VOL. 09 NO. 9,
AUGUST, 2025
E-ISSN 3026-958X
P-ISSN 3027-1169

African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Science Innovation & Technology Research (JSITR)

A Recurrent Neural Network Approach for Detecting and Mitigating Denial of Service (DoS) and Man in the Middle (MitM) Attacks in Cyberspace

Sajo, A.; & Sarjiyus, O.
Department of Computer Science, Adamawa State University Mubi, Nigeria
Corresponding Author: sajo1004@adsu.edu.ng
DOI: https://doi.org/10.70382/ajsitr.v9i9.033

Abstract

The hazards presented by sophisticated cyber threats, especially Denial of Service (DoS) and Man in the Middle (MitM) assaults, which continue to be among the most disruptive to network stability, confidentiality, and integrity, have increased due to the rapid rise of cyberspace. Due to their heavy reliance on static or signature-based machine learning techniques, traditional intrusion detection systems (IDS) frequently fall short in terms of accuracy and real-time flexibility to changing attack patterns. This study offers a Modified Recurrent Neural Network (MRNN) framework intended to more effectively identify and mitigate DoS and MitM attacks in order to overcome these drawbacks. To capture the temporal and sequential properties of network traffic, the study uses benchmark datasets like CICIDS-2017 and CSE-CIC-IDS-2018 and applies stringent pre-processing and feature extraction techniques. In order to improve scalability, decrease false positives, and increase classification accuracy, the MRNN model incorporates optimal parameters and architectural improvements over traditional RNN, LSTM, and GRU structures. The system's ability to discriminate between malicious and valid traffic in real time and to initiate automated mitigation responses, such as IP blocking and session termination, is demonstrated through experimental evaluation in a simulated network environment. According to the results, the suggested RNN performs better than baseline detection models in terms of robustness against encrypted or high-volume communication and reaction time. It also performs best in terms of high Precision=0.997, Recall=0.985, F1=0.991, Support=332), MitM (Precision=0.982, Recall=0.997, F1=0.990, Support=335 and Normal Precision=0.997, Recall=0.994, F1=0.995, Support=333. These numerical values indicate balanced, near-optimal performance across the three classes with only tiny class-specific differences; in particular, MitM

shows slightly higher recall but slightly lower precision relative to DoS, indicating a small trade-off between missed attacks and false alarms. In addition to its technical contributions, the study provides enterprises, network administrators, and legislators with useful information for strengthening Cyber Security infrastructures. This study adds to the growing body of knowledge on intrusion detection by developing AI-driven security mechanisms and offers a scalable way to counteract two of the most harmful and persistent assault vectors in cyberspace.

Keywords: Detection System, DoS Attack, MitM Attack, Deep Learning, RNN Model.

Introduction

The threat of cyber-attacks has increased to previously unheard-of heights in the quickly changing digital ecosystem, posing enormous difficulties to individuals, governments, and enterprises. The Denial-of-Service (DoS) assault is a type of cyber-attack that is especially worrisome since it attempts to prevent or interfere with a target system's capacity to offer resources or services to authorized users. Man in the Middle (MitM) attack is simply a type of cyber-attack in which an intruder intercepts and alters communication between two parties, with the intention to steal sensitive information or conversations. In this case, the intruder positions himself between the victim and the intended recipient, pretending to be the legitimate party (Thakur & Kumar, 2021).

DoS attacks usually cause a target system to become unresponsive or unavailable by flooding it with too many requests or data (Singh & Anand, 2021). Conversely, MitM attacks entail the placement of an attacker in the path of two parties in communication with the intention of intercepting, monitoring, and maybe altering sent data (Thakur & Kumar, 2021). Because these assaults can imitate real-world traffic patterns, identifying and mitigating them can be difficult, necessitating the development of sophisticated techniques.

