ENHANCING MODELLING OF CYBER-SECURITY FRAMEWORK FOR CRITICAL INDUSTRIAL INFRASTRUCTURE USING MACHINE LEARNING TECHNIQUE

***[1]Onyia Adimora U., [2]Eneh I.I.**
[1,2]Department of Electrical and Electronic Engineering (Esut), Enugu State, Nigeria
Email: [1]*saint.onyia007@gmail.com [2]innocent.ifeanyichukwu@esut.edu.ng
[1]Corresponding Author Email: saint.onyia007@gmail.com
Tel: +234 8123971668

Manuscript received 2/4/2024; Accepted 2/7/2024; Published 20/7/2024

Abstract

Over the years the critical industrial infrastructure has suffered many problems of which cyber-attack is among the major concerns. This attack exploits hidden vulnerabilities on the network and penetrates with threat for ransomware. To address this problem, this study presents enhancing the modelling of cyber security framework for critical industrial infrastructure using machine learning technique. The aim is to model a proactive defence mechanism capable of identifying attacks effectively in Industrial Internet of Things (IIoT). The research method used are vulnerability assessment and identifying potential exploits or attack, data collection of cyber threats dataset, data process through normalization and feature extraction, multi-layered neural network, back-propagation algorithm, Intelligent Cyber Threat Detection (ICDS), and cyber threat detection response system. The research design approach used mathematical and structural methods to model the ICDS and the integration of the response system on the IIoT network and then tested the model using MATLAB programming software. The ICDS was evaluated considering key parameters for cyber security success such as precision, recall, accuracy, sensitivity, specificity and latency. The results for precision reported 96.9%, recall was 96.7%, accuracy was 96.9%, sensitivity reported 98.2%, specificity 94.3% and latency: 70.82%. These results implied that the ICDS was able to detect threats penetration on the network and mitigate it with high success rate. To validate the ICDS model, comparative approach was applied considering the IIoT network without IDCS and the improved network with integrated ICDS. The result reported 75.65% improvement for loss, 21.80% improvement for throughput, bandwidth utilization factor reported 46.46% improvement and finally latency reported 70.82% improvement. In conclusion the study identified potential vulnerabilities in IIoT and made recommendations to patch them, while developing an ICDS for the detection of cyber threat and minimizing the impact on IIoT.

Keywords: Intelligent Cyber Threat Detection; Industrial Internet of Things;

1. INTRODUCTION

In the world today, “internet” has become part of our lives and has continuously influence recent decisions, plans, and implementation of ideas made by man in all walks of life. Most recently in the industrial sector, the process design and overall industrial automation process has been improved with the integration of internet, as a result distribution control system, health status of equipment, and real-time monitoring have become more efficient and effective (Alshahrani et al., 2023). However, this increased reliance on internet connectivity also exposes industrial infrastructure to cybersecurity risks and vulnerabilities.

Cyber threats in the industrial sector have become more sophisticated, posing significant challenges to the integrity, confidentiality, and availability of critical industrial systems.

According to Mbanaso et al. (2019), these critical industrial infrastructures are designed by engineers, and there are tendencies for error either in the design configuration, software, settings, or human error due to oversight. These loopholes in Industrial Control System (ICS) can be exploited by attackers for cyber threat type called zero-day attack and subject the industry to financial loss, potential disruption of technical process, ransomware and even threat to human lives due to the implication of hazards (Patidar and Khandelwal, 2018; Ndungu, 2021).

Over the years, many Intrusion Detection System (IDS) have been developed for cyber-attack detection in IIoT, considering cyber-attack model such as malware, ransomware, virus, denial of service attack, advanced persistent threats (APTs), as well as insider threats and social engineering technique (Mvula et al., 2023). In line to detect these threats methods such as signature-based approach, behavioural based approach, anomaly detection, network based, machine learning based, host based and hybrid approaches have all be submitted for cyber threat detection (Mohd et al., 2023). However, one approach which has dominated recent studies is the application of Machine Learning (ML) algorithms for the detection of cyber threats in IIoT.

ML is algorithms which can learn and make decision without explicit programming. It has the capacity to analyze vast amounts of data generated by industrial systems detecting patterns and anomalies that may indicate cyber-attacks or unauthorized activities (Peppes et al., 2023). In addition, it can also predict potential threat and give room for proactive measured to address these threats, through the data analysis of common attack patterns and provide valuable insights for strengthening the overall cybersecurity posture of industrial infrastructure (Mohd et al., 2013).

