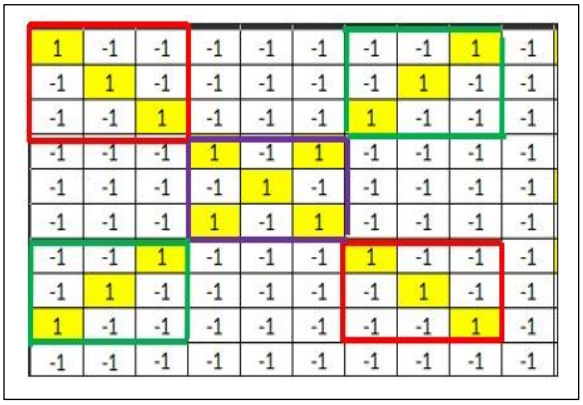

Fig 5.1: *Input Matrix (10*10)*

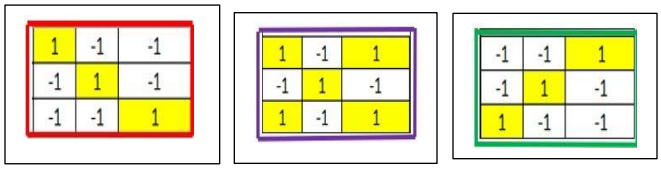

Fig 5.2: *(3*3) features map* (R, B, G)

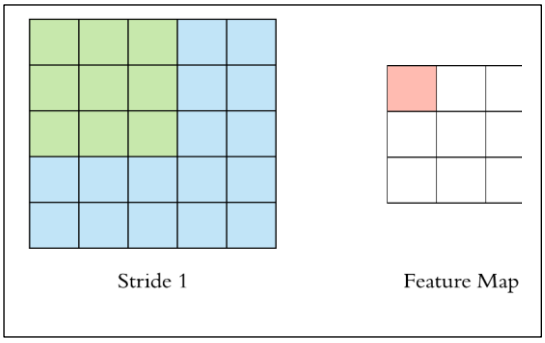

Fig 5.3: *Convolution operation with Input matrix by features matrix*

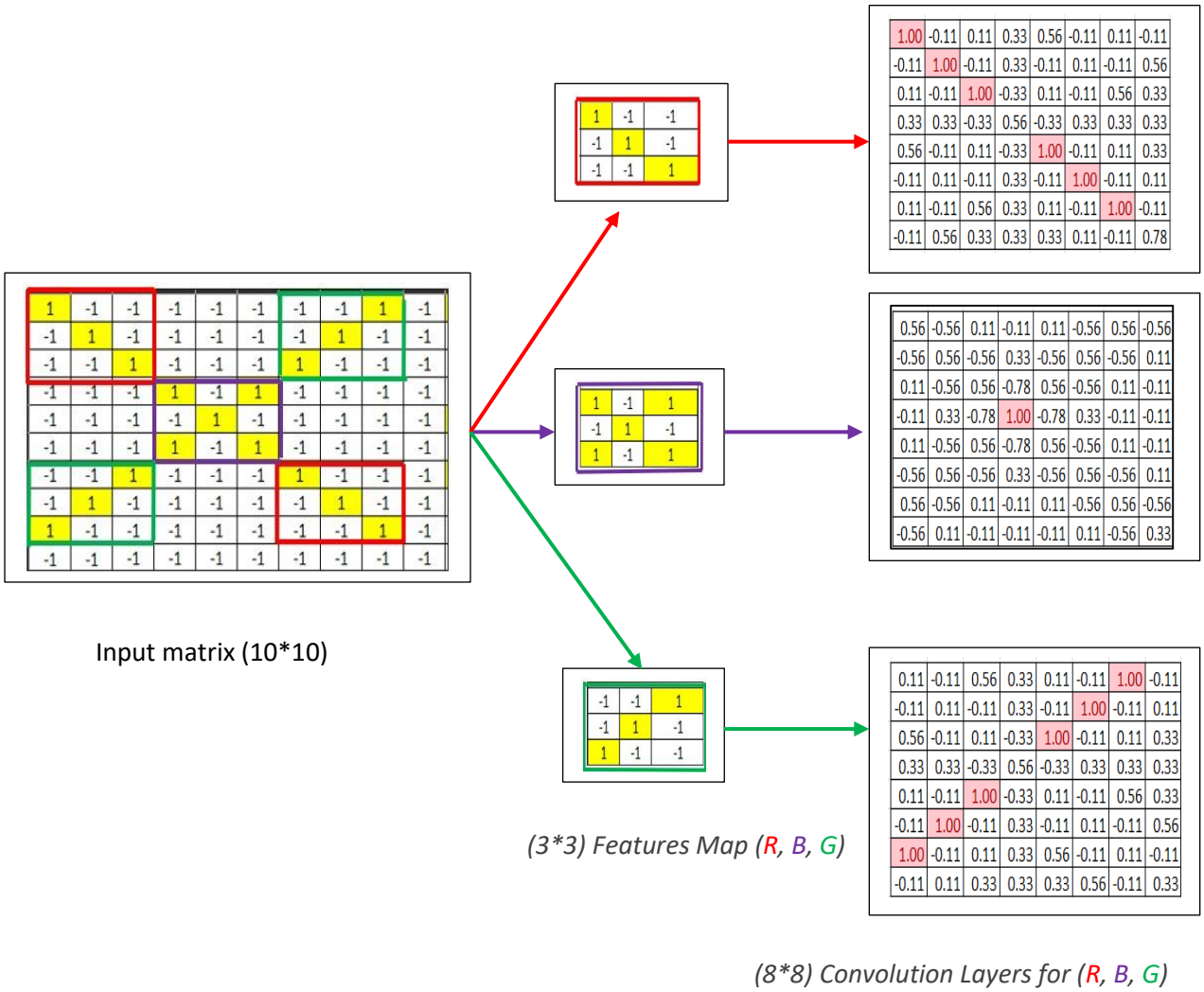

Fig 5.4: *Working steps from (Input matrix -> features maps -> Convolution layers)*

Step 3: After having convolutional layers we applied a ReLU function to brake-up the linearity. Because Images are always non-linear.

The Rectified Linear Unit (ReLU) is a nonlinear function that computes the function f(x)=max(0, x).

That means a ReLU function is 0 for negative inputs and x for all inputs x >0.
This means that theactivation is thresholded at zero.

$$relu(x) = \begin{cases} x \ \ if\ x >= 0 \\ 0\ if\ x < 0 \end{cases} \quad \text{... (iii)}$$

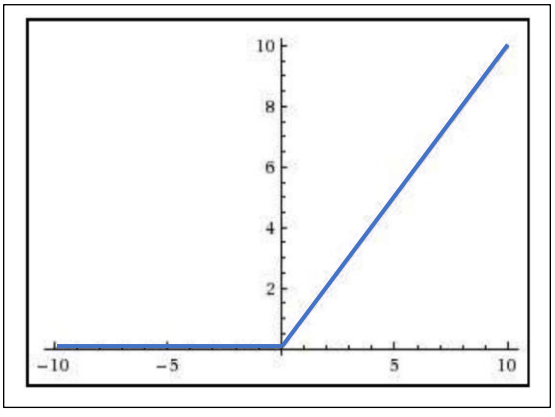

Fig 5.5: *Characteristics of ReLU function*

Step 4: So, as we applied activation ReLU function in all of three different convolution layers therefore, all the negative data transformed to zero or nearer to zero (in positive) and again we get another three different (8*8) convolution layers from previous each (8*8) convolution layer.

Step 5: Now, with the help of max pooling we take maximum value from a particular table of the new convolution layers. It makes down-sampling or sub-sampling. That's why reduce the number of parameters and makes another (2*2) matrix, called polling layers.

Step 6: After flatting all features or classifications of the (2*2) pooling layers now makes all the parameters as a vector form so that, they are applied into the next Neural Network as an Input.

