Artificial Intelligence for Intrusion Detection in Internet of Things

Syed Abidullah Fareed[1], Abdul Jabbar Siddiqui[2]
IRC for Intelligent Secure Systems
Computer Engineering Department, King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
[1]s202185410@kfupm.edu.sa , [2]abduljabbar.siddiqui@kfupm.edu.sa

***Abstract*—Neural networks have grown in popularity as a solution for network intrusion detection systems (NIDS). Because of their ability to learn complicated patterns and behaviors, they are an ideal solution for distinguishing between normal traffic and network attacks. However, one disadvantage of neural networks is the large number of resources required to train them. Many network gateways and routers, which can potentially host a NIDS, lack the memory or computing power to train and execute such models. More crucially, present neural network systems need an expert to manually identify network traffic and update the model on a regular basis. One approach to tackle this problem has been proposed, which is to use a plug and play NIDS that can learn to detect attacks on the local network, without supervision in an efficient manner. However, the rate of anomaly detection in this system can be significantly improved by embedding dynamic ensemble selection classifier (DES) in the ensemble layer of Anomaly detector.**

Keywords— Neural networks, memory, dynamic ensemble selection classifier, anomaly detection

I. INTRODUCTION AND RELATED WORK

The use of neural networks in network intrusion detection systems (NIDS) is becoming more widespread. They have become a perfect choice for differentiating between regular traffic and network attacks due to their capacity to learn complex patterns and behaviors. On the other hand, the significant number of resources needed to train neural networks is one major drawback. Many network gateways and routers, which may be used to host NIDS, are not equipped with the memory or processing capacity to build and run such models. More importantly, current neural network systems require a specialist to manually detect network activity and regularly update the model.

In [1][2] types of intrusion detection systems have been presented along with a brief introduction to the categories of Anomaly detection which is one of the types of Intrusion Detection Systems (IDS). However, because these methods make no assumptions about the machine learning model parameters, they are either too expensive to train and execute on simple gateways or require a labeled dataset to accomplish the training process.

In [3], an effort was made to propose an online anomaly detection mechanism using a lightweight algorithm. This work proposes a new KNN (k-nearest neighbour)-based AD technique for wireless sensor network applications based on hypergrid intuition. The computational complexity is greatly decreased by redefining anomaly from a hypersphere detection area (DR) to a hypercube DR. The weakness of this model is that it either produces unsatisfactory results or requires accumulating large amounts of data for training which is not feasible for simple network gateways.

The Artificial Neural Network (ANN) is a common algorithm for detecting network intrusions. This is due to its capability to learn complicated concepts as well as concepts from the network communication domain.

In [4] The authors have assessed the ANN, as well as other classification algorithms, for the purpose of network intrusion detection and presented a solution based on an ensemble of classifiers using connection-based features. The ANNs used in this paper are either supervised or unsuitable for usage as a simple network gateway.

An autoencoder is an artificial neural network which is trained to reconstruct its inputs (i.e., X = Y). In [5], an innovative object tracker was presented by combining the deep learning technique with the on-line boosting framework. To learn a multi-level image feature descriptor, the authors first implemented a stacked denoising autoencoder (SDAE). The SDAE layers are then combined to form a family of discriminative DNN classifiers. These classifiers are then used with an alternate type of on-line boosting to aid in object/background classification. Despite using autoencoders in an online scenario, the authors did not perform anomaly detection or address the difficulty of real-time processing (which is great challenge with deep neural networks). Moreover, training a deep neural network is challenging and cannot be done on a simple network device. The writers of [6] and [7] suggested using autoencoders to extract features from datasets to improve the detection of cyber threats. Anomaly identification, however, did not use the autoencoders themselves. In the end, classifiers are used by the writers to find the cyber threats. Hence, this method requires professionals to

label the packets which is not very feasible when dealing with large network of IOT traffic.

Reference [8] tanzila presents a CNN-based approach for anomaly-based intrusion detection systems (IDS) that takes advantage of IoT's power, providing qualities to efficiently examine whole traffic across the IoT. The proposed model shows ability to detect any possible intrusion and abnormal traffic behavior. The model is trained and tested using the NID Dataset and BoT-IoT datasets and achieved an accuracy of 99.51% and 92.85%, respectively.

Zhou [9] tested Decision Tree (DT), Gaussian Naive Bayesian (GaussianNB), K-Nearest Neighbor (Knn), Support Vector Machine (SVM) and LogisticRegression on his novel feature embedding method. The above mentioned models achieved 98.82%, 98.77%, 98.85%, 98.8%, and 98.86% accuracy using the NSL-KDD dataset, respectively. Additionally, the models attained 91.9%, 92.29%, 90.35%, 92.52%, and 92.32% accuracy using the UNSW-NB15 dataset.