The automation of cyber threat detection and response process can also e facilitated using ML algorithms. This will reduce the burden on human operators and enable real time response to threat detection, by isolation from the gateway (Ibitoye et al., 2019). However, ML algorithms are not completely the holy grail of IDS, as there also have their limitations. For instance, they are limited by the training dataset, (i.e, they can only detect the attack model trained with), they can be deceived through adversarial attack, they can be bias in decision making among other technical challenges of ML algorithms.

Some of the literatures which used ML for IDS include Li et al. (2019) who used multi-Convolutional Neural Network (CNN) or the classification of adversarial attack and achieved 76.67% accuracy. Similarly, multiple neural network layers were trained for IDS and reported 99% accuracy. In Bae et al. (2019), autoencoder was used to develop IDS for smart factor environment and reported 95% accuracy. While these studies all reported good accuracy, the actual definition of success was not defined. This is because accuracy alone as a criterion for evaluation for evaluation of IDS model may not prove the trustworthiness and reliability to secure IIoT facility.To solve this problem, this research proposes the modelling of cyber-security framework for critical industrial infrastructure using machine learning technique. By integrating machine learning techniques into the cybersecurity framework, this research aims to develop a proactive defence mechanism capable of identifying attacks effectively in IIoT. This when achieved will provide an advanced and adaptive defence mechanism against cyber threats.

2. RESEARCH METHODOLOGY

This research methodology used is the experimental and simulation approach. The aims to develop a proactive defence mechanism capable of effectively identifying attacks in Industrial Internet of Things (IIoT) systems, with a specific focus on a palm fruit distribution control system. The study follows a comprehensive methodology to achieve its objectives. First, data collection is performed, specifically targeting the palm fruit distribution control system. This involves gathering relevant data such as network traffic logs, sensor readings, control system configurations, and historical attack data. It is crucial to ensure that the collected data represents realistic scenarios and includes potential attack patterns. The study then proceeds with vulnerability assessment tests on the IIoT system to identify weaknesses and potential attack vectors. Tools like OpenVAS and Wireshark are utilized to scan the system for known vulnerabilities. The core focus of the research lies in the development of a modelling framework that integrates machine learning techniques for intelligent cyber threat detection in IIoT systems. This framework aims to leverage the power of machine learning algorithms to detect attacks effectively. The specific challenges and characteristics of the palm fruit distribution control system are taken into account during the development process. To evaluate the effectiveness and performance of the proposed cyber-security framework, extensive simulations are conducted using MATLAB. The framework is compared with existing approaches to assess its superiority in detecting attacks in the palm fruit distribution control system. Finally, the developed model is validated, and recommendations and guidelines are provided for the adoption of the proposed cyber-security framework in real-world IIoT environments.

3. RESEARCH DESIGN

To develop the Intelligent Cyber Threat Detection System (ICDS), the methods used are data collection, data processing through data cleaning, data normalization, machine learning algorithms, classification, performance evaluation, generation of the ICDS model.

3.1 Data collection

The data used for the study is the UNSW-NB15 dataset which was generated by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) using the IXIA PerfectStorm tool. It combines real modern normal activities with synthetic contemporary attack behaviours, resulting in raw network packets. A total of 100 GB of raw traffic is captured using the Tcpdump tool, stored as Pcap files. This dataset encompasses nine attack families, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. To generate features with class labels, the Argus and Bro-IDS tools are employed, leading to the creation of 49 features. These features, along with their descriptions, are documented in the UNSW-NB15_features.csv file. The dataset contains a total of 2,540,044 records, divided across four

CSV files: UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv. The ground truth table is named UNSW-NB15_GT.CSV, and the list of events is stored in the UNSW-NB15_LIST_EVENTS file. For training and testing purposes, a partition of this dataset is allocated. The training set, named UNSW_NB15_training-set.csv, consists of 175,341 records, while the testing set, named UNSW_NB15_testing-set.csv, contains 82,332 records. Both sets encompass various types of attacks as well as normal activities.

3.2 Data processing

Due to the huge volume of data collected, the challenges of missing data and unstructured data cannot be ignored. To address these issues and ensure data integrity, data cleaning process using missing value replacement techniques was applied. In the same vein normalization process was also applied and used to prepare the data training. The next step adopted multi-layered neural network and train with the processed data to generate the desired ICDS model.