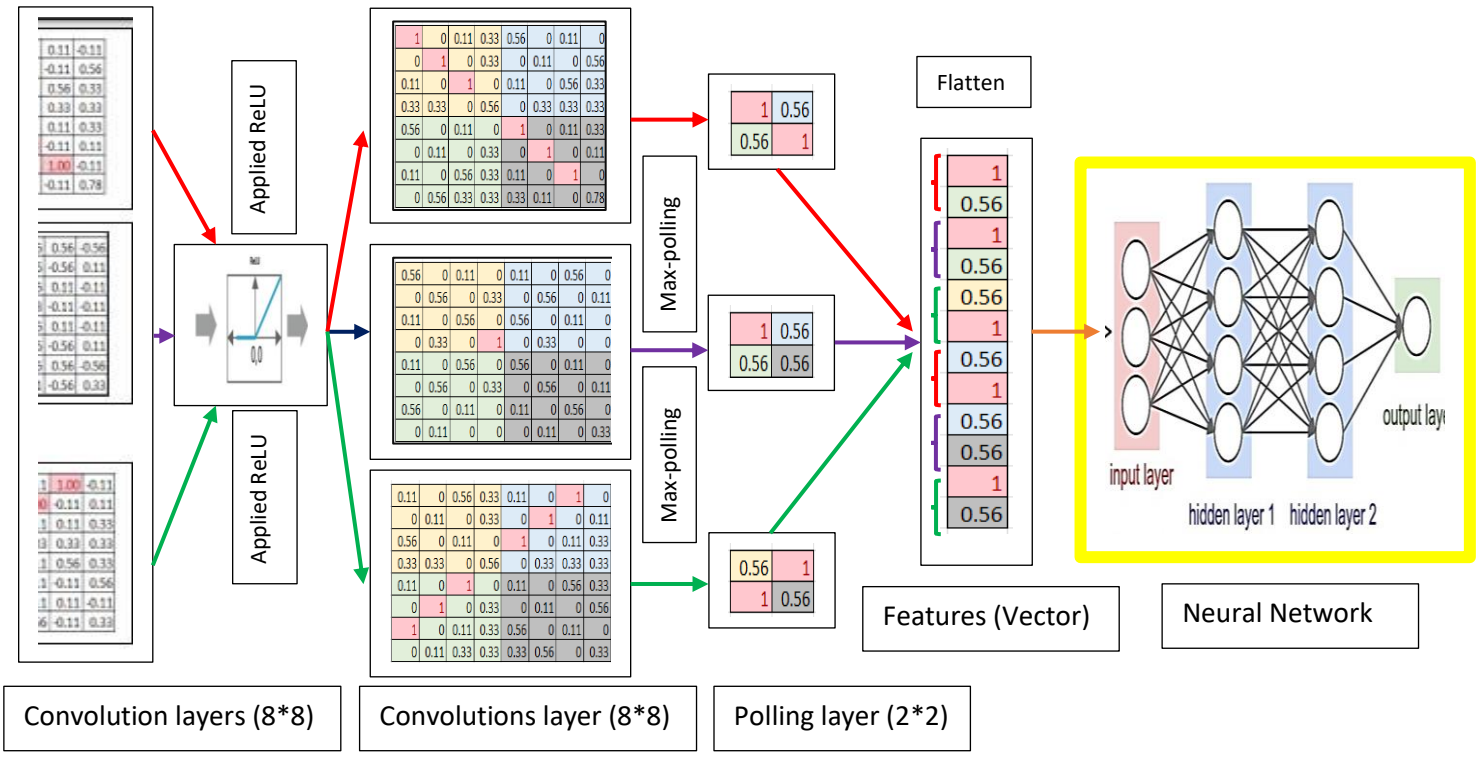

Fig 5.6: *Working steps from (Input matrix -> features maps -> Convolution layers -> Pooling -> Features(vector) > Input -> Neural Network -> Output)*

Step 7: Now our main task is to identify and match that whatever output comes through neural-network and our desired input Image that formed (10*10 input layers matrix).

If output image of the neural network is same as our desired input image ("X"), then we can say that our train and test method are mostly accurate. But if not then we can easily identify the contrast characteristics of input and output image, that makes the difference. Now, if we want to get our output image that came from neural network to form as the desired input image then the operation come, that called back propagation.

With the help of back propagation, firstly we need to travers from output to input of the neural network then need to changes or vary the value or weight of the neurons of the hidden layers until our output form our desired image that we want.

Hence, we conclude that with the help of this methods we can proceed to make our project. Where our project's main motto is to identify input parameters and output parameters are same or not. If not then we known that in power system any faulty condition happened. It may be natural fault or may be manmade or cyber-attack in our parameters. Also, we can mostly fix our problem with the help of back propagation, and proceed a smooth-running power system.

6. Areas of Project described

"Intrusion attack detection with the help of CNN methodology in power system network"

6.1. The figure below shows the power system framework configuration used in generating these scenarios.

- In the network diagram we have several components, firstly, **G1** and **G2** are power generators.
- **R1** through **R4** are Intelligent Electronic Devices (IEDs) that can switch the breakers on or off.
- These breakers are labeled **BR1** through **BR4**.

We also have **two lines L1** & **L2**. L1 spans from breaker one (**BR1**) to breaker two (**BR2**) and L2 spans from breaker three (**BR3**) to breaker four (**BR4**).

Each IED automatically controls one breaker.

R1 controls BR1, R2 controls BR2 and son on accordingly.

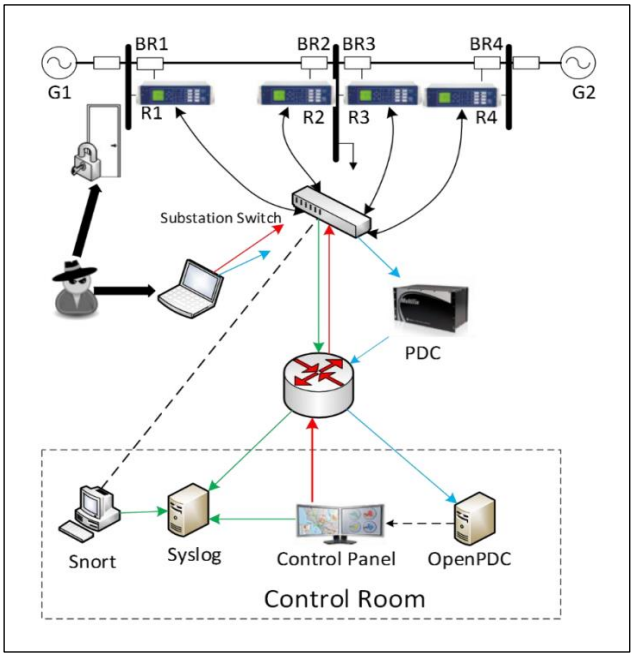

Fig 6.1: *Experiment network diagram*

The **IEDs** use a distance protection scheme which trips the breaker on detected faults whether actually valid or faked since they have no internal validation to detect the difference. Operators can also manually issue commands to the IEDs R1 through R4 to manually trip the breakers BR1 through BR4. The manual override is used when performing maintenance on the lines or other system components.

Problem Formulation

6.2 Types of Scenarios:

i. **Short-circuit fault –** this is a short in a power line and can occur in various locations along the line, the location is indicated by the percentage range.
ii. **Line maintenance –**one or more relays are disabled on a specific line to do maintenance for that line.
iii. **Remote tripping command injection (Attack) –** this is an attack that sends a command to a relay which causes a breaker to open. It can only be done once an attacker has penetrated outside-defenses.
iv. **Relay setting change (Attack) –** relays are configured with a distance protection scheme and the attacker changes the setting to disable the relay function such that relay will not trip for a valid fault or a valid command.
v. **Data Injection (Attack) –** here we imitate a valid fault by changing values to parameters such as current, voltage, sequence components etc. This attack aims to blind the operator and causes a black out.

- Tables I, II and III show the types of scenarios included.
- Table IV shows the distribution of instances in the binary classification group and
- Table V shows the distribution of instances in the binary classification group.