Works on Recurrent Neural Networks (RNN), a Deep Learning (DL) algorithm, were done by Putchala [10]. In order to reduce the complexity of the system, this model used a single hidden layer and a single node with a Gated Recurrent Unit (GRU). However, the author took into account the KDD99 dataset, which is not an IoT dataset. The proposed neural network model using all network layers attained 0.9891 and 99.59% F and accuracy scores, respectively, when all the dataset features are employed.

Lopez [11] proposed an integrated system that merged Convolutional Neural Networks (CNN) with RNN, eliminating the need for feature engineering. The proposed IDS employed CNN to extract features from a real network traffic dataset called RedIRIS. Accordingly, the RNN was used as a detection model. The highest accuracy and F scores were 96% and 0.96.

A two-phase system: OpCode sequence Graph Generation and Deep Eigenspace Learning (DEL) Phase was proposed by Amin [12]. The IDS utilized the OpCode to transform the traffic into a graph. Subsequently, it applied DEL to learn the relations among vertices. The system attained 0.9848 and 99.68% F and accuracy scores on a local dataset, respectively. Abeshu [13] utilized a stacked autoencoder with backpropagation and SGD. This proposed IDS attained 99.2% accuracy on the NSL-KDD dataset. Feng [14] utilized the Deep Belief Network (DBN). The DBN model was made up of two layers limiting the Boltzmann machine. Feng IDS attained 95.25% accuracy using the NSL-KDD dataset.

Kitsune, a novel NIDS based on neural networks, was developed in [15] for the effective detection of anomalous patterns in network traffic on simple network routers in real-time. Kitsune was therefore created with a minimal memory footprint and a simple computational structure. Kitsune's main algorithm (KitNET) detects odd traffic patterns by embedding an ensemble of autoencoders. KitNET is supported with a feature extraction framework that tracks the patterns of each network channel efficiently. Each autoencoder in the ensemble oversees detecting abnormalities in a particular aspect of the network's behaviour. According to the authors, this algorithm performed almost as well as, and in some cases better than, other batch / offline algorithms.

However, the performance of this NIDS, Kitsune, can significantly be improved by selecting only prominent outputs of the autoencoders in ensemble layer of kitNET, which is the anomaly detector of Kitsune's framework. The use of a dynamic ensemble selection classifier could accomplish this. Our study is largely focused on achieving the aforementioned objectives, and the data we have gathered shows that such a system is remarkably efficient while maintaining a high level of anomaly detection accuracy.

II. Problem Statement and Research Objectives

A. Describing the Problem

Due to the process involved in the electronic transformation of data, the usage of computer systems and the Internet in recent times has resulted in major security, privacy, and confidentiality difficulties. Even though there has been a lot of work made into enhancing computer systems' security and privacy, these issues still persist. In actuality, there isn't a system in existence that is 100 percent secure. There are also many kinds of network attacks, which develop whenever a new signature with unusual behavior appears in the database of signatures. Several tools are being developed and employed in various sorts of network attacks as a result of the introduction of numerous attack types. The network intrusion detection systems are one of these tools (NIDSs). A variety of network systems, cloud computing platforms, and information systems can all be monitored with this tool.

Network intrusion detection systems (NIDS) are increasingly using neural networks. Due to their potential to understand intricate patterns and behaviors, they have emerged as the ideal option for distinguishing between normal traffic and network attacks. Unfortunately, one notable disadvantage is the considerable number of resources required to train neural networks. Numerous network gateways and routers, which may be used to host NIDS, lack the memory or processing power necessary to create and use such models. More crucially, modern neural network systems necessitate the manual detection of network activity and frequent model updating by a professional.

One approach to deal with this problem has been the implementation of a plug-and-play NIDS (Kitsune) that can learn to detect attacks on wireless networks. However, by including a dynamic ensemble selection classifier (DES) in the ensemble layer of kitsune's algorithms, the rate of anomaly identification in this system can be greatly increased as well as reducing energy consumption.

B. *Scope of the Research Project*

To Study and investigate potential enhancements of an autoencoder neural network-based network intrusion detection method through the application of dynamic ensemble selection to reduce computation costs and energy consumption while maintaining a high-level of accuracy.

C. *Research Goals and Objectives*

- To review anomaly detection and autoencoder-based state-of-the-art methods to detect network intrusions.
- Reproduce results of a state-of-the-art autoencoder-based method using a real-world Internet of Things dataset.
- Research and investigate dynamic ensemble selection as a potential means to improve performance of the selected autoencoder-based intrusion detection method.
- Compile the results and findings into a potential conference/journal publication.

