BIU Journal of Basic and Applied Sciences 10(1): 1 – 12, 2025.

ISSN: 2563-6424

ENHANCED MODEL FOR INTRUSION DETECTION

***ODIKAYOR-OGBOMO, I. F., INYANG, A. I. AND UKUEJE, P. O.**
Department of Computer Science, Benson Idahosa University, Edo State, Nigeria
*Corresponding author: odogbomo@biu.edu.ng

ABSTRACT

The cyber security industry is one of the fastest growing today. As networks get larger, the attack surface for hackers increases, making cyber risk a prevalent problem. As hackers and attackers get sophisticated, the defense to prevent such attacks must be sophisticated as well. Thus, a necessity was created to develop a model that can learn the structure of network data and differentiate normal from abnormal network traffic. This research adopted the Zhang 2019 model which uses the Deep Generative Neural Network (DGNN) that performs adversarial learning, a process in which a machine learning model is fooled by being given spoofed input. DGNN has two components: G-net and D-net. G-net generates malicious inputs (intrusion samples) while D-net identify the malicious inputs from the real inputs. As the model continues to run, G-net will generate better malicious inputs as D-net get better at identifying them. The developed model adopted the Zhang 2019 model and enhanced it by combining two machine learning techniques: Support Vector Machine and Bayesian network by normalizing and classifying all data packets. Data are collected and moved to the data pre-processing stage where they are normalized and classified into normal data (ND) or intrusion data (ID) using machine learning techniques. Both the normal data and intrusion data are sent to the data partition stage where the normal data is classified as normal network request while the intrusion data (network intrusion) go through data Augmentation process of Zhang and finally to the Deep learning training model. This is the training phase. The testing phase includes feeding the model with test data which was obtained using a data packet receiver. The result is gotten from classification of the intrusion data as True positive, false positive, false negative and True negative from which a comparison/performance evaluation was done. The performance evaluation shows high rate of True positive and true negative values. The enhanced model developed when implemented, was found to be able to make strong predictions, detects intrusion attacks, unauthorized users (intruders), provide information about malicious network traffic and alert system personnel that a network invasion maybe in progress.

KEYWORDS: *Internet, Intrusion, Detection, Computing, Machine learning*

INTRODUCTION

The days when only strong passwords and firewalls were all that was required to secure corporate networks, have long passed. Data integrity cannot be protected from outside intruders in today's Internet environment using common mechanisms. Intruder attack methodology has become more targeted and sophisticated, measures beyond those

normally expected of an intranet system should always be made on any system connected to the internet. Intrusion detection system takes that one step further by providing an extra layer of protection to a system. An intrusion detection system (IDS) monitors and analyzes data to detect any intrusion in a system or network. An intrusion detection system identifies malicious activity, detect external/internal hackers, network-based attacks and alerts network administrators or responds by taking predefined action. The challenge of modelling the behaviour of normal and abnormal network traffic became a necessity. Hence a model for intrusion detection is required to learn the structure of network data and differentiate normal from abnormal network traffic. To achieve this, an enhanced model for intrusion-detection was developed. The model adopted the (Zhang, 2019) model and enhanced it by combining two machine learning techniques (support vector machine and Bayesian network) to aid in the classification of normal data and intrusion data. The model will learn the necessary features of a dataset to make strong predictions and improve the existing system in terms of providing more accurate and more efficient intrusion detection.

RELATED WORKS

Kleber and Schulter (2010) proposed a hybrid intrusion detection system for cloud and grid environment. This system cannot be deployed into a real time distributed environment, as the system cannot synchronize well with the other intrusion detection systems in the network and the architecture and working of both models is completely different. Tupakula *et al.* (2011) have proposed a Hybrid Intrusion Detection System for Infrastructure as a Service Cloud. This system cannot handle large scale, dynamic, multithread and data processing environment. Since the system has been proposed for Infrastructure as a Service Cloud, the synchronization character is not applicable to the system. Software as a Service and Platform as a Service are the other two services of cloud, which has not been considered by the authors. Hisham and Baiardi. (2012) proposed a framework for intrusion detection in cloud systems. This framework can partially handle large scale, dynamic data which is another drawback of the system. The authors did not narrate the scope for implementing the algorithm for the private cloud environment. Zamani and Movahedi (2013) presented a review article on some influential algorithms based on machine learning approaches used in intrusion detection. Zamani explored that using a machine learning approach for intrusion detection enables a high detection rate and low false-positive rate with the capabilities of quick adaptation toward changing intrusive behaviour. The analyzed algorithms have been categorized into artificial intelligence (AI) and computational intelligence bases. Elike *et al*. (2017) presented a taxonomy that classifies DL (Deep learning) models into generative and discriminative architectures. However, the authors also note that both CNN (Convolutional Neural Network) and DBN (Deep Neural Network) have not been exploited in the field of IDS (Intrusion Detection System) to detect attacks. Further, the authors compared the performance of DL and shallow learning models in the field of IDS. Aleesa *et al.* (2019) reviewed and