Natural Event	
Scenario	**Natural Events (SLG faults)**
1	Fault from 10 – 19% on L1
2	Fault from 20 – 79% on L1
3	Fault from 80 – 90% on L1
4	Fault from 10 – 19% on L2
5	Fault from 20 – 79% on L2
6	Fault from 80 – 90% on L2
	Natural Event (Line maintenance)
13	Line L2 maintenance
14	Line L2 maintenance

Table I: *Natural Event Scenarios*

Regular Operation	
Scenario	No Events (Normal Operation)
41	Normal Operation load changes

Table II: *No Event Scenarios*

Attack Event Scenarios	
Scenario	**Attack Type**
	Data Injection
	Attack Sub-type (SLG fault replay)
7	Fault from 10-19% on L1 with tripping command
8	Fault from 20-79% on L1 with tripping command
9	Fault from 80-90% on L1 with tripping command
10	Fault from 10-19% on L2 with tripping command
11	Fault from 20-79% on L2 with tripping command
12	Fault from 80-90% on L2 with tripping command
	Remote Tripping Command Injection
	Attack Sub-type (Command injection against single relay)
15	Command Injection to R1
16	Command Injection to R2
17	Command Injection to R3
18	Command Injection to R4

Attack Event Scenarios	
Scenario	**Attack Type**
	Attack Sub-type (Command injection against single relay)
19	Command Injection to R1 and R2
20	Command Injection to R3 and R4
	Relay Setting Change
	Attack Sub-type (Disabling relay function - single relay disabled & fault)
21	Fault from 10-19% on L1 with R1 disabled & fault
22	Fault from 20-90% on L1 with R1 disabled & fault
23	Fault from 10-49% on L1 with R2 disabled & fault
24	Fault from 50-79% on L1 with R2 disabled & fault
25	Fault from 80-90% on L1 with R2 disabled & fault
26	Fault from 10-19% on L2 with R3 disabled & fault
27	Fault from 20-49% on L2 with R3 disabled & fault
28	Fault from 50-90% on L2 with R3 disabled & fault
29	Fault from 10-79% on L2 with R4 disabled & fault
30	Fault from 80-90% on L2 with R4 disabled & fault

Attack Event Scenarios	
Scenario	**Attack Type**
	Attack Sub-type (Disabling relay function - two relays disabled & fault)
35	Fault from 10-49% on L1 with R1 and R2 disabled & fault
36	Fault from 50-90% on L1 with R1 and R2 disabled & fault
37	Fault from 10-49% on L1 with R3 and R4 disabled & fault
38	Fault from 50-90% on L1 with R3 and R4 disabled & fault
39	L1 maintenance with R1 and R2 disabled
40	L1 maintenance with R1 and R2 disabled

Table III: *Attack Event Scenarios*

	Attack Event	Natural Event	No Event
Scenarios	7,8,9,10,11,12,15,16,17,18,19, 20,21,22,23,24,25,26,27,28, 29,30,35,36,37,38,3 9,40	1,2,3,4,5,6,13,14	41

Table IV: *Three-class Classification groups*

	Attack Event	Normal Operation
Scenarios	7,8,9,10,11,12,15,16,17,18,19, 20,21,22,23,24,25,26,27,28, 29,30,35,36,37,38,3 9,40	1,2,3,4,5,6,13,14, 41

Table V: *Binary Classification*

The 128 features are explained in the table below. There are 29 types of measurements from each phasor measurement units (PMU). A phasor measurement unit (PMU) or synchrophasor is a device which measures the electrical waves on an electricity grid, using a common time source for synchronization. In our system there are 4 PMUs which measure 29 features for 116 PMU measurement columns total. The index of each column is in the form of "R#-Signal Reference"

that indicates a type of measurement from a PMU specified by "R#". The signal references and corresponding descriptions are listed below.

For example,

R1-PA1: VH means Phase A voltage phase angle measured by PMU R1. After the PMU measurement columns, there are 12 columns for control panel logs, Snort alerts and relay logs of the 4 PMU/relay (relay and PMU are integrated together). The last column is the marker. The first three digits on the right is the load condition (in Megawatt). Another three digits to their left are fault locations, for example, "085" means fault at 85% of the transmission line specified by scenario description. However, for those that do not involve fault, e.g., "line maintenance", these digits will be set to 000. The most left one digit or two digits indicate(s) the scenario number.

Feature	Description
PA1: VH – PA3: VH	Phase A - C Voltage Phase Angle
PM1: V – PM3: V	Phase A - C Voltage Phase Magnitude
PA4: IH – PA6: IH	Phase A - C Current Phase Angle
PM4: I – PM6: I	Phase A - C Current Phase Magnitude
PA7: VH – PA9: VH	Pos. – Neg. – Zero Voltage Phase Angle
PM7: V – PM9: V	Pos. – Neg. – Zero Voltage Phase Magnitude
PA10: VH - PA12: VH	Pos. – Neg. – Zero Current Phase Angle
PM10: V - PM12: V	Pos. – Neg. – Zero Current Phase Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA: Z	Appearance Impedance for relays
PA: ZH	Appearance Impedance Angle for relays
S	Status Flag for relays

6.3. TEST BED ARCHITECTURE

A. Distance Protection for Transmission Lines:

The distance protection scheme is the most popular scheme for protecting transmission lines. The principle of operation recognizes that the impedance of a high-voltage transmission line is approximately proportional to its length. This means the impedance "seen" by the relay during a fault is proportional to the distance between the point of fault and the relay. Distance relays are encoded with multiple protection zones. Each zone is assigned an apparent impedance threshold and a trip time. Relays have over lapping protection zones to provide system protection redundancy. One relay's zone 1 is part of another relay's zone 2 and so forth.

For this case study, the distance protection scheme was simplified by disabling reverse time delay back up and limiting the number of protection zones for each relay to 2. (Fig. 6.3.1) shows a three-bus two-line transmission system that is modified from four-bus three-generator system. Relay R1's zones 1 and 2 are shown as dashed line boxes.

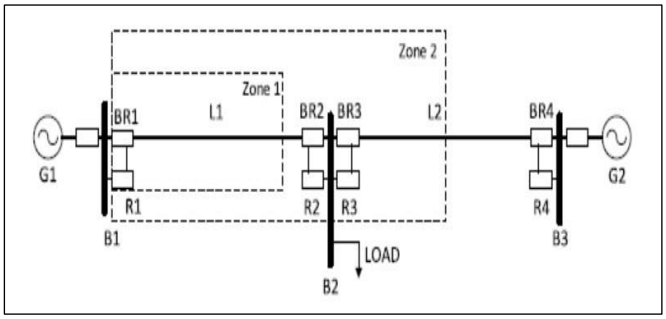

Fig 6.3.1: *Distance protection scheme in a three-bus two-line transmission*

Each relay provides primary protection up to 80% of the line (zone 1 protection) and backup protection (zone 2 protection) up to 150% of the line in case that the primary protection fails. The trip time for zone 1 protection is configured to be instantaneous while the trip time for the zone 2 protection is time-delayed to avoid false tripping unless the primary relay fails.

B. Test Bed Architecture

A hardware-in-the-loop test bed, shown in (Fig 6.3.2), was used for power system scenario implementation and data generation. A real time digital simulator (RTDS) was used to simulate transmission lines, breakers, generators, and load. Four physical relays were wired to the RTDS in a hardware-in-the-loop configuration. The relays implemented the two zones distance protection scheme. The relays trip and open the breakers when a fault occurs on a transmission line.

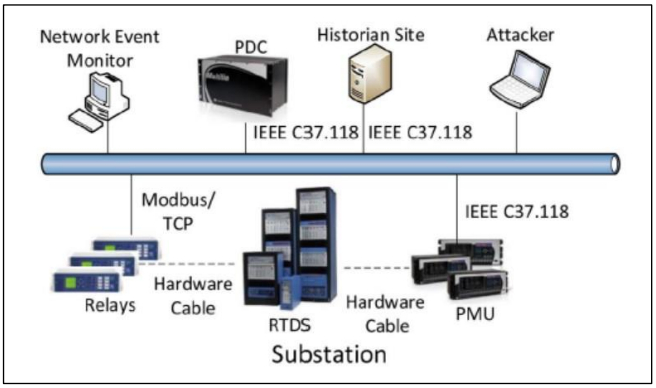

Fig 6.3.2: *Hardware in the loop test bed*

All relays included integrated phasor measurement unit (PMU) functionality to measure power system transmission line state, however, the PMUs were drawn separately in the graph because relays are controlled by Modbus/transmission control protocol (TCP) and PMUs stream synchrophasor measurements using the IEEE C37.118 protocol. The PMUs streamed real-time synchrophasor measurement data at a rate of 120 samples/s, to the phasor data concentrator (PDC) which aggregates network frames from multiple PMU and forwards the aggregated synchrophasor frames to the Open-PDC application. A set of scripts control the simulation by inducing random state changes, capturing measurements, labeling captured data by scenario type, and merging data from multiple sources into a single file. The synchrophasor measurement data includes of frequency, current phasors, voltage phasors, and sequence components. The four relays were sources of time stamped relay state changes. A signature-based IDS, Snort, runs on a PC to detect network activity. Snort provides alerts when it detects remote tripping command activities in the network. Snort, by itself, cannot distinguish between legitimate and illegitimate remote trip commands since they appear the same on the network.

A control panel computer simulates energy management system (EMS) functionality. The EMS simulation was used to disconnect a transmission line for maintenance by remotely tripping relays via a Modbus/TCP network packet. An EMS log provides the TS of such a line maintenance event. For this paper, it is assumed that an attacker computer has successfully penetrated the utility's operational network and can launch cyber-attacks from a node on the operational network. Scenarios of power system disturbances, normal operations, and power system cyber-attacks are applied against the simulated power system and its components. Data logs were captured from the synchrophasor system, relays, Snort, and the simulated EMS. All data logs were time stamped and with the name of the scenario being simulated.