III. RESEARCH PLAN AND EXECUTION

The research has been executed in the following manner:

1. To study about neural networks and deep learning.
2. Review anomaly detection and autoencoder-based state-of-the-art methods to detect network intrusions.
3. Reproduce results of a state-of-the-art autoencoder-based method using a real-world Internet of Things dataset.
4. Research and investigate custom ensemble selection as a potential means to improve performance of the selected autoencoder-based intrusion detection method.
5. Research and investigate about dynamic ensemble selection.
6. Compile the results and findings.

A. *To Study About Neural Networks and Deep Learning.*

Studying about neural networks and deep learning from recorded lectures and PowerPoints.

- Architecture of neural networks and their flow of operation.
- State of the art algorithms used in neural networks e.g.Gradient descent algorithm, stochastic gradient descent algorithm
- Autoencoders and their specialty.
- Application of neural networks for detecting malicious packets.

B. *Reproduce Results of A State-of-the-art Autoencoder-Based Method Using A Real-World Internet of Things Dataset.*

1. Downloading and reviewing kitsune dataset which includes the following types of attacks:
   - Active Wiretap
   - ARP MitM
   - Fuzzing
   - Mirai
   - OS Scan
   - SSDP Flood
   - SSL Renegotiation
   - SYN DoS
   - Video Injection

2. Reviewing and building upon authors' published code in response to upcoming experiments:
   1) Testing and training the model:
      - Analysing by providing custom feature mapping input.
      - Analysing by using the feature mapper function.
      - Analysing by providing distinct number of maximum autoencoder size.

   2) Implementing additional functions to comply with future experiments:
      - getDistingDatasets()- function that takes a general dataset and refines them into malicious and benign according to the provided labels.
      - train ()- training the model on a given dataset.
      - test ()- testing the model on a given dataset and return the RMSE's values.
      - score ()- calculates the accuracy of our model.

C. *Research and investigate custom ensemble selection as a potential means to improve performance of the selected autoencoder-based intrusion detection method.*

1. Studying and surveying about dynamic ensemble selection:
   - Dynamic ensemble selection is an ensemble learning technique that automatically selects a subset of ensemble members just-in-time when making a prediction.
   - The method entails fitting several machine learning models to the training dataset, then choosing the models based on the characteristics of the example to be predicted which are expected to perform best when making a prediction for a specific new example.

2. Implementing custom ensemble selection to the code developed in previous task:
   - Deactivating 50% autoencoders: turning off every odd autoencoder.
   - Deactivating partial number of autoencoders (20%, 60%, 80%) in ensemble layer using custom lists.

3. Conducting experiments and collecting results (AUC scores, ROC Curves, Confusion matrices and Time graphs) for detecting different types of attacks.

D. Research and investigate about dynamic ensemble selection.

Reading and understanding a review paper [16] of various DES methods.

E. Compile the results and findings.

1. Analysis of the results by using the dynamic ensemble selection models of DESlib, performance enhancements, comparing strengths and weaknesses of different methods.
2. Preparing a manuscript based on the research findings for potential publication in a scholarly journal/conference.

IV. Experiments and Results

To achieve the deactivation of odd Autoencoders in the Ensemble Layer, a technique was developed where the ensemble of Autoencoders was selectively modified. In the traditional ensemble approach, all Autoencoders in the ensemble contribute equally to the final prediction. However, in this research, a modification was made to deactivate the odd-numbered Autoencoders in the ensemble.

By deactivating the odd Autoencoders, the ensemble's dynamics and behavior were altered. This modification aimed to explore the impact of selectively removing certain Autoencoders on the overall performance of the ensemble system. It allowed for an investigation into the potential benefits of reducing the number of active Autoencoders in the ensemble layer.

In addition to deactivating odd Autoencoders, another approach was explored in this study, which involved deactivating random Autoencoders for a given percentage. This technique aimed to introduce an element of randomness in the ensemble layer by deactivating a certain percentage of Autoencoders in a non-deterministic manner.

The deactivation of random Autoencoders was achieved by implementing a randomized algorithm that randomly selected Autoencoders for deactivation based on the given percentage. This approach allowed for a dynamic and variable composition of the ensemble, exploring the impact of different Autoencoder combinations on the overall performance of the system.

By deactivating random Autoencoders for a given percentage, the research project aimed to assess the effect of ensemble diversity and the role of individual Autoencoders in the overall ensemble's performance.

Overall, the research successfully achieved the deactivation of odd Autoencoders in the Ensemble Layer, as well as the deactivation of random Autoencoders for a given percentage. These approaches allowed for the exploration of different ensemble configurations and their impact on the performance and behavior of the ensemble system. Following are the results of the statistics which show the potential benefits of minimizing the ensemble layer of the NIDS.