analyzed the research landscape for intrusion detection systems (IDSs) based on deep learning (DL) techniques and identified the research gap. They focused on deep learning, intrusion and attack and their variations in four major databases. Zhang *et al.* (2019). Proposed a system to detect intrusion using Deep Generative Neural Network (DGNN) that performs Adversarial learning with Data Augmentation in intrusion detection. With the use of data Augmentation in intrusion detection, the detection rate and precision were better. But when there is not enough test data for the system to train with, the system experiences data scarcity and imbalance. Zeeshan *et al.* (2021) discussed the cyber security technology trends in intrusion detection utilizing ML (Machine Learning) and DL (Deep Learning) methods. However, the present work does not cover all the methods in the intrusion detection domain; furthermore, the authors use few benchmark datasets for the model, and the analysis is not uniform. None of the work covers a deep and insightful analysis of the performance of the model. Adel *et al*. (2022) identified the power of various machine learning (ML) algorithms and analyzed the effect of ML algorithms for intrusion detection. Sudhanshu and Bichitrananda (2023) discussed intrusion detection system using machine learning techniques with the use of KDD CUP '99' Intrusion detection dataset for training and validating machine learning models. Alamin *et al,* (2023) introduced a hybrid machine learning model to enhance network intrusion detection by combining machine learning and deep learning to increase detection rates while securing dependability. Synthetic minority oversampling technology (SMOTE) was used for data balancing and XGBoost (Extreme Gradient Boosting) for feature selection.

METHODOLOGY

The methodology adopted is the Object-Oriented Analysis and Design (OOADM). This methodology focused on the definition of classes and the manner in which they collaborate with one another to effect customer requirements. Unified modeling language (UML) and the Unified Process are predominantly features of the methodology adopted.

System Design

The Zhang Model

Deep Generative Neural Network (DGNN) are neural networks that perform adversarial learning. Adversarial learning is the process in which a machine learning model is fooled by being given spoofed input. Thus, the DGNN has two components: G-net and the D-net. Both, the G-net and the D-net are Deep Neural Networks (DNN). The G-net tries to generate malicious inputs, which in this case are intrusion samples, while the D-net tries to identify the malicious inputs from the real inputs. As the model continues to run, G-net will generate better malicious inputs as D-net gets better at identifying them (Zhang, 2019). The DGNN, through adversarial learning, will converge to create augmented intrusion data which is close to real intrusion data. This augmented data is now mixed in with normal data in the Data Augmentation module then to the shallow or deep learning machine module, which is the training phase. Figure 1 shows a model of the described Zhang model. (Zhang, 2019).