C. Test Bed Scenarios

The power system scenarios used to train and validate the IDS presented in this paper have been grouped into three categories:

a. power system single-line-to-ground faults;
b. normal operations; and
c. cyber-attacks.

Each category is described in this section with details. There are a total 25 scenarios each named with capital "Q" along with a number. The system load was randomized at the beginning of each scenario. Power system SLG faults belong to the shunt fault family and account for up to 70% of faults in a power system. For this paper, only phase-*a*-to-ground faults were simulated as each phase to ground fault has similar characteristics. The phase-*a*-to-ground fault is abbreviated as "fault" in the rest of this paper. Table II provides a summary of the simulated scenarios used to validate the proposed IDS.

For the SLG fault scenarios (Q1 and Q2) the relay operates instantaneously for zone 1 and after a time delay for faults in zone 2. The auto-reclosing scheme models a high speed three-phase reclosing scheme which closes the breaker after one second. The SLG fault replay attacks (Q3 and Q4) attempt to emulate a valid fault by altering system measurements followed by sending an illicit trip command to relays at the ends of the transmission line. This attack may lead to confusion and potentially cause an operator to take invalid control actions. A python script was used to initiate a MITM attack between the hardware PDC and the OpenPDC application. The attacks replay synchrophasor measurements from a valid SLG fault hen replay commands to trip the relays on the affected line. The transmission line maintenance scenarios (Q5 and Q6) simulate the situation when an operator remotely trips relays to open breakers at both ends of a transmission line to take the line out of service for line maintenance.

Scenario Name	Description
Q1-Q2	Single line-to-ground fault on L1 or L2 respectively.
Q3-Q4	Fault replay attack which mimics a valid fault on L1 or L2 respectively.
Q5-Q6	Remotely open both relays at both ends of transmission line (L1 or L2 respectively) for maintenance.
Q7-Q10	Command injection attack to remotely open one relay (R1, R2, R3, R4 respectively)
Q11-Q12	Command injection attack to remotely open two relays (R1 and R2, or, R3 and R4 respectively). This attack mimics the maintenance scenarios (Q5-Q6).
Q13-Q16	One relay (R1, R2, R3, R4 respectively) disabled during a fault on the line connected to that relay.
Q17-Q20	One relay (R1, R2, R3, R4 respectively) disabled during a maintenance event on a line connected to that relay.
Q21-Q22	Two relays (R1, R2, R3, R4 respectively) disabled during a fault on the line connected to those relays.
Q23-Q24	Two relays (R1, R2, R3, R4 respectively) disabled during a maintenance event on a line connected to that relay.
Q25	Normal system operation. No event occurring.

Table 6.3.1: *SIMULATES SCENARIOS*

The transmission line maintenance scenarios (Q5 and Q6) simulate the situation when an operator remotely trips relays to open breakers at both ends of a transmission line to take the line out of service for line maintenance. The operator initiated remote trip commands are recorded and time stamped in the control panel log.Power system cyber-attacks may originate from insiders, amateur hackers, political activists, criminal organizations, governments, and terrorists. Cyber-attacks may appear as a nuisance or may bring the system to collapse. Attacks can be carried out from within power system substations, a control center, or in transmission and distribution infrastructures by exploiting weaknesses in physical security policies. Alternatively, attacks may take advantage of

security flaws and vulnerabilities in software, devices, communication infrastructures, or communication protocols to electronically infiltrate power system operational networks.

Three types of attacks are simulated:
1) relay trip command injection;
2) disabling relay
function; and
3) SLG fault replay.

Relay trip command injection attacks (Q7–Q12) create contingencies by sending unexpected relay trip commands remotely from an attacker's computer to the relays at the ends of the two transmission lines. The trip command injection attack used for this paper closely mimics the line maintenance scenario. The malicious trip command originates from another node on the communications network with a spoofed legitimate IP address. Since the attack is not from the control panel computer there will be no record in the control panel log, however, the Snort network traffic monitor will detect this remote trip command. The disabled relay attacks (Q13–Q24) mimic the effects of insiders taking illicit control actions or malware taking control of software systems to manipulate control devices. A python script accesses a relay's internal registers via Modbus/TCP commands sent from the attacker's computer which modify the relevant relay settings. The disabled relay attacks overlap fault and maintenance events. The final scenario, Q25, represents a stable system state. For this scenario, the load may change, but no other attacks, disturbances, or control actions are simulated. Scenarios start and end with the system in a stable state. As such, all faults are cleared, transmission lines taken out of service for maintenance are returned to service, and all attacks end before the next scenario is simulated.

D. Test Data

Test data used for this paper includes data logs associated with 10 000 simulated instances of the 25 aforementioned scenarios. The data log is a comma separated file with labeled tuples that include 56 sensor measurements and a TS. The 56 data sources consist of 52 synchrophasor measurements; 13 from each relay location on (Fig 6.3.1). The synchrophasor data from a single relay consists of phase voltage and current phasor magnitude, zero, positive, and negative sequence voltage, and current phasor and apparent line impedance. The synchrophasor data was sampled at 120 times/s. Relay status information, breaker events, Snort alerts, and control panel alerts were also logged. All logged data was merged into a single dataset. An instance of a single scenario is represented by approximately 2000 tuples in the test data set. This corresponds to approximately 17 s of simulated system time per scenario.

In total, the test data has more than two million tuples. Each tuple in the test data is labeled. Approximately, half of the test data was used to train the classifier and half was used to test classification accuracy. For this paper, 15 features were used; phase current magnitude measured at each relay, relay status for each relay, Snort alert status for each relay, and control panel remote trip status.

6.4. TRAINING THE IDS

This section documents the IDS construction process. First, the data formatting step converts input data logs to a measured events database (MED). Next, the specification learning steps process the MED to learn common paths, a unique set of system states in temporal order, for each labelled scenario. Finally, a graph is constructed which includes common paths for all scenarios.

A. Data Formatting

The first step of the data formatting process is featuring quantization. Feature quantization requires domain expertise. Features with values which can take continuous values are mapped into finite ranges to limit state space size. Features which take discrete values are generally left unchanged unless the number of discrete values is large. The phase current measurement is a real number and therefore should be grouped into discrete ranges. Phase current magnitude was separated into normal and high ranges. The normal range was 0–1199 A. The high range was all values greater than or equal to 1200 A. The relay status, Snort alert, and control panel remote trip status features are all binary. Possible relay status values are tripped and not tripped. Possible Snort alert status values are alert and no alert. Possible control panel remote trip status values are tripped and not tripped.

The MED is a merged compressed data set with quantized features. Data from sensors with lower sample rates is up sampled to match the sampling rate of the sensor with the highest sampling rate. The up-sampling process depends upon the sensor type. Continuously sampled sensors update their value at each sample period-based upon the current measured state. The current magnitude and relay status are continuously sampled. Event-based sensors provide a single message when a state change occurs. The Snort alert and control panel remote trip status features are event-based. For each, when the sensor detects the presence of an event the sensor provides a message indicating event occurred. In a data log, a continuously sampled sensor measurement takes a value and holds that value across multiple samples until the state changes.

Conversely, in the data log, event-based features are asserted for a single sample for each measured event. When up sampling, continuously sampled sensor measurements are mapped to the nearest sample period after the measurement. All samples without a value take the value of nearest preceding sample. Event-based sensor measurements are also mapped to the nearest sample period after the measurement. All samples without a value take the nonasserted value. For this paper, the current magnitude measurements were measured at 120 samples/s which is the highest sampling rate of all features. Relay status, Snort alerts, and control panel log features were up sampled according to the aforementioned procedure.

An MED represents one instance of a scenario. The TSs of rows in the MED are normalized by subtracting the time of the first row from all other rows. This causes all MEDs to start from time0.

B. Creating and Grouping Paths

A path is a list of observed system states arranged in temporal order. Paths are extracted by down-sampling the MED while preserving all state transitions. A state change is a change on any sensor value between two MED samples. The MED is parsed to identify all periods of consistent state. Consistent state periods are down-sampled using a user defined sample period. For this paper, the sample period was 0.5 s.

Each unique state is assigned a state identifier (*S*id) and all known states are stored in a state data base.
A path is extracted for each MED. A single scenario will have many unique paths due to the dynamic nature of power systems, variations in the order of states within a path, and due to variations in event timing. Using the raw paths derived from the extraction process for classification results in poor classification accuracy. The common path mining algorithm is used to shrink the larger group of paths into a representative set of common paths which represent normal variation and serve as a set of signatures for each scenario. Grouping is an optional step which pre-processes input data to separate large classes into smaller sub-classes.