A. Deactivating odd Autoencoders in the Ensemble Layer.

The threshold used to collect the confusion matrices was determined by Youden's J Statistic method [17].

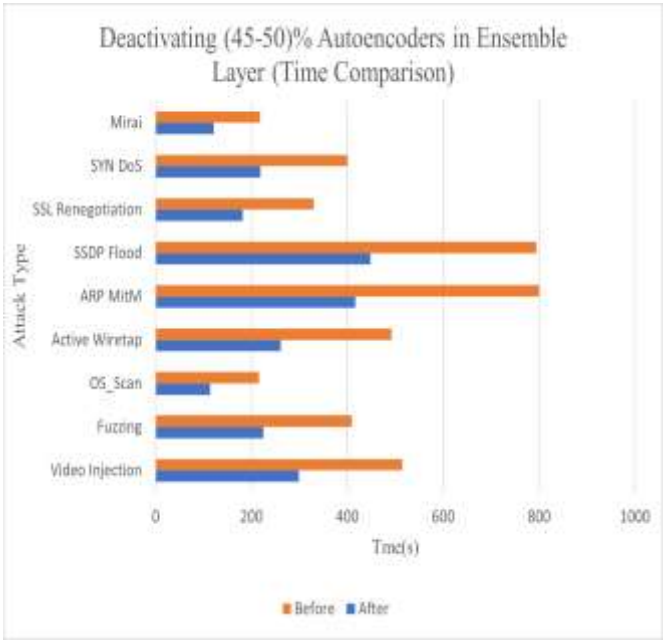

Fig. 1. Deactivating 45-50% Autoencoders (Time Comparison).

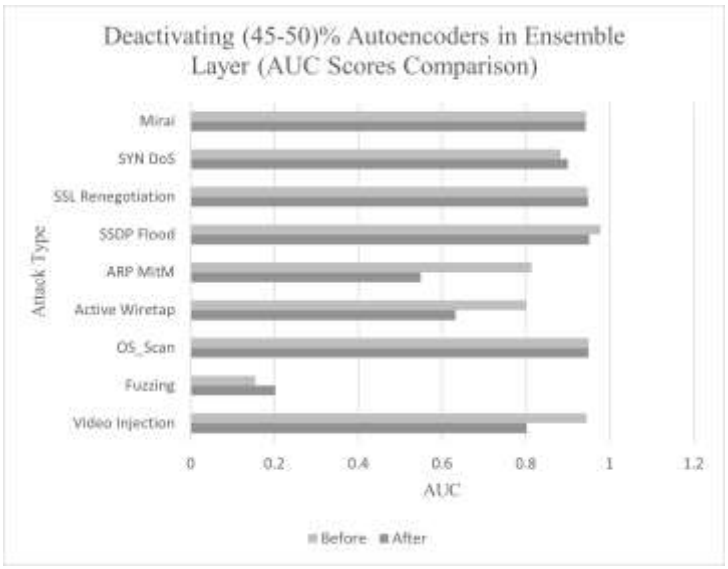

Fig. 2. Deactivating 45-50% Autoencoders (AUC Scores Comparison).