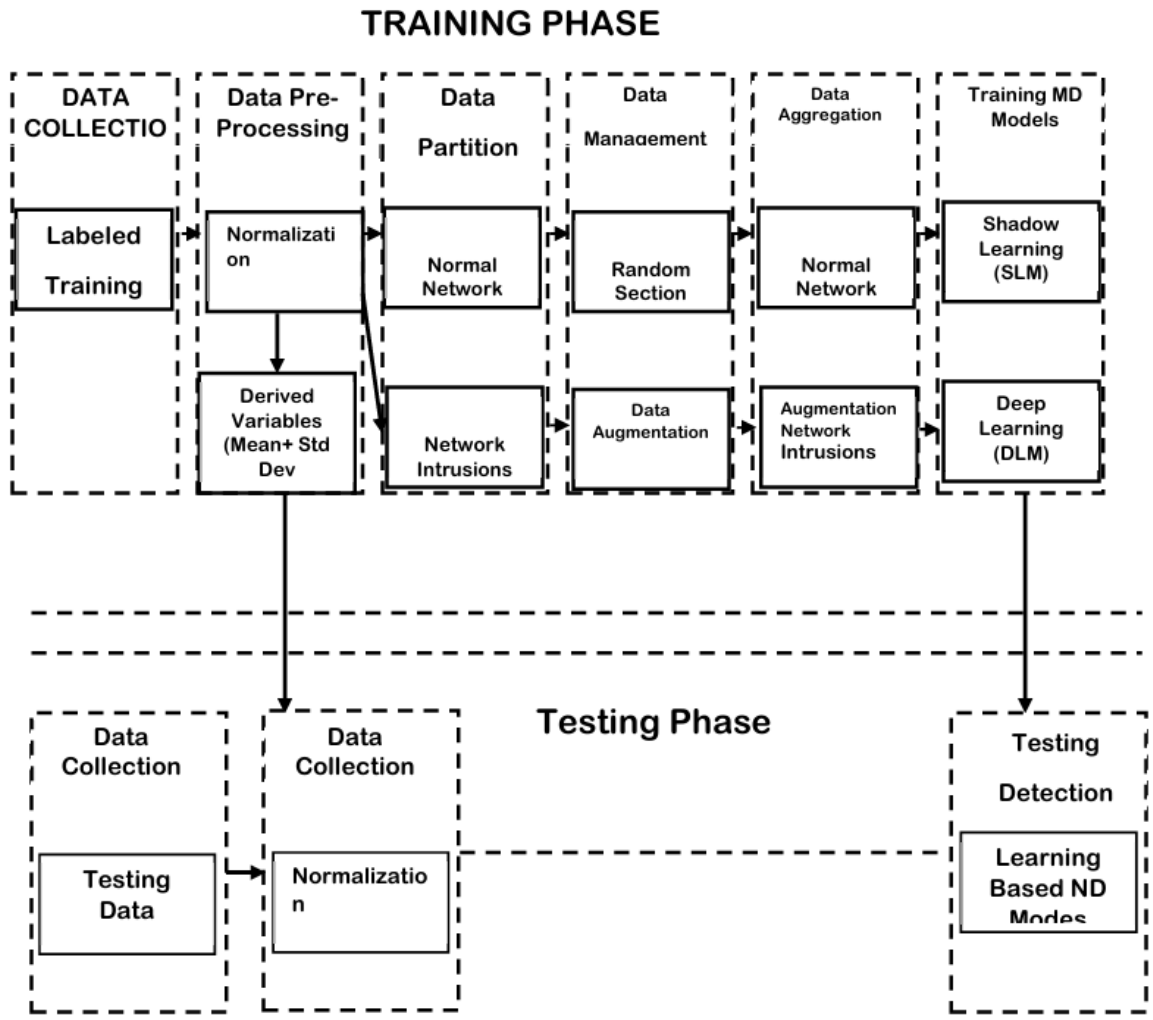

Fig. 1: Network Intrusion Detection model using Data Augmentation (Zhang, 2019)

The New System

The new system adopted the Zhang model and enhanced it by combining two machine learning techniques (support vector machine and Bayesian network) to aid in the classification of normal data and intrusion data. From figure 2. Data are collected and moved to the Data pre-processing stage, where they are normalized, to reduce the height and volume of the data, and then they are classified into Normal Data (ND) or Intrusion Data (ID) using the two-machine learning technique (support vector machine and Bayesian network). Both the Normal Data and Intrusion data are sent to the next stage which is the Data Partition where the Normal Data are now classified as Normal Network Request while the Intrusion data are classified as Network Intrusion. The Normal Network requests are sent through the Date management and Data aggregation process to the shallow learning, while the Network Intrusion (Intrusion Data) go through Data Augmentation process of Zhang (Zhang *et al*., 2019) and finally to the Deep Learning Training Model. The above description

constitutes the Training Phase for the models.

The testing phase include feeding the model with test data, which are then normalize and classified and a result of weather they are intrusion Data (ID) or Normal Data (ND) is made based on the result from the shallow learning process or Deep learning process of the trained model.