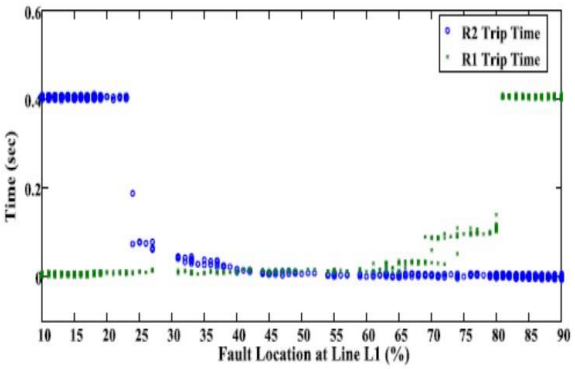

Fig 6.4.1: *Relay trip time versus fault location for relays R1 and R2.*

Grouping can lead to more accurate classification when the sub-classes are sufficiently different from one another. (Fig 6.3.3), clearly shows zones 1 and 2 trip boundaries for both relays. Additionally, Fig 6.3.3. shows that the relay trip times vary with fault location especially in the fault location region from 24% to 79% of the transmission line. The relay trip time for Fig 6.3.3 was calculated from the MED as the time relay status is transitions from closed to open minus the initial time the line current equals is high. System behaviour also varies as the system load changes. Ideally, instances of SLG faults from a two-zone distance protection scheme can be separated into three groups according to the area of the line in which the fault occurs. Group 1 includes faults from the length of the line which is protected by relay R1's zone 1 and relay R2's zone 2. From Fig 6.3.2, group 1 includes faults which occur between 10% and 23% of the line. For group 1 faults, relay R1 should trip instantly and R2 should trip after 0.4 s. Group 2 includes faults protected by relay R1's and R2's zone 1. Both relays should trip instantly for group 2 faults. From Fig 6.3.3, group 2 faults occur between 24% and 79% of the line. Group 3 includes faults protected by relay R1's zone 2 and relay R2's zone 1. Relay R1 should trip after 20 cycles and R2 should trip instantly for group 3 faults. From Fig 6.3.3, group 3 faults occur between 80% and 90% of the line.

Observed trip times in group 2 tend to increase as the fault approached the zones 1 and 2 boundary points. To compensate for this observed behaviour the SLG fault paths were grouped by fault location per the following groups: 10%–23%, 24%–29%, 30%–35%, 36%–40%, 41%–60%, 61%–65%, 66%–70%, 71%–80%, and 81%–90%. Additionally, it was observed that trip times partially correlated to the system load.

As a result, the SLG fault paths were grouped by fault location and load. Four load ranges were used: 200–249, 250–399, 300–349, and 350–399 MW. This grouping subdivided the SLG fault paths into $9 * 4 = 36$ sub-groups.

C. Common Path Mining

For this experiment the set **G** consists of 5000 raw paths from 5000 instances of the 25 scenarios. The common path mining algorithm produced 477 common paths across all scenarios. The minimum and maximum number of common paths for a single scenario were 4 and 53, respectively. The 15 SLG fault scenarios had 421 common paths spread among them. The remaining ten scenarios had 56 common paths. The large number of common paths for the SLG faults is due to the large variation in relay trip times as fault location and system load varies.

Common paths can be mapped into 2-D coordinates with the *y*-axis indicating the state identification code (state ID) and the *x*-axis indicating normalized TSs. An edge between two vertices represents the temporal transition between two states. Each vertex is marked with state information. Note that, only necessary features are displayed to save space.

Fig 6.3.4 shows common paths for two scenarios, a fault in the 36%–40% fault location of line L1 and a fault replay attack on line L1. The fault and fault replay paths both start at the system normal state. For real faults, the PMU will measure high current when a fault is present while for the fault replay attack, the attacker injects high current measurements to the PDC. This makes the second state of both common paths high current detected at relay R1, i.e., *IR*1 = high. However, these paths differ. immediately because for the fault replay, the attacker has to inject relay trip commands to relay R1 and R2 at the same time. As such, the second state for the fault replay attack has the trip commands to R1 and R2 detected by Snort,

i.e., SNT = (R1, R2) in Fig. 4

.

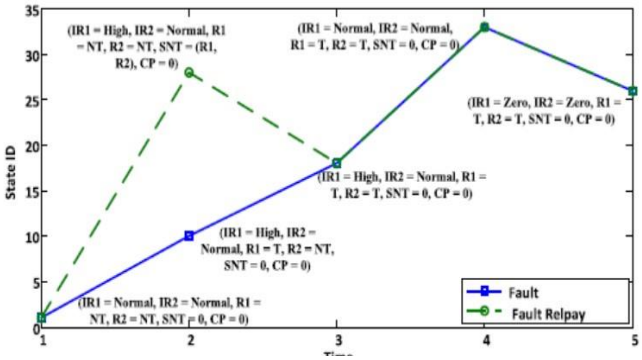

Fig 6.4.2: *2-D coordinates documenting fault versus fault replay attack common paths*.

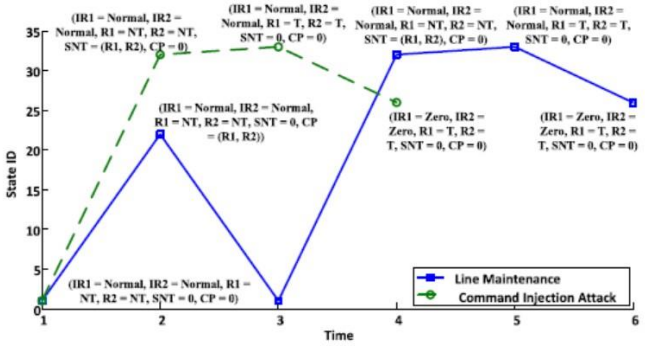

Fig 6.4.3: *2-D coordinates documenting line maintenance versus command injection attack common paths.*

Fig 6.3.5 shows common paths for line maintenance and command injection attack scenarios. The primary difference between the two scenarios is the command to open relays R1 and R2 originates from the control panel computer for the line maintenance scenario.

This causes the control panel log to include a trip command message. The common path for the line maintenance scenario includes a state noting the detection of control panel log events

[i.e., CP = (R1, R2)] and states showing Snort detecting remote trip command network packets

[i.e., SNT = (R1, R2)].

The common path for command injection includes the Snort alert but excludes the control panel log state. Figs. 6.3.4 and 6.3.5 demonstrate that common paths contain the critical states for different scenarios. The primary contribution of the common path mining algorithm is the ability to automatically create unique paths for each scenario type from data sets which measure behaviour associated with the scenarios.

6.5. EVALUATION

Three approaches were used to evaluate the IDS. First, the IDS was used to classify 5000 instances of scenarios from the test data set described in Section IV of this paper. Confusion matrices are provided to show IDS accuracy. A detailed review of the algorithms ability to classify SLG faults by fault location is also provided. Second, training and testing was repeated with sets of four scenarios missing from the data set. This test was used to demonstrate the IDSs ability to detect zero-day attacks and unknown scenarios.

Finally, IDS cost and performance was measured by measuring the amount of processing time and memory required during training and evaluation.

Tables VI and VII provide confusion matrices for the 25 tested scenarios. The confusion matrices were separated into two tables to allow them to fit in the column width of this paper. The row labeled "Oth" represents scenarios Q14–Q25 in Table VI and Q1–Q13 in Table VII. The row labeled "Unk" provides the number of instances which were unclassified due to no matching common path. Finally, the row labeled "Unc" provides the number of instances with uncertain classification due to matching more than one common path from more than one scenario.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Q1	505	0	31	0	0	0	0	0	0	0	0	0	0
Q2	0	502	0	34	0	0	0	0	0	0	0	0	0
Q3	10	0	301	0	0	0	0	0	0	0	0	0	0
Q4	0	6	0	321	0	1	0	0	0	0	0	31	0
Q5	0	0	0	0	130	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	108	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	67	0	0	0	0	0	0
Q8	0	0	0	0	0	0	0	54	0	0	0	0	0
Q9	0	0	0	0	0	0	0	0	99	0	0	0	0
Q10	0	0	0	0	0	0	0	0	0	57	0	0	0
Q11	0	0	6	0	1	0	0	0	0	0	127	0	0
Q12	0	0	0	0	0	0	0	0	0	0	0	104	0
Q13	0	0	0	0	0	0	0	0	0	0	0	0	179
Oth.	1	2	3	1	0	0	0	0	0	0	0	0	0
Unk	3	1	4	0	0	1	35	32	0	26	0	0	0
Unc	0	2	8	4	0	0	0	0	0	0	0	0	0

Table 6.5.1: *CONFUSION MATRIX FOR SCENARIOS Q1–Q13*

In total, 90.4% of the tested instances were correctly classified and 2.7% of the instances were misclassified. 4.7% of instances were classified as unknown and 2.2% were classified as uncertain. All of the cases of uncertain classification were related to SLG fault instances which matched a common path for more than one fault scenario. The IDS can generate false positives, especially, in the case of scenarios which are designed to mimic a nonattack scenario or event. For this paper, false positives rates were calculated for all nonattack scenarios misclassified as attacks. Scenarios Q1 and Q2, both SLG faults, had 2.1% and 1.6% false positive rates, respectively.