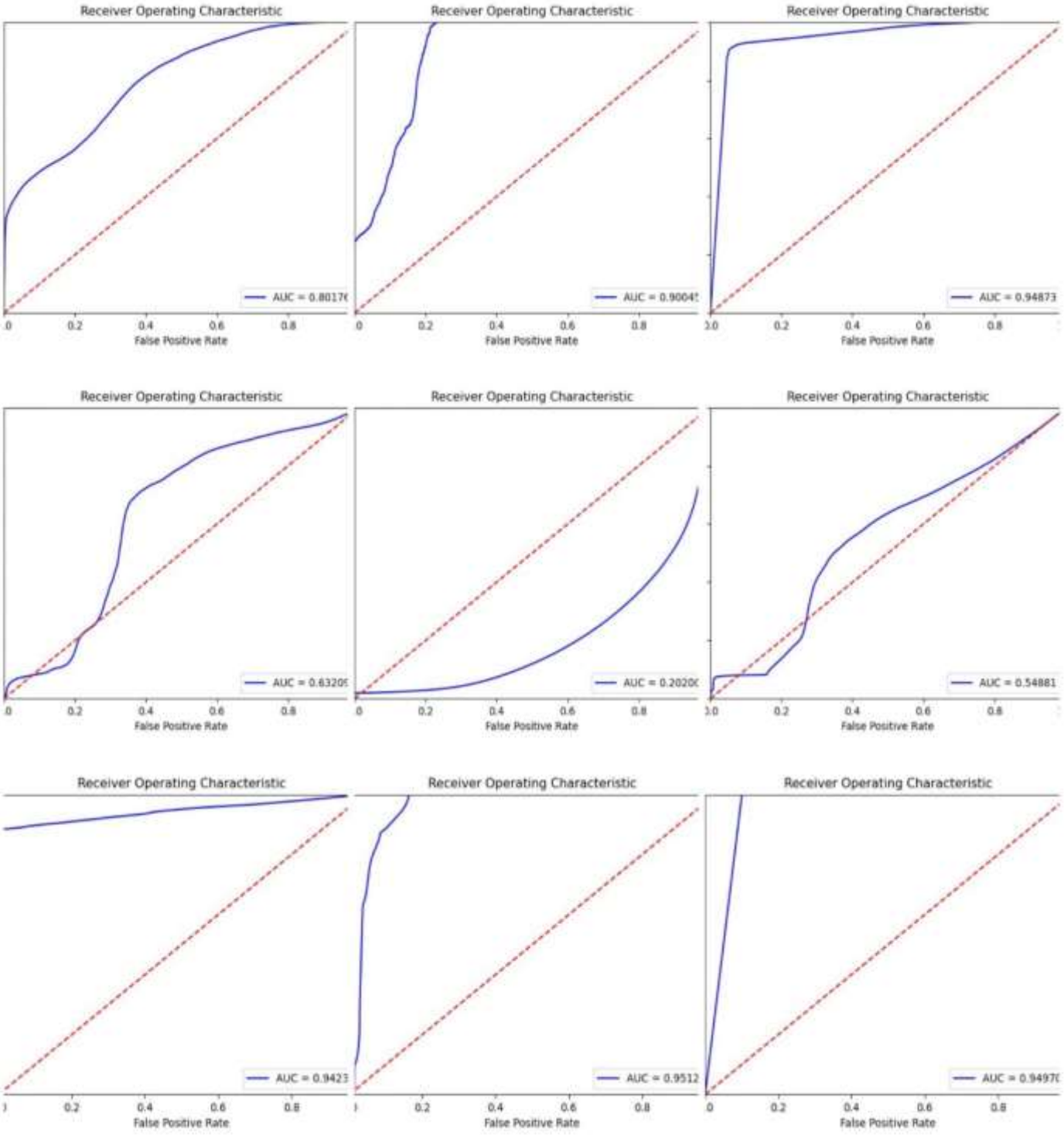

Fig. 3. ROC Curves for 9 different types of Attacks, after deactivating every odd autoencoder in ensemble layer.

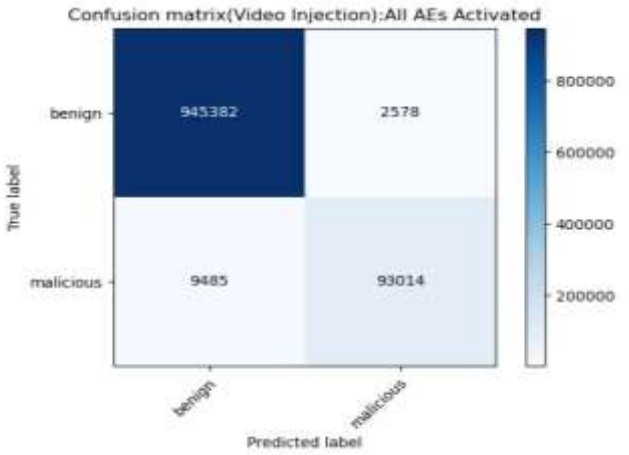

Fig. 4. Confusion Matrix for Video Injection Attacks (All AEs).

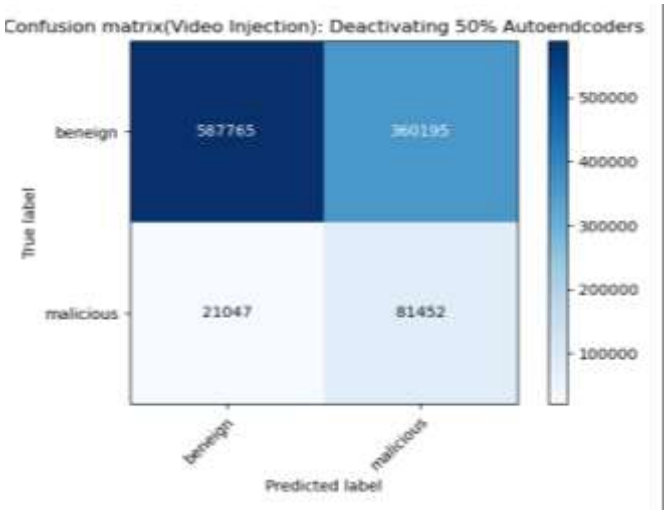

Fig. 5. Confusion Matrix for Video Injection Attacks (50 % AEs).