Model of the New System

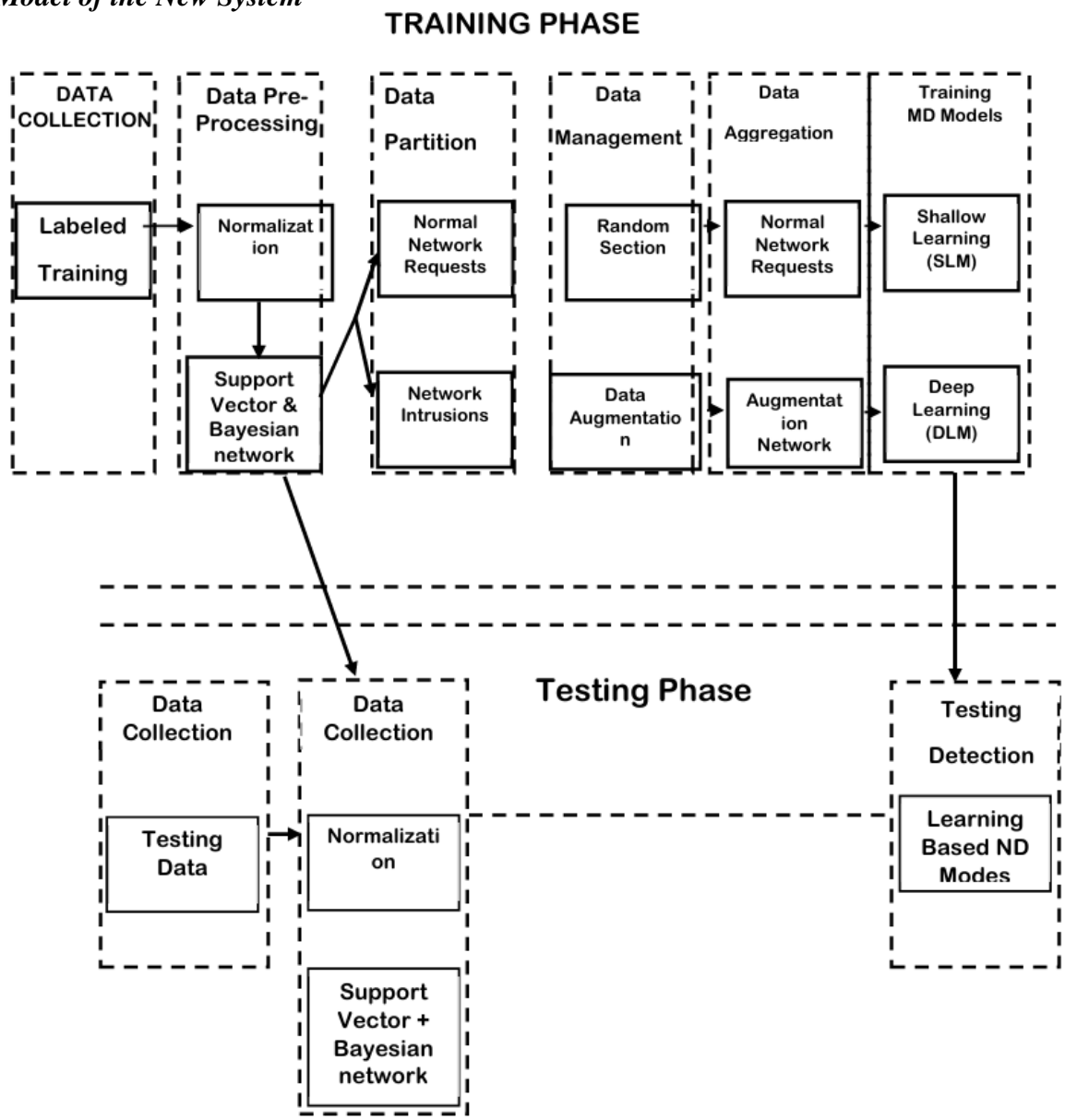

Fig. 2: Model View of the New System

High level Model of the New System

The high level model explains the architecture that would be used for developing the automated system. The high level diagram shown in figure 3 provides an overview of the entire system, identifying the main components that would be developed and their interfaces.