3.3 Data cleaning

Data cleaning is a crucial step in preparing data for analysis, and the method adopted is the imputation techniques to handle missing values using K-Nearest Neighbour (KNN) algorithm. In this process, the K-Nearest Neighbour (KNN) algorithm is utilized to predict and replace missing values. The KNN imputation technique identifies the K most similar observations to the one with the missing value based on other variables present in the dataset. The similarity is typically determined using distance metrics such as Euclidean distance or cosine similarity. Once the K most similar observations are identified, their corresponding values for the missing variable are used to impute the missing value.

3.4 Data normalization

Data normalization is a pre-processing technique used to transform data into a standard range or distribution. It helps to eliminate the influence of different scales and units in the data, making it easier for machine learning algorithms to process and compare variables. The methods used for the process is Min-Max Scaling (Normalization): This technique rescales the data to a specific range, typically between 0 and 1. It is achieved by subtracting the minimum value of the variable and dividing by the range (maximum value minus minimum value).

4. MACHINE LEARNING ALGORITHM

The study employed a multi-layered neural network as the chosen machine learning algorithm. This type of neural network consists of multiple layers that enable it to learn intricate representations and extract complex patterns from the data. By leveraging its depth and interconnected layers, the multi-layered neural network has the capability to capture and model intricate relationships, making it a powerful tool for addressing the complexities of the problem at hand. Through the iterative process of forward and backward propagation, the network can optimize its parameters to enhance its predictive capabilities. The multi-layered neural network used in this study serves as a robust and flexible framework for achieving accurate predictions and generating valuable insights from the given data.

4.1 The Algorithm of the MLP

The mathematical model of a multi-layered neural network involves defining the operations performed at each layer, as well as the mathematical equations that govern the forward and backward propagation processes. Here's a high-level overview of the mathematical model for a feed-forward multi-layered neural network:

1. Initialization: The network parameters, encompassing weights (W) and biases (b), are initialized for each layer to prepare for the subsequent training process where these parameters will be learned.
2. Forward Propagation:

- Iterating over each layer (l) from 1 to L (total number of layers):
- The input to layer l, denoted as z[l], is computed by applying a linear transformation to the output of the previous layer: z[l] = W[l] * a[l-1] + b[l].
- An activation function, represented as g[l], is then applied element-wise to the input, resulting in the output of layer l: a[l] = gl.

3. Cost Function: A cost function, such as cross-entropy loss, is defined to quantify the dissimilarity between the predicted outputs and the true labels.
4. Backward Propagation:

- The gradients of the cost function with respect to the output of the last layer (da[L]) are computed using the selected loss function.
- For each layer (l) from L-1 to 1:
- The gradients of the cost function with respect to the input (dz[l]) are computed using the chain rule: dz[l] = dJ/da[l] * g'[l].
- The gradients of the cost function with respect to the weights (dW[l]) and biases (db[l]) of layer l are calculated based on the input and the computed gradients of the subsequent layer: dW[l] = (1/m) * dz[l] * a[l-1].T and db[l] = (1/m) * np.sum(dz[l], axis=1, keepdims=True).
- The gradients of the cost function with respect to the output of the previous layer (da[l-1]) are obtained by multiplying the weights with the computed gradients of the current layer: da[l-1] = W[l].T * dz[l].

5. Update Parameters:

- The weights and biases of each layer are updated using an optimization algorithm, such as gradient descent or its variants. This involves taking a step in the opposite direction of the computed gradients: W[l] = W[l] - learning_rate * dW[l] and b[l] = b[l] - learning_rate * db[l].

6. The steps of forward propagation, backward propagation, and parameter updates (steps 2 to 5) are iteratively repeated until convergence or a predefined number of iterations.

This mathematical model provides a foundational framework for a multi-layered neural network, which was feed with the dataset collected and trained as follows.