	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25
Q14	220	0	0	0	0	0	0	0	0	0	0	0
Q15	0	208	0	0	0	0	0	0	0	0	0	0
Q16	0	0	162	0	0	0	0	0	0	0	0	0
Q17	0	0	0	40	0	0	0	0	0	0	0	0
Q18	0	0	0	0	73	0	0	0	0	0	0	0
Q19	0	0	0	0	0	33	0	0	0	0	0	0
Q20	0	0	0	0	0	0	45	0	0	0	0	0
Q21	0	0	0	0	0	0	0	424	0	0	0	0
Q22	0	0	1	0	0	0	0	0	413	0	0	0
Q23	0	0	0	0	0	0	0	0	0	122	0	0
Q24	0	0	0	0	0	0	0	0	0	0	112	0
Q25	0	0	0	0	0	0	0	0	0	0	0	114
Oth	0	0	0	2	1	5	0	0	0	0	0	0
Unk	10	12	2	23	0	12	0	6	33	16	19	0
Unc	37	58	0	0	0	0	0	0	0	0	0	0

Table 6.5.2: *CONFUSION MATRIX FOR SCENARIOS Q14–Q25*

In both cases, the majority of false positives were classified as fault replay attacks. Replay attacks are designed to mimic SLG attacks. One out of eleven false positives were classified as a relay disable attack. Scenarios Q5 and Q6, both line maintenance events, had 0.8% and 0.9% false positive rates, respectively, which was one false positive for Q5 and Q6, respectively. For the Q5 scenario, the false positive was a command injection attack to open both relays at the end of the transmission line. For the Q6 scenario, the false positive was a fault replay attack. In both cases, the sequence of states in the common paths for the actual scenario and the misclassified scenario have overlapping sub-sequences of states. This overlap combined with variability in observed data due to power system and measurement system dynamics can lead to false positives. Additional evaluation was performed for classifications of the sub-groups of scenarios Q1, a SLG fault on line L1. The paths for Q1 were grouped into sub-groups by fault location and circuit load as previously mentioned. The SLG fault with grouping accuracy rate was 84.6% while 11.35% of the paths were misclassified.

Further analysis showed that a majority of misclassification occurred when SLG fault groups were classified as members of a neighbouring or nearby fault group. The grouping experiment demonstrates the common path mining algorithm's strength of finding unique paths for even similar scenarios. Tenfold cross-validation was used to evaluate the detection
accuracy of zero-day attack scenarios as shown in Table V. For each round of testing four scenarios were randomly selected to be excluded from training but present in the testing data set. The average detection accuracy for zero-day attack scenarios was 73.43%. However, there were cases where the detection rate for zero-day attack was low. For example, analysis of round three results showed that scenario Q6 (command injection to trip relays R1 and R2) was always misclassified as scenario Q3 (fault replay attack on line L1). This occurs because the expected common paths for Q6 and Q3 are similar. Therefore, when Q6 is unavailable in training, instances of Q6 are classified as instances of Q3 which leads to misclassification. In this case, both Q6 and Q3 are attacks and the zero-day attack is classified as another attack which is better than classifying as a nonattack. To improve the classification accuracy between similar scenarios additional sensors are needed to illuminate events which are different between the two scenarios. Of course, in the zero-day case it is difficult to predict which additional sensors may be required. Training and classification processing time and memory usage were measured using an Ubuntu Linux Virtual Machine with 3.5 GHZ CPU and 2 GB memory. Training required 0.33 s per scenario instance and 34 MB memory. Classification of test cases required 0.85 s per scenario instance to complete and 26.2 MB of memory. Multiple batch processing-based data mining algorithms were used to classify power system faults and cyber-attacks in using the same data used for the work presented
in this paper. The results in [22] were for classification with binary classes (attack and nonattack), three classes (attacks, non-attacks, and normal), and multiclass (all classes maintained).

The common paths mining-based IDS outperformed all traditional methods in [22] for overall accuracy in the multiclass case. A combination of the JRipper and Adaboost algorithms produced accuracy approaching 90% which is similar to the accuracy of the common paths mining-based IDS. All other test approaches had significantly lower accuracy than the IDS presented in this paper. The binary and three-class methods in [22] lead to improved accuracy at the expense of classification precision. The common paths mining-based IDS provides accurate and precise classification of each scenario type. Precise classification by scenario type is needed to speed understanding of attacks and to enable automated or manual response. Binary and three-class IDS

need post processing to provide additional detail before response. The primary advantage of common paths mining-based IDS over a traditional batch processing IDS is the ability to process data as a stream rather than collecting batches of data for off line analysis.Stream processing minimizes the amount of memory required to train and classify and therefore is better suited for IDS at the scale of a power system.

The common paths mining-based IDS provides stateful monitoring of an electric transmission distance protection system by leveraging a fusion of synchrophasor data and information from relay, network security logs, and EMS logs. The IDS is trained using a common path mining algorithm. Common paths are hybrid signatures and specifications which described patterns of system behavior associated withpower system events. The algorithm provides a time-domain data analysis approach to overcome transients present in the measurements. This is done by mining shared states out of a group of observed paths. Common paths are used to describe system responses to power system disturbances, control actions, and cyber-attacks.

The IDS matches monitored system state traversal to common paths to make classification decisions. Classification is specific to each trained scenario rather than simply an indication of normal or abnormal activity. In this paper, the IDS was trained an evaluated for a three-bus two-line transmission system which implements a two-zone distance protection scheme. Twenty-five scenarios consisting of stock ticker SLG faults, control actions, and cyber-attacks were implemented on a hardware-in-the-loop test bed. Scenarios were run in a loop 10 000 times with randomized system parameters to create a dataset for IDS training and evaluation. The IDS correctly classified 90.4% of tested scenario instances. Evaluation also included a tenfold cross validation to evaluate the detection accuracy of zero-day attack scenarios. The average detection accuracy for zero-day attack scenarios was 73.43%. The common paths mining-based IDS outperforms traditional machine learning algorithms and is better suited for the high volume of data present in power systems.

Currently, the common paths mining-based IDS builds common paths from captured data logs. Capturing such data logs for real systems is difficult. As such, future work is required to limit the amount the number of captured scenarios instances required to train the algorithm. The IDS was tested by offline review of test data sets. Future work is needed to update the IDS to perform real time classification from live system inputs and to incorporate the classifier with an intelligent adaptive control framework to achieve increased automation in of power systems.

Round	Excluded Scenarios	Z.D. Acc. (%)
1	Q3, Q11, Q18, Q22	76.3
2	Q2, Q8, Q12, Q23	67.3
3	Q6, Q11, Q16, Q17	50.5
4	Q1, Q5, Q8, Q10	73.3
5	Q1, Q9, Q19, Q21	91.8
6	Q5, Q13, Q20, Q23	64.7
7	Q5, Q10, Q15, Q16	63.8
8	Q12, Q13, Q19, Q24	70.7
9	Q2, Q7, Q9, Q17	76.3
10	Q9, Q10, Q16, Q19	99.8

Table 6.5.3: *DETECTION ACCURACY FOR FOUR RANDOM ZERO-DAY*

Methodology -II

Implementation of CNN in IDS of Power System

Let's take a look at how we implement Convolution Neural Network (CNN) in our project's main field and all the processes that helps to find out the result of this project.

Step 1: At first, we take all the necessary data, like (Amplitude, Magnitude, Phase angle of all the Three-phases (r,y,b) and also the status flag for each and every Relays or (IED's). Also take some another parameter. Then we make a data sheet with the help of these parameters and converted to csv (comma separated value) file. As CNN operates the datasheet in csv format.