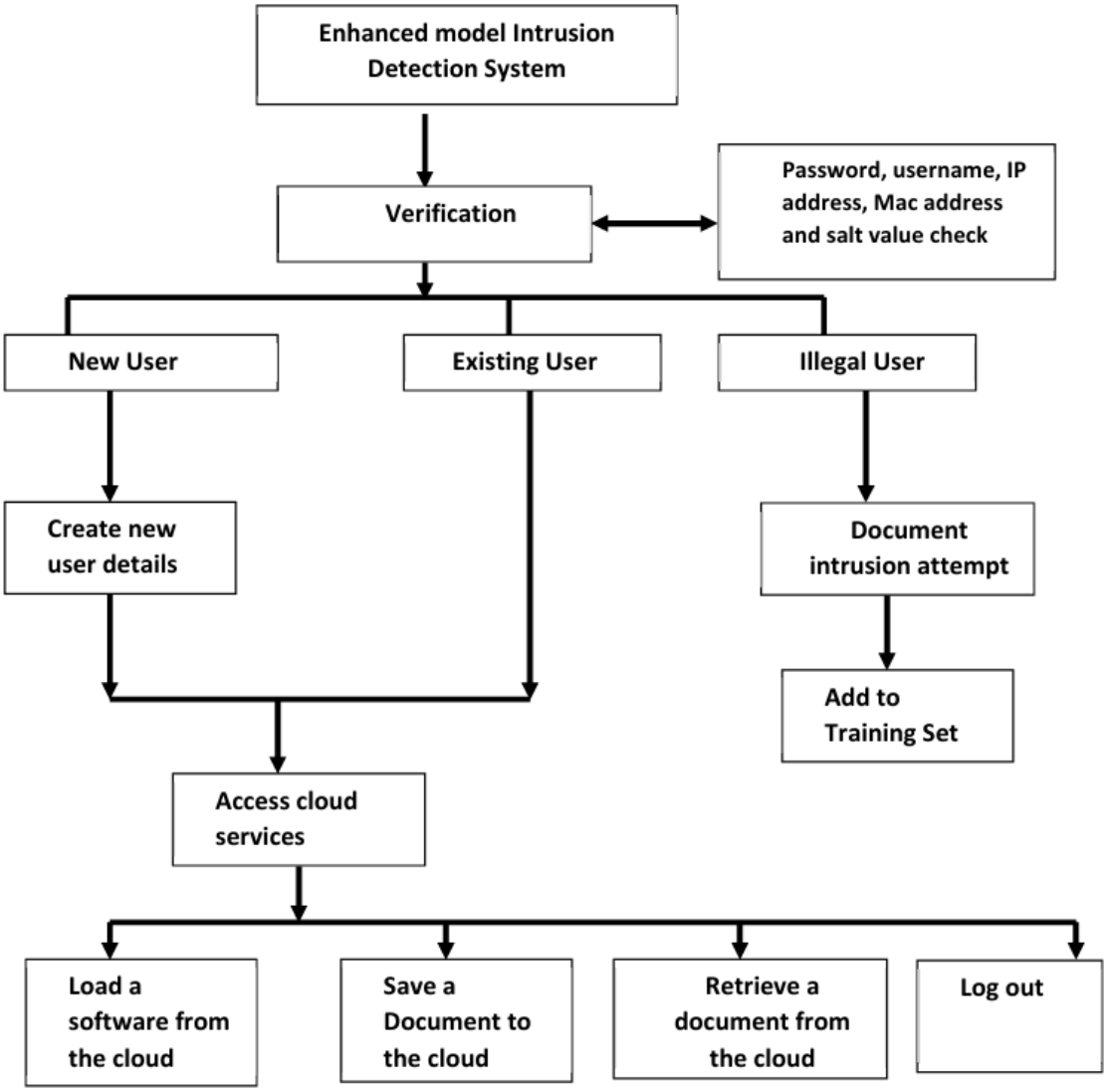

Fig. 3: High level Model view of the new system

Sequence Diagram of the New System

The Sequence Diagram of the system, model the time sequencing of message within the system and show how objects interact with message. It can be used to show or describe the detail implementation of system behaviours as shown in figure 4.