4.2 Training of the Multi-Layered Neural Network

Algorithm: Multi-Layered Neural Network Training
Input:

- Training dataset: X (input features), Y (true labels)

- Network architecture: Number of layers (L), number of neurons per layer
- Learning rate: α
- Number of iterations: num_iterations

Output: Trained network parameters: W (weights), b (biases)

1. Initialize the network parameters:
   - Randomly initialize weights (W) for each layer
   - Initialize biases (b) for each layer
2. Repeat for num_iterations: a. Perform Forward Propagation: - Compute input to each layer (z[l]) using the previous layer's output: z[l] = W[l] * a[l-1] + b[l] - Apply activation function (g[l]) element-wise to compute the layer's output: a[l] = gl

b. Compute Cost Function: - Calculate the cost function (J) to measure the discrepancy between predicted outputs and true labels, using Y and the output of the final layer (a[L])

c. Perform Backward Propagation: - Compute the gradient of the cost function with respect to the output of the last layer (da[L]) - For each layer l from L-1 to 1: - Compute the gradients of the cost function with respect to the input (dz[l]) using the chain rule: dz[l] = dJ/da[l] * g'l - Compute gradients of weights (dW[l]) and biases (db[l]) for the current layer using dz[l] and the previous layer's output: dW[l] = (1/m) * dz[l] * a[l-1].T db[l] = (1/m) * sum(dz[l]) - Compute gradients of the output of the previous layer (da[l-1]) using the weights and dz[l]: da[l-1] = W[l].T * dz[l]

d. Update Parameters: - Update the weights and biases for each layer using the computed gradients: W[l] = W[l] - α * dW[l] b[l] = b[l] - α * db[l]

3. Return the trained network parameters: W, b

This algorithm outlines the steps involved in training a multi-layered neural network, including initialization, forward propagation, cost computation, backward propagation, and parameter updates. It iteratively adjusts the network parameters to minimize the cost function and improve the model's predictions. The resulting trained parameters were used to detect cyber-attack on the IIoT Network.

4.3 Development of the ICDS framework

The ICDS developed with the MLP based threat detection model operated by loading the attack dataset and initializing the MLP model. Then, for each set of input features, the algorithm performs forward propagation to obtain the predicted output. If the output indicates an attack, the result is marked as a potential threat, and control measures are activated. If the output does not indicate an attack, the algorithm returns to feature identification to analyze further. The process continues until all input features have been evaluated. The algorithm is presented as;

ICDS Algorithm

1. Load Attack Dataset
2. Initialize MLP Model for attack detection
3. Forward the data features of data to the trained MLP to predict output
4. For
5. Threat feature classification
6. True
7. Flagg off as threat
8. Return to step 3

5. SYSTEM IMPLEMENTATION

The implementation of the ICDCS using MATLAB involves utilizing the set of tools and functionalities of the ICDS model developed, the control model for threat developed and then integrated into the IIoT network using programming. To achieve these goals toolboxes for each model were selected and configured to in with programming to implement the model. The data acquisition toolbox was used to import the IIoT data for cyber threats and then train MLP algorithm with neural network toolbox to generate the ICDS model for the detection of threat. These extensive libraries of MATLAB facilitate the development and implementation of algorithms for the protection of IIoT network against attacks.

6. SYSTEM RESULTS

To evaluate the performance of the neural network model trained for the detection of cyber threat on the IIoT facility, various parameters which clearly defined the success of the model such as receiver operator characteristics curve, confusion matrix, cross entropy, recall, precision, accuracy and F1score were all utilized to evaluate the model. The figure 1 presented the ROC curve of the result. The graph was used to measure the area under the curve for the training, test and validation set; then the overall average of the three multi sets was used to determine the overall area under curve. While analysis the ROC, the aim is tailored towards achieving a value of approximately 1. This implies that the model was able to correctly detect denial of service attack when tested.