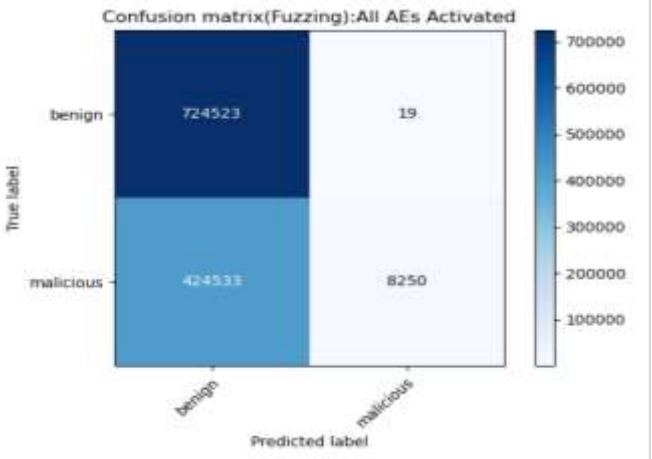

Fig. 6. Confusion Matrix for Fuzzing Attacks (All AEs).

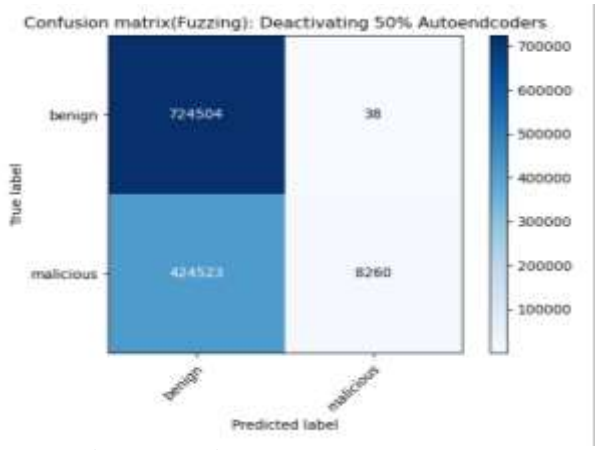

Fig. 7. Confusion Matrix for Fuzzing Attacks (50 % AEs).

B. Effect of Deactivation Rate Through Custom Deactivation.

1) Deactivating 80% Autoencoders

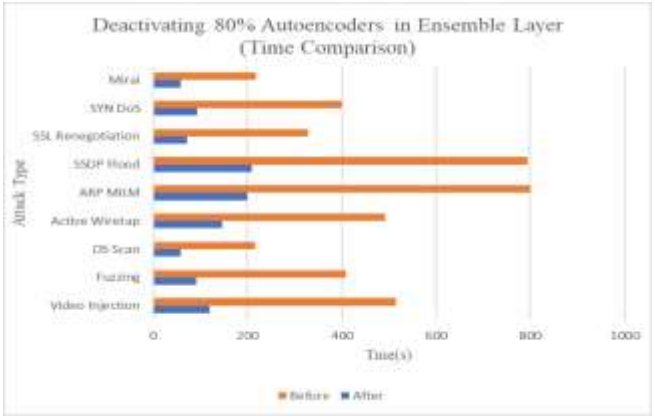

Fig. 8. Deactivating 80% AUs (Time Comparison).

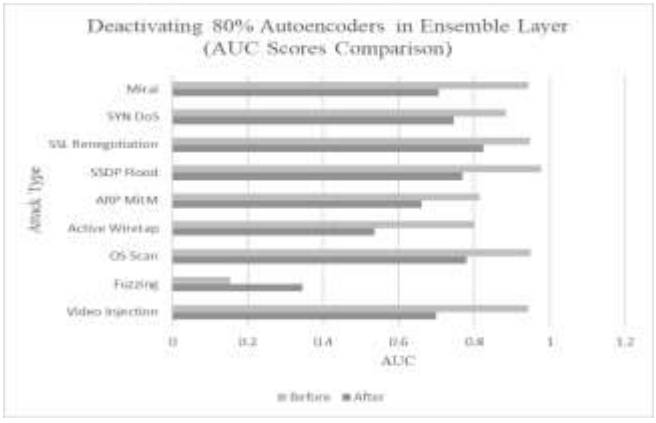

Fig. 9. Deactivating 80% AUs (AUC Scores Comparison).