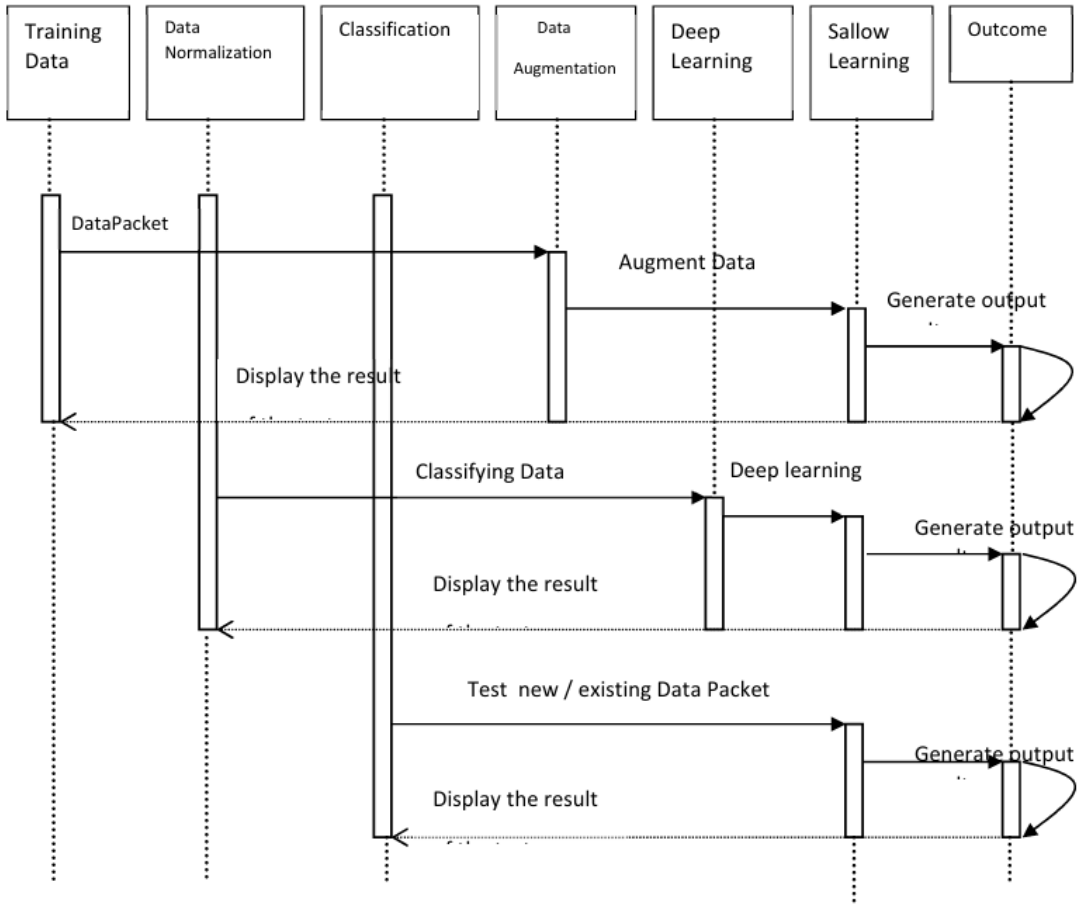

Fig. 4: Sequence diagram for the enhanced intrusion detection system

The dashed lines hanging down from the object and the actor are called life lines. A message being sent from the object to another is shown as arrow between the life lines. Each message is labelled with a name as shown in figure 4. Argument appears either in the parenthesis that follows the name or next to data tokens. Time is in vertical dimension, so the lower a message appear the later it is sent as illustrated in figure 4.

Class Diagram of the New System

The class diagram of the system allows us to denote the content and relationship between classes. A class is depicted on the class diagram as a rectangle with three horizontal sections. The upper section shows the class name (such as Training Data, Test Data) the middle section contains the class attributes that shows the various properties of the class, and the lower section contains the class functions or operations performed by the class.

Rectangles represent classes and arrows represent association in which one object holds a reference to and invokes methods from the other. A dash (-) character in front of the variable in the class icon denotes private. A (+) character in front of the function or operation in the class icons denotes public. The type of variable or a function argument is shown after the colon following variable or argument name as illustrated in figure 5.