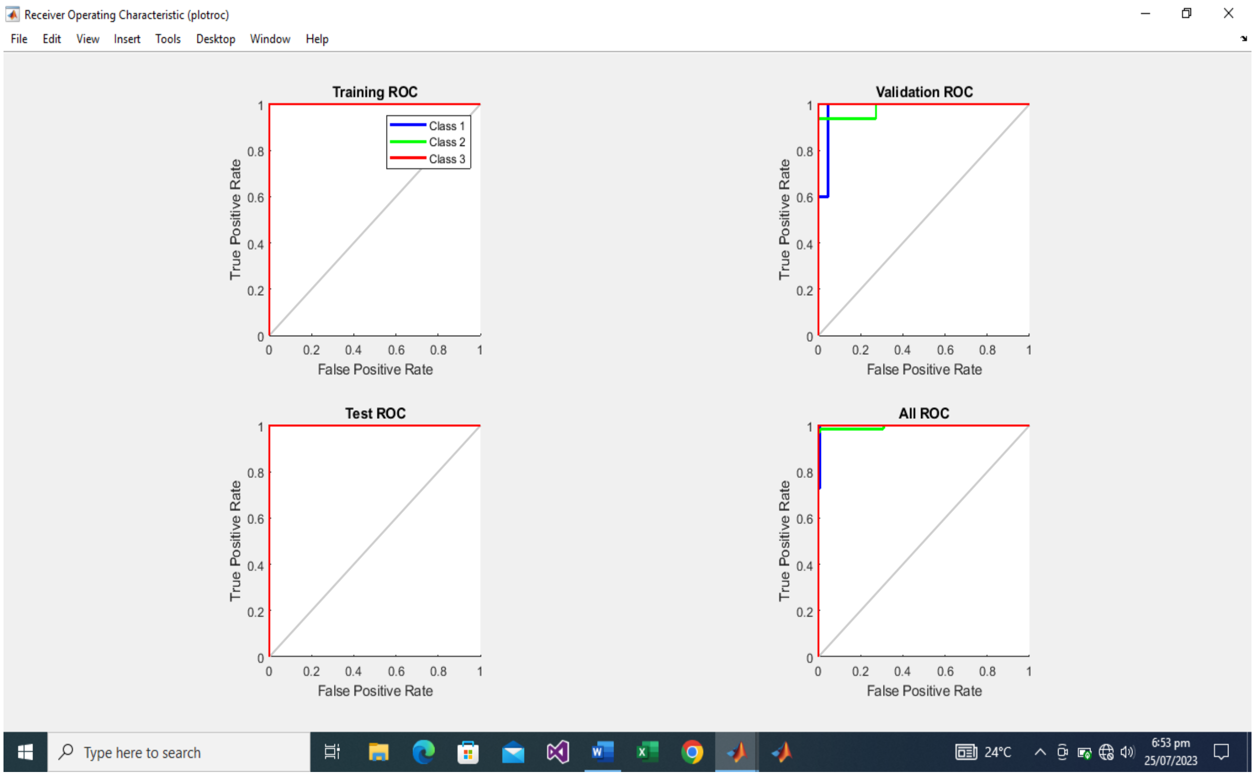


Figure 1: The ROC curve of the ICDS model

From the result of the figure 1 the ROC performance of the ICDS model was presented, showing the three classes which are the training, test and validation sets of the model, using the relationship between the true positive rate (i.e correct threat classification) and false positive rat (i.e correct not threat classification), to determine the ROC which overall is 0.9873. What this mean is that the IDCS was able to correctly classify denial of service attack when tested and also was able to correctly classify normal packet of the process design equipment which are used for the IIoT monitoring. The next result presented the cross-entropy performance which is another parameter to evaluate the training performance of the MLP when generating the ICDS model. The aim in this model is to achieve cross entropy value which is the error between the actual threat and the predicted threat is approximately zero. The figure 2 presented the result.