2) Deactivating 40% Autoencoders

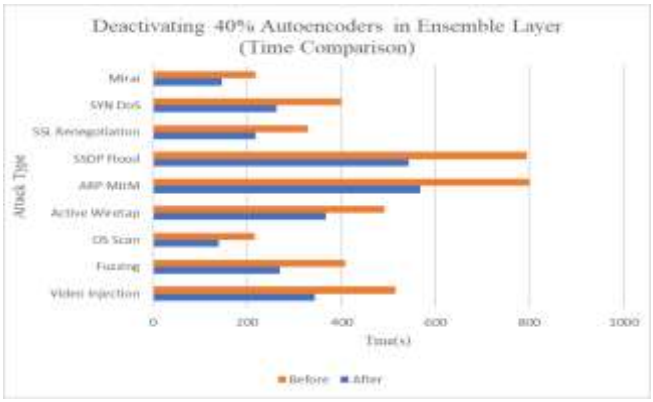

Fig. 10. Deactivating 40% AUs (Time Comparison).

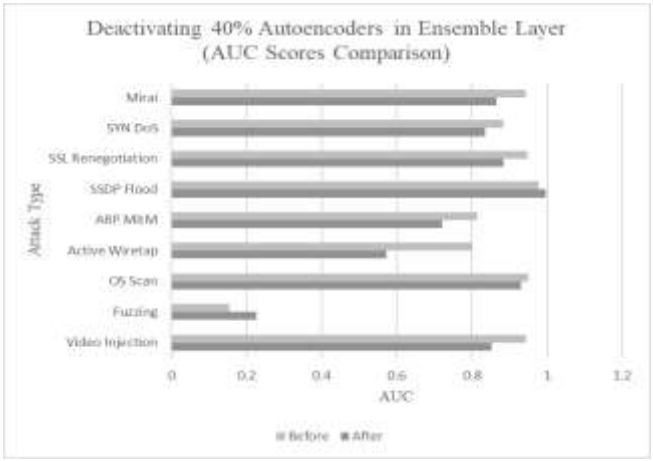

Fig. 11. Deactivating 40% AUs (AUC Scores Comparison).

3) Deactivating 20% Autoencoders

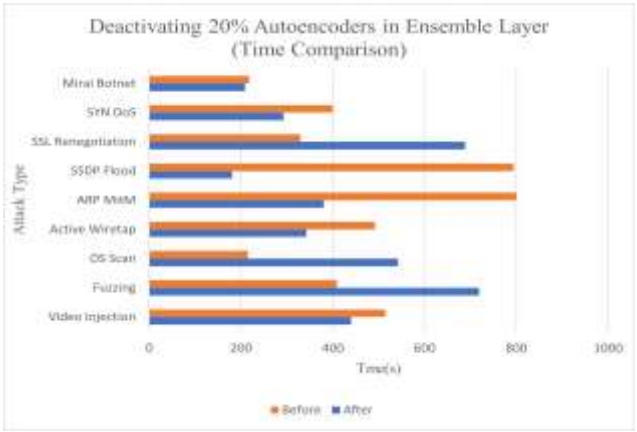

Fig. 12. Deactivating 20% AUs (Time Comparison).

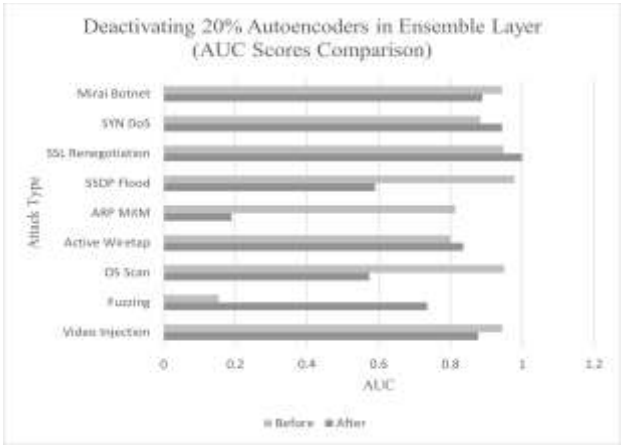

Fig. 13. Deactivating 20% AUs (AUC Scores Comparison).

The results of the experiments offer valuable insights into the optimization of anomaly detection systems, emphasizing the importance of balancing the number of active autoencoders with the desired processing time and accuracy.

It was observed that deactivating approximately half of the autoencoders in the ensemble layer of an anomaly detection system resulted in a significant decrease in both processing time and AUC (Area Under the Curve) scores. The findings indicate that the processing time is directly proportional to the number of active autoencoders in the ensemble layer. By deactivating half of the autoencoders, a 50% reduction in time was recorded, with a 10% decrease in the AUC scores. This suggests that there is a trade-off between processing efficiency and the accuracy of anomaly detection. Additionally, the limited impact on the AUC Scores suggests that deactivating autoencoders based on their competence level using techniques such as KNORAE [7], can help the system operate more efficiently. It is worth noting that this optimization has the potential to greatly improve the overall performance of the system and make it more scalable, especially in real-world applications where processing speed is crucial.