For processing sequential data and identifying patterns within it, recurrent neural networks (RNNs) have become highly effective artificial intelligence technologies (Alzubaidi et al., 2022). They are perfect for examining network traffic and seeing any irregularities that might be signs of DoS or MitM attacks because of these capabilities. An efficient detection and mitigation system can be developed by using an RNN trained on network traffic data, which teaches the model to distinguish between patterns of regular traffic and attack traffic (Pashayev & Iqbal, 2022).

With emphasis on recent developments from 2021 to 2024, presented an approach in this research to train an RNN model on a dataset that includes both regular and attack network traffic. DoS attacks usually cause a target system to become unavailable or unresponsive by flooding it with requests or data (Kim et al., 2021). In contrast, MitM attacks entail the placement of an attacker in the path of two parties in communication with the intention of intercepting and maybe

altering the data being communicated (Algarni & Malaiya, 2022). Because these assaults might mirror real-world traffic patterns, it can be difficult to identify them, which emphasizes the need for advanced detection and mitigation techniques.

With emphasis on recent developments from 2021 to 2024, presented an approach in this research to train an RNN model on a dataset that includes both regular and attack network traffic. DoS attacks usually cause a target system to become unavailable or unresponsive by flooding it with requests or data (Kim et al., 2021). In contrast, MitM attacks entail the placement of an attacker in the path of two parties in communication with the intention of intercepting and maybe altering the data being communicated (Algarni & Malaiya, 2022). Because these assaults might mirror real-world traffic patterns, it can be difficult to identify them, which emphasizes the need for advanced detection and mitigation techniques.

Recurrent neural networks (RNNs) are strong artificial intelligence instruments that perform exceptionally well in pattern recognition and sequential data processing (Liu et al., 2023). Because of these features, RNNs are perfect for examining network traffic and spotting irregularities that could be signs of DoS or MitM attacks. An efficient detection and mitigation system can be achieved by training an RNN model on network traffic data, which enables the model to distinguish between patterns of normal traffic and attack activity (Alheeti et al., 2022).

This research describes a technique for using a dataset containing both legitimate and malicious network traffic to train an RNN model. To improve online system security, the model will be adjusted to precisely detect DoS assaults, with an emphasis to MitM attacks. Through the use of sequential data processing power of RNNs, the initiative seeks to aid in the creation of cutting-edge cyber security defences against these damaging assaults.

The rising of cyber threats, particularly Denial-of-Service (DoS) and Man-in-the-Middle (MitM) attacks, has exposed the limitations of conventional intrusion detection frameworks. Many existing security solutions still rely on signature-based and static machine learning approaches, which fail to adapt to evolving network behaviors and generate high false-positive rates (Sharafaldin *et al*. 2021; Alshammari and Zincir-Heywood, 2022). While recurrent neural networks (RNNs), including Long Short Term Memory (LSTM) and GRU, in this research have demonstrated strong learning capabilities in terms of addressing time latency, precision, accuracy by building a modified RNN capable of mitigating DoS and MitM attacks.

Literature Review

Over the past few decades, a great deal of research and development has gone into the topic of intrusion detection. The initial research in this field concentrated on signature-based detection, which matched patterns of known network traffic attacks. This method has drawbacks, too, in that it is unable to identify polymorphic or zero-day attacks that do not correspond with known signatures. Researchers have created a number of more advanced methods for intrusion detection in response to these restrictions, including as anomaly detection, machine learning, and network traffic analysis.

A key method for identifying and thwarting DoS and MitM attacks is anomaly detection. Security analysts can discover and stop these attacks before they cause significant harm by using anomaly detection to find strange traffic patterns or unexpected behaviour in network data. Singh & Behal (2020). In the authors view, the methods used for detecting anomaly cannot be 100% efficient, since attackers' device multiple ways to flood in assaults on the system network.

Thus, this section entails to analyse and combine the available research on this subject to present a thorough picture of the field's current status. In addition to identifying gaps and possibilities for additional study, this review would probably include an overview of prior studies on RNN-Based Anomaly Detection and their efficacy in identifying and thwarting these kinds of attacks. It would offer a critical analysis of the RNN-Based Anomaly Detection research that has already been done, pointing out areas in which additional study is required and assessing the advantages and disadvantages of the current strategies. For instance, the evaluation might assess the scalability and accuracy of RNN-Based anomaly detection techniques and contrast them with alternative strategies like deep learning or machine learning. It might also look into how well these techniques operate in various network environments, such on-premises or cloud-based systems, for identifying and thwarting DoS and MitM assaults. Kim et al. (2021).