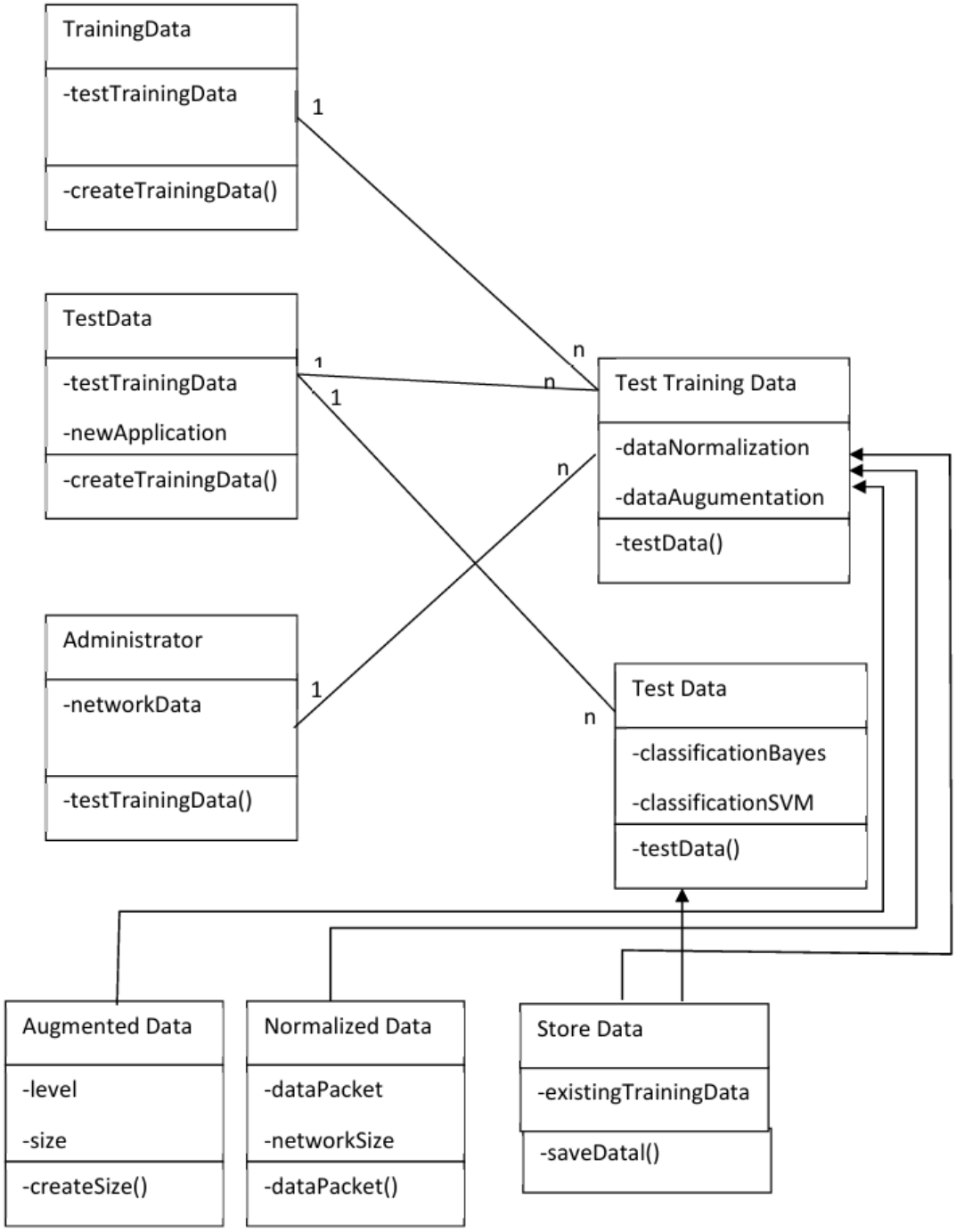

Fig. 5: Class Diagram for the enhanced intrusion detection system

RESULTS AND DISCUSSION

The model adopted the Zhang model and enhanced it by combining two machine learning techniques (support vector machine and Bayesian network) to aid in the classification of normal data and intrusion data at the data pre-processing stage during normalization of data as

shown in figure 2 to enhance the Zhang model.

A test data was obtained with the use of data packet receiver and classification of the intrusions as True positive, false positive, true negative and false negative as shown in table 1. Each frame was analyzed for true positive and true negative as shown in table 2, the result gotten from the classification and analyses of the intrusions were used to compare and evaluate the performance of the enhanced intrusion detection model to ascertain if it was better than the Zhang model as shown in table 3. The True positive rate and false positive rate as shown in table 4 was gotten from table 3 using Excel's built-in functions. From table 3 and 4. It shows that the new enhanced intrusion detection system is more reliable than that of Zhang, as shown in the high rate of true positive and true negative value.

Table 1. Classification of intrusions

TRUE POSITVE	**FALSE POSITVE**
Reality: An intrusion Attack occurs **Enhanced IDS:** Detects an Attack occurs **Output:** Record the attack as TP	**Reality:** No intrusion Attack occurs **Enhanced IDS:** Detects an Attack occurs **Output:** Record the attack as FP
FALSE NEGATIVE	**TRUE NEGATIVE**
Reality: An intrusion Attack occurs **Enhanced IDS:** Does not Detects an Attack **Output:** Record the attack as FN	**Reality:** No intrusion Attack occurs **Enhanced IDS:** Does not Detects an Attack **Output:** Record the attack as TN

Table 2: Showing the Analysis of each Frame for True Positive and True Negative

ID	**Frame**	**Byte**	**Result**
1	Frame 1:	208	2.05807365439093
2	Frame 2:	42	1
2	Frame 3:	42	1
4	Frame 4:	208	2.05807365439093
5	Frame 5:	208	2.05807365439093
6	Frame 6:	208	2.05807365439093
7	Frame 7:	208	2.05807365439093
8	Frame 8:	208	2.05807365439093
9	Frame 9:	208	2.05807365439093
10	Frame 10:	208	2.05807365439093
11	Frame 11:	208	2.05807365439093
12	Frame 12:	92	1.31869688385269
13	Frame 13:	92	1.31869688385269
14	Frame 14:	208	2.05807365439093
15	Frame 15:	92	1.31869688385269
16	Frame 16:	80	1.24220963172805
17	Frame 17:	158	1.73937677053824
18	Frame 18:	66	1.15297450424929
19	Frame 19:	66	1.15297450424929
20	Frame 20:	54	1.07648725212465
21	Frame 21:	54	1.07648725212465
22	Frame 22:	72	1.19121813031161

Performance Evaluation

Table 3: Performance evaluation carried out on the new system

Number of Packet Frame Analyzed	Number of True Positive	Number of True Negative
12362	1908	10454
6120	954	5166
12240	1908	10332
6006	945	5061
5945	945	5000
5823	943	4880
11829	1888	9941

Analysis presenting TPR (True Positive rate) and FPR (False Positive rate)

Table 4: Showing True positive rate (TPR) and False positive rate (FPR)

Row Labels	Grand Total	TPR=TP/(TP+FN)	FPR=FP/(FP+TN)
4880	943	0.161944	0.838056
5000	945	0.158957	0.841043
5061	945	0.157343	0.842657
5166	954	0.155882	0.844118
9941	1888	0.159608	0.840392
10332	1908	0.155882	0.844118
10454	1908	0.154344	0.845656
Grand Total	9491		

Receiver Operating Characteristic (ROC) Curve

The ROC (Receiver Operating Characteristic) curve of the system is a graphical representation of the performance of the binary classification model. It plots the True positive Rate (Sensitivity) against the False Positive Rate (1- Specificity).

ROC curve helps to evaluate the model's ability to distinguish between positive and negative classes and compare the performance of the different models. It also evaluates the performance of predictive models and classification algorithms. As shown in Table 5 and figure 4. The ROC curve was created using Excel's built-in functions.