Step 2: Now, we import this datasheet (csv file) in our CNN system, with the help of '*pandas*' *library*

Step 3: Then we extract the parameters only for **Relay-1** from our main datasheet which we import to our system and proceeds for further steps.
Where size of the entire datasheet (15582 rows x 6 columns)

Here, we take the parameters of Relay-1 (Voltage and Current) for all the three-phases and status flag.

Step 4: As we want to make an Image with the help of Relay-1 data where only the Voltage and currents values are enough to implement our desired Image. So, now we drop status flag from that. And make that Image with the help of those parameters by CNN. Which is our input parameters.

CNN takes those parameters and make this Image which is a 3-Dimentional array like (3, 2, 2)

Step 5: Now, we apply the *label Encoder* to transform and fit the Relay-1 status flag as an output. By which we can identify that all the parameters of Relay-1 are in natural or faulty conditions.

Here,

Fault_Data_TrainX.shape
Fault_Data_TrainY.shape

```
(15582, 3, 2, 2)
(15582, 6)
```

Step 6: Then, from our extract datasheet (15582 rows x 6 columns), we divide [8:2] for training purpose and rest we left for testing purpose.

So, here

```
x_train shape (14023, 3, 2, 2)
x_test shape (1559, 3, 2, 2)
```

```
y_train shape (14023, 6)
y_test shape (1559, 6)
```

Step 7: Now, with the help of CNN we extract the features of that Image and apply ReLU functio n the newly convolution layer's then by max polling we got again another layers, that's called pol ling layers then dropout the features and set for flatten for the operation parameters as an input of neural network.

Here,

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_18 (Conv2D)          (None, 3, 2, 8)           408

 max_pooling2d_18 (MaxPoolin  (None, 1, 1, 8)          0
 g2D)

 dropout_18 (Dropout)        (None, 1, 1, 8)           0

 conv2d_19 (Conv2D)          (None, 1, 1, 8)           584

 max_pooling2d_19 (MaxPoolin  (None, 1, 1, 8)          0
 g2D)