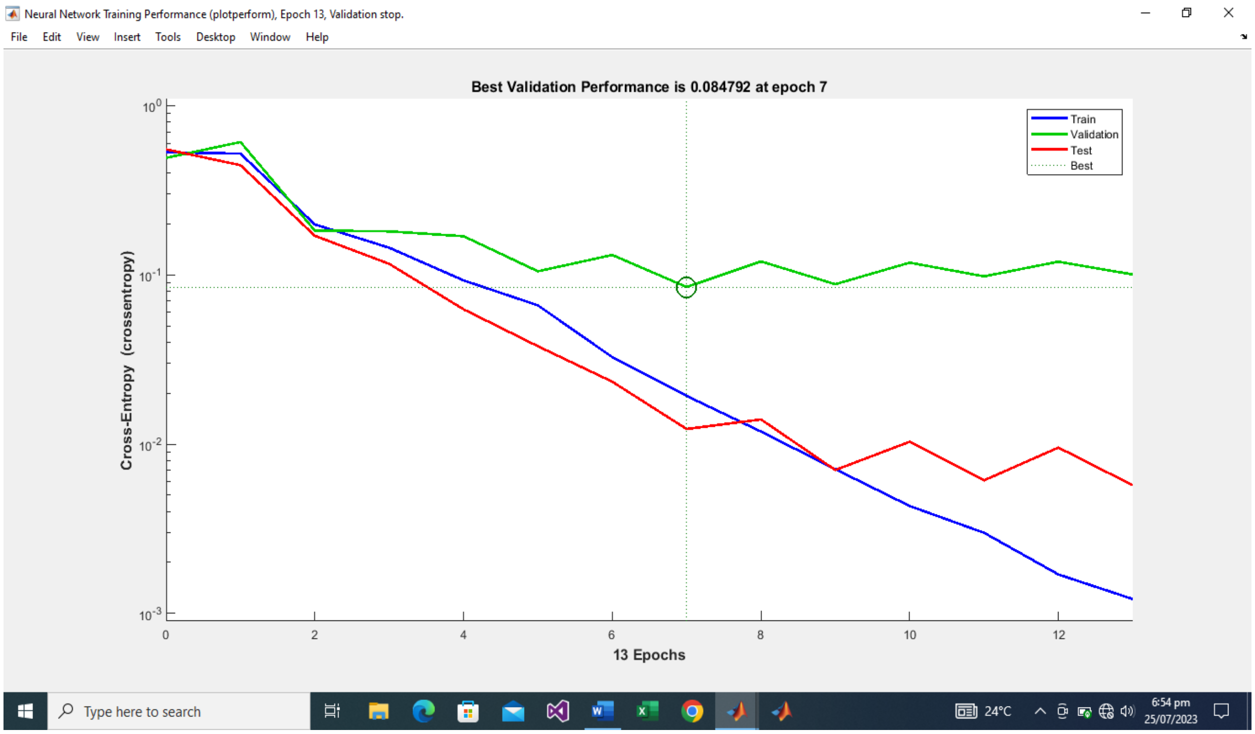
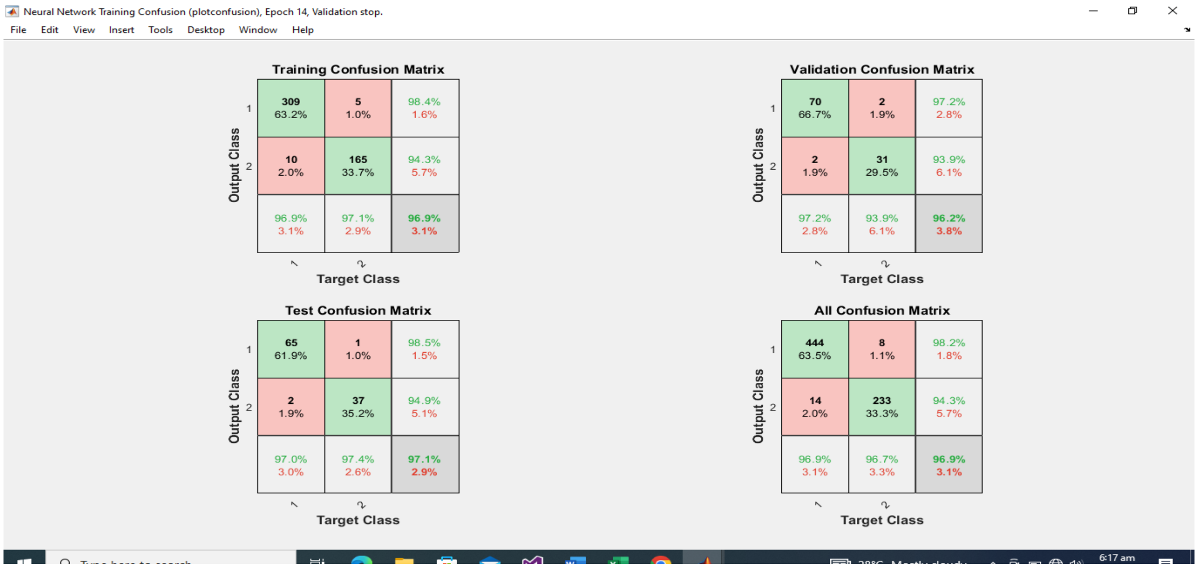

Neural Network Training Performance (plotperform), Epoch 13, Validation stop.
File
Edit
View
Insert
Tools
Desktop
Window
Help
Best Validation Performance is 0.084792 at epoch 7
Train
Validation
Test
Best
10^0
10^{-1}
10^{-2}
10^{-3}
Cross-Entropy (crossentropy)
0
2
4
6
8
10
12
13 Epochs
Type here to search
24°C
6:54 pm
25/07/2023

Figure 2: Result of the cross entropy

From the figure 2, the performance of the MLP training to generate the ICDS, considering the error which occurred during the training process was presented. From the result it was observed that the best error which occurred in the training, test and validation set is 0.084792 at epoch 7. What this mean is that at this point, the training process, testing and validation classes of the IDCS model achieved the lowest error and then the training stops; thus, implying good training performance, since the error is approximately zero. The confusion matrix which was sued to measure the precision, recall, accuracy, sensitivity, specificity, was presented in the figure 3.