V. Future Works

In the next phase of this research, we plan to perform an analysis of the results by implementing the dynamic ensemble selection models of DESlib, with a focus on performance enhancements and a comparison of the strengths and weaknesses of different methods. This examination will offer insightful information regarding the practical use of these models. Furthermore, we aim to prepare a manuscript based on the research findings for potential publication in a scholarly journal or conference. This will allow us to disseminate our work to a wider audience and contribute to the advancement of the field.

VI. Conclusion

In conclusion, neural networks have emerged as a popular solution for network intrusion detection systems (NIDS) due to their ability to effectively differentiate between normal network traffic and malicious attacks. However, their resource-intensive nature poses challenges for implementation on network gateways and routers that may have limited memory and computing power. Furthermore, to enhance the anomaly detection rate of the NIDS system, a dynamic ensemble selection classifier (DES) has been suggested for embedding in the ensemble layer of the Anomaly detector. This integration of DES can significantly improve the system's ability to identify and respond to network attacks, further enhancing its effectiveness in real-time threat detection. By incorporating DES into the ensemble layer, the plug and play NIDS system can leverage the benefits of both ensemble learning and dynamic classifier selection, leading to improved accuracy and robustness in detecting network anomalies.

Overall, this research highlights a promising direction for enhancing the performance and efficiency of NIDS systems, making them more accessible and effective for deployment in resource-constrained network environments. The proposed approach holds the potential to provide reliable and autonomous network security, protecting local networks from evolving cyber threats without the need for manual intervention.

VII. Acknoledement

The author is grateful to the Student Success Centre of King Fahd University of Petroleum and Minerals for hosting the Uxplore research project, which provided undergraduate students with valuable research opportunities. Participating in the Uxplore project has been highly satisfying, allowing the author to delve into neural networks and develop crucial skills in this area. The program coordinators and mentors provided essential guidance and support. The author appreciates the assistance, feedback, and collaboration of all involved in the project. The author also acknowledges the support of Interdisciplinary research centre of Intelligent Secure System.

References

[1] Garcia-Teodoro et. al. Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1):18–28, 2009.

[2] Harjinder Kaur, Gurpreet Singh, and Jaspreet Minhas. A review of machine learning based anomaly detection techniques. arXiv preprint arXiv:1307.7286, 2013.

[3] Miao Xie, Jiankun Hu, Song Han, and Hsiao-Hwa Chen. Scalable hypergrid k-nn-based online anomaly detection in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 24(8):16611670, 2013.

[4] Srinivas Mukkamala, Andrew H Sung, and Ajith Abraham. Intrusion detection using an ensemble of intelligent paradigms. Journal of network and computer applications, 28(2):167–182, 2005.

[5] Xiangzeng Zhou, Lei Xie, Peng Zhang, and Yanning Zhang. An ensemble of deep neural networks for object tracking. In Image Processing (ICIP), 2014 IEEE International Conference on, pages 843–847. IEEE, 2014.

[6] Mahmood Yousefi-Azar, Vijay Varadharajan, Len Hamey, and Uday Tupakula. Autoencoder-based feature learning for cyber security applications. In Neural Networks (IJCNN), 2017 International Joint Conference on, pages 3854–3861. IEEE, 2017.

[7] hmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[8] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," Computers and Electrical Engineering, vol. 99, p. 107810, Apr. 2022, doi:

[9] Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Apr. 2018.

[10] M. Putchala, "Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network using Gated Recurrent Neural Networks (GRU)," Browse all Theses and Dissertations, Jan. 2017.

[11] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things, IEEE Access 5 (Sep. 2017).

[12] A. Azmoodeh, A. Dehghantanha, K.R. Choo, Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigenspace Learning, IEEE Transactions on Sustainable Computing 4 (1) (Jan. 2019).

[13] A. Abeshu, N. Chilamkurti, Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing, IEEE Communications Magazine 56 (2) (Feb. 2018).

[14] F. Qu, J. Zhang, Z. Shao, S. Qi, An Intrusion Detection Model Based on Deep Belief Network, in: Proceedings of the 2017 VI International Conference on Network, Communication and Computing - ICNCC 2017, Kunming, China, Dec. 2017.

[15] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici and Asaf Shabtai. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection, NDSS '18, 18-21 February 2018, San Diego, CA, USA Copyright 2018 Internet Society, ISBN 1-1891562-49-5

[16] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," Information Fusion, vol. 41, pp. 195–216, May 2018,

[17] Bartlett, P. L., Montanari, A., & Rakhlin, A. Deep learning: A statistical viewpoint 2021.