 dropout_19 (Dropout)        (None, 1, 1, 8)           0

 flatten_9 (Flatten)         (None, 8)                 0

=================================================================
Total params: 992
Trainable params: 992
Non-trainable params: 0
_________________________________________________________________
```

len(X_train):14023
X_train.shape :(14023, 3, 2, 2)
len(Y_val: 1559
Y_val.shape : (1559, 6)

Step 8: Now, this flattens parameters makes a input as a vector format and proceed by Neural Ne twork and we get the output. Now we match this output with our desired input image. If the outp ut is not same then we apply back-propagation method which flows output to input into the neura l network and changed the weights of required neurons until output image will be our desired Image.

Result:

Here, there may be some variation loss happened due to testing and training purpose so we need to apply required epochs and batch size so the validation loss will be minimum.

In this case, here the comparison of output image with desired image shown below and the result ant confusion matrix. By which we identify the output image or our Relay-1 data is in Natural or faulty condition, if in fault condition then how much fault happened.

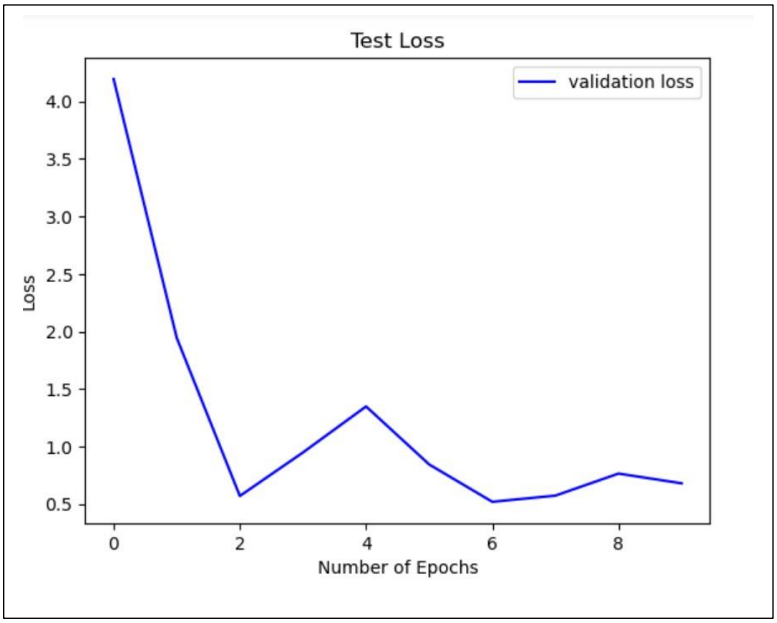

Fig 6.5: *Validation loss of the output image*

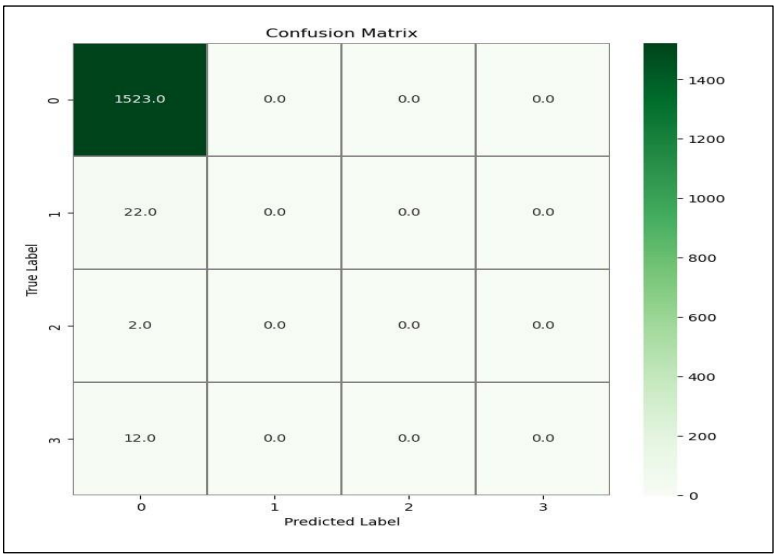

Fig 6.6: *Confusion matrix of the output image*

7. Proposed Solution

To judge the viability of using machine learning for intrusion detection on smart grid electrical systems we tested various popular learners using Weka as the machine learning framework and open-source simulated power system data provided by Mississippi State University. The classification of events was performed using three different classification schemes:

• Multiclass - Each of the 37 event scenarios, which included attack events, natural events, and normal operations, was its own class and was predicted independently by the learners,

• Three-class – The 37 event scenarios were grouped into 3 classes: attack events (28 events), natural event (8 events) or "No events" (1 event).

• Binary – The 37 event scenarios were grouped as either an attack (28 events) or normal operations (9 events).

The data was drawn from 15 data sets which included thousands of individual samples of measurements throughout the power system for each event type. The datasets were randomly sampled at 1% to reduce the size and evaluate the effectiveness of small sample sizes. For this analysis, there was an average of 294 "No event" instances, 3,711 attack instances and 1,221 natural events instances used across the classification schemes. The date and time information were removed since scenarios were run sequentially and time and date would perfectly classify the data.

For each of the three schemes, Multiclass, Three-class and Binary, we tested 7 learners on 15 datasets. When running the experiments we chose to use the tenfold or 10x cross validation methodology. When testing using this method we partitioned the dataset into 10 sets randomly selecting instances from each category. The model was built on a ninety percent selection from the data and tested on the remaining ten percent of the data to evaluate the learner's performance. We repeated this for each learner and each dataset then taking the average over the fifteen datasets to summarize the results.

7.1. The classification algorithms we tested were:

OneR – This is a learner with a very simplistic method that evaluates each feature's optimum rule and chooses the best one from all feature rule sets.

NNge – a nearest-neighbor-like algorithm that classifies examples by comparing to those already seen and comparing the new examples to its surrounding data points.

Random Forests – this is an ensemble of tree predictors where each tree casts a vote for the most popular class on input of a new instance [23]. The collection of decision trees are created from randomly pulled training data samples.

Naïve Bayes - is a probabilistic classifier based on the Bayes' theorem [25] that reflects the conditional probability distribution of a set of random variables, and was adopted into the field of machine learning in 1992 [26].

SVM – Support vector machines [28] trained using sequential minimal optimization [29]. An SVM model is a representation of the examples as points in a space, with classes divided by a mathematically determined set of hyperplanes that maximize the margin between the classes. New examples are then predicted to belong to a class based on their position in that space relative to the hyperplanes.

JRipper – Incremental Reduced Error Pruning algorithm that uses a separate-and-conquer methodology developed in [30] and modified by Cohen as shown in [31] to generate a sophisticated rule set.

Adaboost – short for Adaptive boosting, this is an algorithm use to improve the performance of other types of learning algorithms [35]. It is an ensemble learning method where each new model instance focuses on training examples that were misclassified in the previous models. By combining Adaboost with our strongest performer we achieve much better results. AdaBoost M1 method used in Weka can be used in conjunction with learners to improve their performance.

7.2. The classifiers we used can be grouped under these categories:

- Probabilistic classification (Naïve Bayes)
- Rule induction (OneR, NNge, JRipper)
- Decision tree learning (Random Forests)
- Non-probabilistic binary classification (SVM)
- Boosting, a meta-algorithm for learning (Adaboost)

8. Future Plan

Based on the experience made during working for this publication, we found out two research directions for IDSs safeguarding IoT networks that, in our opinion, seem worthwhile to be pursued. They are introduced in the following.

8.1. Intrusion Detection as a Service in Fog Computing

Table 2 gives the impression that one has more possibilities to apply approaches existing for WSNs, MANETs, and CPSs also for an IoT network if it contains at least some nodes with sufficient processing and energy capabilities. That holds particularly when these high-performance nodes are plugged such that energy issues are alleviated. These devices can then execute the computing intensive centralized IDS approaches while the resource limited nodes only assist by delivering data. This fits well to the novel Fog Computing concept; see, e.g., Bonomi et al. [97]. Fog Computing is seen as an alternative to traditional Cloud Computing in which the various cloud services are not provided by remote data centers but by local machines that are under the control of the local network operator. For instance, local WLAN routers that are provided with greater processing power and storage facilities can, besides routing data packets between the wired and the wireless network segments, offer various services known from the cloud.

Since border routers connecting an IoT system with the outside world are often WLAN routers, the new Fog Computing technology can easily be integrated into the network. For instance, it could run a centralized IDS protecting the IoT network nodes to which it is connected or take processing- and energy-intensive tasks of the implementation strategies discussed in this paper. Moreover, if the IoT is larger and applies several border routers, one can use their Fog Computing capabilities to realize a hierarchical IDS. In consequence, we see the integration of IDSs on Fog Computing platforms as a promising future research direction. Following the highly virtual nature of the platforms, the IDS functionality can then, like other cloud-based functionality, be offered in form of services, which could be named intrusion detection as a Service.

8.2. Reducing Active Channel Listening Times When Rating Network Behavior

To realize an IDS is more difficult for IoT systems when all nodes are resource-constrained, Table 2 reveals for this case that there are three basic strategies available. One is voting-based IDSs that are already sufficiently lightweight to be used in a resource-friendly way. Unfortunately, their accuracy is still suboptimal and further research is needed to reduce the rate of false negatives.

The second strategy is to reduce the workload by splitting it into subtasks executed by different cooperating nodes. That is done by hierarchical IDSs as well as the Distributed and Collaborative IDSs. The problem here is that the reduction of computation efforts takes place at the expense of more data exchange which leads to a faster battery draining. To avoid that, one should investigate the research and development of IDSs that allow the nodes to cooperate with each other minimizing the amount of data to exchange. Here, recent developments in communication protocol technology will be of help. An example is the new IEEE 802.15.4 protocol (see Bhar [98]) that

reduces active channel listening. For that, the data frames are divided into a number of slots, and a station has to only listen at time intervals when slots dedicated to itself are transmitted. For larger systems, that reduces the idle listening time of a station significantly.

The third strategy is to use reputation and trust management that provides IDSs with lightweight computation and storage mechanisms. The approaches using trust management, however, are subject to increased active channel listening since a node now also needs to listen to the communication towards its neighbors, the behavior of which shall be evaluated. If our node has to listen continuously, this can consume a lot of energy. Therefore, it might be helpful to conduct research in the combination of the approaches with resource-friendly communication protocols. For instance, a first analysis to adapt the approach presented in Khan and Herrmann [43] and Khan et al. [85] to the IEEE 802.15.4 protocol revealed that the active channel listening time can be easily reduced by two-thirds when the listening strategy is slightly changed. When our station wants to check if a message sent by itself to another station is correctly forwarded to rule a selective forwarding attack out, it only needs to listen to the slots to itself and the one through which the other node forwards the message of interest. Thus, the additional listening cost can be effectively limited. Altogether, the dedication of research in combining energy-efficient networking with reputation-based IDSs seems a promising field of research.

CONCLUSION

The conclusion of the "Intrusion Attack Detection in Power System Network" project would depend on the specific findings and results obtained during the research. However, based on the general context of intrusion attack detection in power system networks, a typical conclusion may include the following key points:

Importance of Intrusion Detection: The project emphasizes the significance of intrusion detection in power system networks. With the increasing interconnectedness and digitization of power systems, the vulnerability to cyber-attacks has grown substantially. Therefore, having effective intrusion detection mechanisms is crucial for ensuring the security and reliability of the power grid.

Research Objectives: The project aimed to investigate and develop effective intrusion detection techniques tailored to the unique characteristics and requirements of power system networks. These techniques may include anomaly detection, signature-based detection, machine learning algorithms, or a combination of these approaches.

Methodology and Experimentation: The project involved designing and implementing a suitable experimental setup to evaluate the performance of the proposed intrusion detection techniques. The dataset used for evaluation may have included real-world power system data or synthetic data that accurately represents the behavior of power systems.

Evaluation of Intrusion Detection Techniques: The project evaluated the effectiveness and efficiency of the developed intrusion detection techniques. Metrics such as detection accuracy, false positive rate, false negative rate, detection time, and computational resources required were considered during the evaluation process.

Results and Findings: The project may have yielded positive results, indicating that the developed intrusion detection techniques are capable of effectively detecting and mitigating various types of intrusion attacks in power system networks. The results might demonstrate improved detection accuracy, reduced false alarms, and faster response times compared to existing approaches.

Practical Implications: The project emphasizes the practical implications of the developed intrusion detection techniques. It highlights their potential for deployment in real-world power system networks to enhance security and protect against cyber-attacks. Additionally, the project may discuss the challenges and considerations for integrating these techniques into existing power system infrastructure.

Future Research Directions: The project concludes by identifying potential areas for future research and improvement. This may include exploring advanced machine learning algorithms, considering the impact of emerging technologies (such as the Internet of Things), investigating the

resilience of intrusion detection systems against sophisticated attacks, or extending the research to address other aspects of power system security.

Overall, the conclusion should summarize the project's contributions, highlight the significance of the research findings, and provide recommendations for further development and application of intrusion detection techniques in power system networks.

References

[1] N. Falliere, L. O'Murchu and E. Chien, "W32.Stuxnet Dossier", Online: http://goo.gl/kzVOSC, Nov. 2010.

[2] D. E. Bakken, A. Bose, C. H. Hauser, E. O. Schweitzer III, D. E. Whitehead, and G. C. Zweigle, "Smart Generation and Transmission with Coherent, Real-Time Data," Technical Report TR-GS-015. August, 2010.

[3] R. Moxley and D. Dolezilek, "Case studies: Synchophasors for widearea monitoring, protection, and control," Proc. 2nd IEEE PES International Conf. and Exhibition on Innovative Smart Grid Technologies (ISGT Europe), pp.1-7, 5-7, Dec. 2011.

[4] S. Horowitz, D. Novosel, V. Madani, and M. Adamiak, "System-Wide Protection", IEEE Power & Energy Magazine, vol. 6, no. 6, pp. 34 – 42, Sep. 2008.

[5] SEL; "Mitigating the Aurora Vulnerability with Existing Technology." Online: http://goo.gl/9hkAJb, Oct. 2009

[6] M. Masera and I. Nai Fovino, "Effects of intentional threats to power substation control systems", Int. J. Critical Infrastructures, vol. 4, no. 1/2, pp.129–143, 2008.

[7] T. Morris, S. Pan, J. Lewis, J. Moorhead, B. Reaves, N. Younan, R. King, M. Freund, and V. Madani, "Cybersecurity Testing of Substation Phasor Measurement Units and Phasor Data Concentrators," (CSIIRW '11), pp. 12-14, Oct. 2011.

[8] Chee-Wooi Ten; Junho Hong; Chen-Ching Liu, "Anomaly Detection for Cybersecurity of the Substations," Smart Grid, IEEE Transactions on, vol.2, no.4, pp.865,873, Dec. 2011

[9] Y. Chen and B. Lou, "S2a: Secure smart household appliances," in Proc. 2nd ACM Conf. Data Application Security Privacy, San Antonio, TX, USA, pp. 217-228, Feb. 2012.

[10] Mitchell, R.; Ing-Ray Chen, "Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications," Smart Grid, IEEE Transactions, vol.4, no.3, pp.1254, 1263, Sept. 2013.