Neural Network Training Confusion (plotconfusion), Epoch 14, Validation stop.
File Edit View Insert Tools Desktop Window Help
6:17 am

Training Confusion Matrix

Output Class \ Target Class	1	2	
1	309 63.2%	5 1.0%	98.4% 1.6%
2	10 2.0%	165 33.7%	94.3% 5.7%
	96.9% 3.1%	97.1% 2.9%	96.9% 3.1%

Validation Confusion Matrix

Output Class \ Target Class	1	2	
1	70 66.7%	2 1.9%	97.2% 2.8%
2	2 1.9%	31 29.5%	93.9% 6.1%
	97.2% 2.8%	93.9% 6.1%	96.2% 3.8%

Test Confusion Matrix

Output Class \ Target Class	1	2	
1	65 61.9%	1 1.0%	98.5% 1.5%
2	2 1.9%	37 35.2%	94.9% 5.1%
	97.0% 3.0%	97.4% 2.6%	97.1% 2.9%

All Confusion Matrix

Output Class \ Target Class	1	2	
1	444 63.5%	8 1.1%	98.2% 1.8%
2	14 2.0%	233 33.3%	94.3% 5.7%
	96.9% 3.1%	96.7% 3.3%	96.9% 3.1%

Figure 3: Confusion Matrix of the Intelligent Cyber threat detection system

From the confusion matrix in figure 3, the overall average precision reported 96.9%, recall was 96.7%, accuracy was 96.9%, sensitivity was 98.2% and specificity 94.3%. From the evidence of these set of results, it was observed that there is a level of consistencies in the results achieved during the training of the training and evaluation of the ICDS model generated for threat detection in the IIoT facility.

Overall, these results demonstrate the ICDS model's ability to accurately identify positive samples (potential threats) among all instances it classified as positive, with a very low rate of false positives (specificity of 94.3%). The high recall rate (96.7%) indicates that the ICDS model effectively detects most of the actual threats present in the IIoT facility, minimizing false negatives. Furthermore, the model's exceptional accuracy (96.9%) showcases its overall proficiency in correctly classifying both positive and negative samples.

6.1 Result of system Integration with IDCS on the IIoT

This section presented the performance of the improved firewall developed with the ICDS. The IDCS was integrated on the conventional firewall and then tested with denial of service. The process first allows the normal packet network flow for 10mins, then denial of service was introduced to the network using the metaspolit tool, and the performance of the monitored with the wireshark tool considering the volume of packet, loss, latency, throughput, bandwidth utilization factor and reported in table 1;

Table 1: Result of the system integration

Experiment (min)	Packet Data Size (Mb)	Latency (ms)	Loss (%)	Throughput (%)	Bandwidth Utilization Factor (%)
1	21.3420	209.22	2.3231	89.6757	0.426848
2	22.3235	209.22	2.4677	89.0673	0.429292

3	22.4332	223.25	2.5232	88.2353	0.446478
4	23.0754	232.35	2.5675	87.6546	0.448664
5	23.4534	234.32	2.6430	86.2334	0.450688
6	24.3423	234.33	2.7343	85.6456	0.461508
7	26.4646	243.45	2.8234	85.3345	0.469068
8	27.5344	276.73	3.3564	85.3045	0.486846
9	28.0990	301.24	3.3654	85.2034	0.521673
10	28.3421	301.24	3.3670	84.6655	0.521824
11	41.0912	330.24	3.4546	84.0901	0.521824
12	41.0912	339.25	3.4645	79.0923	0.524224
13	42.0912	340.95	3.4745	79.0923	0.541570
14	42.0912	341.28	3.4765	79.0284	0.541824
15	42.4332	342.35	3.5231	78.8344	0.541824

The table 1 presented the performance of the network with ICDS during test with denial-of-service attack. From the result, overall, it was observed that considering the ITU standard for the IIoT network analysis, the average throughput, latency, loss and utilization factor was good, despite the denial-of-service attack. What actually happened in this result was that the after ICDS based firewall was able to detect the threat on the network and isolate from the server.