Types of Man in the Middle (MitM)

One of the well-known computer security threats that has drawn attention from numerous researchers is the MITM attack. The authors of the paper have investigated MITM attacks on networks that are physically connected. This paper's result categorizes attackers into four types: state, vulnerability, behaviour, and target. This creates a taxonomy of MITM on https. This line of reasoning has led to the exploration of a mathematical model on MITM attacks with secure socket layer protocols. Numerous studies on MITM attacks that take into account UMTS and OSI layers have also been presented in addition to the study effort. Additionally, has been discussed on the impact of MITM attack in relation to the MAC layer for ad hoc mesh topology. Fahad *et al*. (2018).

Delayed Message in MitM

The applications like VANET (Vehicular network), banking transactions and battlefield are severely affected with MITM with message delay due to sensitive nature of messages delay due to sensitive nature of messages it can create. For instance, consider a battlefield scenario where delay of single message by a malicious node can cause catastrophic situations where legitimate nodes are unable to receive the message on time. As a result, such situations can also put human life in danger. Ahmed *et al* (2018).

Dropped Message in MitM

During dropped message attack, the attacker intentionally drops the authentic message by supressing the further propagation of PG. As a consequence, the attacker obstructs the authentic user (AU) from receiving message related to safety and non-safety measures. For example, during VANET when a vehicle broadcasts legitimate information about road conditions, and if that information gets dropped then human life will be in danger. Ahmed *et al* (2018).

Tempered Message in MitM

In this attack, the attacker has targeted the content of the received message. Whenever the intruder node receives the content, the attacker manipulates the sensitive information. For instance, in banking transactions the broadcasting of messages during any serious declaration affects seriously to society act. In this case, the tampering of sensitive content can lead to serious consequences. This intruder message is misleading the legitimate nodes and creates disaster in the network. Further every message of WSN contains three important components i.e., data, time and location. This kind of attack enables the attacker to manipulate the data like location timings and contents. Javeed *et al.* (2020).

Related Works

Kim *et al*. (2020) proposed a model based on a Convolutional Neural Network (CNN) for detecting DoS attacks using the KDD-99 dataset and the CICIDS2018 dataset. In the experiment they converted the input features into an "image", the proposed CNN model then receives a grey-scale or RGB image as input. Different number of layers was explored as they performed both binary and multiclass classification. The proposed model was then evaluated against a Recurrent Neural Network (RNN) model with the proposed models performing better in both binary and multiclass classification. Their method achieved an accuracy of over 99% on both classification types.

Sabeel *et al.* (2019) proposed DNN and LSTM models for binary prediction of unknown DoS and DDoS attacks. These models were trained on the CICIDS2017 dataset. The authors then generated a new test dataset, ANTS2019, in a simulated environment to measure performance of their proposed models. Their proposed DNN method was able to achieve an accuracy of 99.68% when it was trained on CICIDS2017 and part of ANTS2019 datasets.

Wu et al. (2019) proposed a hierarchical CNN + RNN neural network which they called LuNet. It consist of multiple levels of CNN and RNN where both network learns jointly from their input data. Their proposed model was tested on the NSL-KDD and UNSWNB15 datasets. They carried out binary and multiclass classification and achieved a maximum accuracy of 99.36% and 99.05% respectively. Both results were on the NSL-KDD dataset.

Almomani *et al.* (2018) used eight different machine learning models in detecting DoS attacks which are: Naïve Bayes (NB), Decision Trees (DT), Random Forests (RF), Support Vector Machine (SVM), J48, Artificial Neural Networks (RNN), K-Nearest Neighbour (KNN) and

Bayesian Networks (BN). They used the WSN-DS dataset for their experiment and performed feature selection based on expert survey. The authors reported that the Random Forest algorithm achieved the best results with a true positive of 98.3%.