Table 5. TPR and FPR values

True positive rate (TPR)	False Positive rate (FPR)
0.161944	0.845656
0.159608	0.844118
0.158957	0.844118
0.157343	0.842657
0.155882	0.841043
0.155882	0.840392
0.154344	0.838056

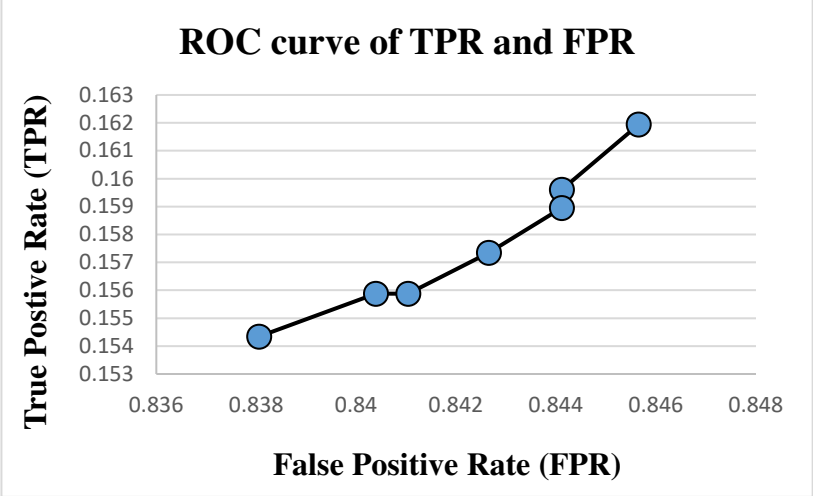

Fig. 4: The ROC curve of TPR and FPR

CONCLUSION

Networks security problems vary widely and can affect different security requirements including authentication, integrity, authorization, and availability. Intruders can cause different types of attacks on systems in an organization. These attacks need to be detected as soon as possible to prevent further damages to organizations sensitive data which may cause financial loss.

This research adopted the Zhang model (Zhang *et al.*, 2019) and enhanced it by combining the model with two machine learning techniques (support vector machine and Bayesian network) to aid in the classification of normal and intrusion data during normalization at the pre-processing stage of the training phase of the model. By this process improved on the Zhang model in terms of providing more accurate and more efficient intrusion detection. It also provided worthwhile information about malicious network traffic; helping to identify intruders and alerting system personnel (administrators) that a network invasion may be in progress.

REFERENCES

Adel, B., Haya, A., Thavavel, V. and Dinesh, M. (2022). An investigation and comparison of ML approaches for intrusion detection in IoMT network. *J. Supercomput*., 78: 17403-17422.

Alamin, T. M., Khondokar, F. H., Manowarul, I., Ashraf, U., Arnisha, A., Mohammand, A.Y., Fares, A. and Mohammad, A. M. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, 72: February 2023. 103405 Elsevier.

Elike, H., Xavier, B., Andrew, H., Christos, T. and Robert. A. (2017). Shallow and deep networks intrusion detection system: a taxonomy and survey. arXiv preprint arXiv: 1701.02145 (2017). 1-43.

Hisham, A.K. and Baiardi F. (2012). CIDS: A framework for intrusion detection in cloud systems. *Proceedings of 9th IEEE International Conference on*

Information Technology-New Generations. 379–85.

Kleber, S. (2010). "Intrusion Detection for Grid and Cloud computing", IEEE Journal: IT Professional.

Sudhanshu, S. T. and Bichitranand, B. (2023). Performance Evaluation of Machine Learning Algorithms for Intrusion Detection System. *Journal of Biomechanical Science and Engineering. Japan Society of Mechanical Engineers*. ISSN: 1880-9863. DOI 10.17605/OSF.IO/WX6CS 621 April 2023.

Tupakula, U., Varadharaja, V. and Akku, N. (2011). Intrusion detection techniques for infrastructure as a service cloud. Proceedings of 9th IEEE International Conference on Dependable, Autonomic and Secure Computing. p. 744–51.

Zamani, M. and Movahedi, M. (2013). "Machine learning techniques for intrusion detection" *arXiv:* 1312.2177.

Zeeshan, A., Adnan-Shahid, K., Wai hiang, C., Johari, A. and Farhan, A. (2021). Network intrusion detection system: a systematic study of ML and DL approaches. *Transactions on Emerging Telecommunications Technologies*, 32: Article e4150

Zhang, H., Xingrui, Y., Peng, R., Chumbo, L. and Geyong, M. (2019). Deep Adversarial learning in intrusion detection. A data Augmentation Enhanced Framework. *ArXiv*: 1901.07949v3 [CS.CR]