7. CONCLUSION AND RECOMMENDATION

This study focused on the development of a proactive defence mechanism for identifying and responding to zero-day attacks effectively in Industrial Internet of Things (IIoT) systems. Significant contributions to the field of IIoT security was made by addressing the issue of network security with equal importance as process control systems. The ICDS model represents an innovative approach to proactively detect and respond to emerging cyber threats in IIoT systems. The evaluation of the ICDS model's performance revealed promising results in terms of network latency, loss, throughput, and bandwidth utilization factor. The validation of ICDS demonstrated a significant reduction in latency by approximately 70.82%, indicating improved responsiveness and data transmission efficiency. Moreover, the model successfully reduced data loss by approximately 75.65%, ensuring the integrity and completeness of transmitted information from the process design network. Furthermore, the ICDS model's positive impact on throughput was evident, achieving an improvement of approximately 21.80%. This enhancement in data transfer rates contributes to more efficient and effective utilization of network resources. Additionally, the model effectively optimized the bandwidth utilization factor by approximately 46.46%, leading to better allocation and management of available network bandwidth. These quantitative improvements further highlight the effectiveness and viability of the ICDS model as a proactive defence mechanism for IIoT security. By not only strengthening the network's security attacks but also demonstrating significant enhancements in network performance metrics, the ICDS model stands as a valuable asset in ensuring the overall robustness and efficiency of IIoT systems in industrial applications.

7.1 Recommendations

This study recommends the ICDS model to be practically applied by the Cisco company to upgrade their firewall to ensure more robust security outcome. Secondly the recommended solution or the vulnerability should be immediately applied to the IIoT network for protection against zero-day attack.

8. REFERENCES

Alshahrani, H.; Khan, A.; Rizwan, M.; Reshan, M.S.A.; Sulaiman, A.; Shaikh, A. Intrusion Detection Framework for Industrial Internet of Things Using Software Defined Network. Sustainability **2023**, 15, 9001. https://doi.org/10.3390/ su15119001

Bae, G., Jang, S., Kim, M., and Joe, I. (2019). Autoencoder-Based on Anomaly Detection with Intrusion Scoring for Smart Factory Environments. Springer, pp. 414-423.

Ibitoye, O., Shaq, O., and Matrawy, A. (2019). Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). doi:10.1109/GLOBECOM38437.2019.9014337.

Li, Y., Xu, Y., Liu, Z., Hou, H., Zheng, Y., Xin, Y., and Cui, L. (2019). Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Measurement, 154, Article 107450. doi:10.1016/j.measurement.2019.107450.

Mbanaso U., Victor Kulugh, Habiba Musa and Gilbert Aimufua (2019)"Conceptual Framework for the Assessment of the Degree of Dependency of Critical National Infrastructure on ICT in Nigeria" 15th International Conference on Electronics Computer and Computation (ICECCO 2019) 978-1-7281-5160-1/19

Mohd K. and Al-Kadhimi, Amjed and Singh, Manmeet. (2023). Recent Developments in Game-Theory Approaches for the Detection and Defense against Advanced Persistent Threats (APTs): A Systematic Review. Mathematics. 11. 1353. 10.3390/math11061353.

Mvula P., Branco P., Jourdan G., and Viktor H., (2023) "A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning", Springer: Review Discover Data (2023) 1:4 | https://doi.org/10.1007/s44248-023-00003-x

Ndungu G. (2021) "Detecting zero-day attacks using Recurrent Neural Network" Faculty of Information Technology StrathmoreUniversity [Thesis, Strathmore University]. http://hdl.handle.net/11071/12942

Patidar C., and Khandelwal H., (2018) "Zero Day Attack Detection using Machine Learning Techniques", IJRAR19J1648 International Journal of Research and Analytical Reviews (IJRAR) Pp 1364-1368

Peppes N., Alexakis T., Adamopoulou E., andDemestichas K., (2023)"The Effectiveness of Zero-Day Attacks Data Samples Generated via GANs on Deep Learning Classifiers", Sensors 2023, 23, 900. https://doi.org/10.3390/s23020900