Vinayakumar *et al*. (2019) proposed a scalable and hybrid DNN framework called Scale-Hybrid- IDS-AlertNet, which can effectively monitor network traffic and host-level events in real-time to proactively alert for possible cyber-attacks. The authors turned the model on the KDD-99 dataset and applied it to other datasets such as NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS2017 as benchmark. For the WSN-DS dataset, they achieved accuracy of 99.2 and 98.0% for binary and multiclass classification respectively.

Park et al. 2018, proposed a Random Forest (RF) classifier to detect the type of DoS attacks in the WSN-DS dataset. The proposed model achieved a best F1- Score of 99%, 96%, 98%, 100% and 96% for Blackhole, Flooding, Grayhole, Normal, and Scheduling (TDMA) attacks respectively. They achieved an overall accuracy of 97.8%.

Abdullah *et al*. (2018) proposed used several ML classifiers for detecting intrusions in WSNs. These classifiers are SVM, Naïve Bayes, Decision Tree, and Random Forest. They used the WSN-DS dataset for training and the WEKA data mining tool for implementing their classifiers. The SVM classifier achieved the highest accuracy of 96.7% compared to the other classifiers.

Premkumar and Sundararajan (2020) Presented a Deep Learning-based Defence Mechanism (DLDM) to identify and isolate DoS assaults in the Data for-warding phase (DFP). DoS assaults such as fatigue, jamming, homing, and flooding may now be detected more reliably thanks to a novel methodology described in research. It is more resistant to denial of service (DoS) assaults because we do extensive simulation studies to separate the enemies adequately. Their system's detection, throughput, packet delivery ratio, and accuracy in the simulation are all high. It also cuts down on wasted energy and the number of false alarms. The complexity and frequency of Distributed Denial of Service (DDoS) attacks on web based services have grown tremendously with the emergence of modern wireless technology and current computing paradigms. As a result, it is critical to identify these attacks in the sea of data packets.

Asad *et al*. (2020) provided a unique deep neural network detection technique for reliably detecting numerous application layer DDoS assaults in research using feed-forward back-propagation. On a state-of-the-art dataset compassing several types of DDoS assaults, the neural network architecture suggested here can detect and utilize the essential high level aspects of packet flows with a precision of 98. The primary threat to the WSN is posed by the fact that the nodes in the network broadcast their signals. As a result, the security of WSNs is an essential task that must be completed. As a result, to overcome these challenges or hazards, we are attempting to identify them utilizing artificial intelligence technologies. In order to categorize different sorts of assaults, using Machine Learning and Deep Learning, which are emerging domains, we may use a wide range of algorithms. Once we have identified the assault correctly, we may take the necessary steps to avoid it. We are making use of WSN-

DS. It has four types of assaults: Grayhole, Blackhole, TDMA (Scheduling), and floading, all of which fall under Denial of Service Attacks.

Loukas *et al.* (2017) used LSTMs achieved 86.9% accuracy covers all attacks types, including DDoS, command injection, and network malware. This accuracy better than what other standard machine learning methods have achieved. They also tested LSTM Out-performs other Attacks against untrained malware attacks again machine learning methods.

Shaban *et al*. (2019) recommended a CNN model to detect DDoS attacks. The authors compared their proposed model with classification injection, and network algorithms KNN, DT, SVM, NN in more than two datasets: (simulated network traffic) and (NSL-KDD) has been observed. The proposed model compares well well with this model. The other four classification algorithms, such as KNN, DT, SVM, and NN with 99% accuracy two records. In this method, a single column is populated used to convert data into matrix form. Therefore, it affects the learning of the model.

(Wazirali and Ahmad 2021) evaluated the effectiveness of machine learning classification algorithms in detecting (1) flooding, (2) grey hole, and (3) black hole distributed denial of service attacks in wireless sensor networks. We conducted our review using a WSN-based dataset, referred to as WSN-DS, and took the accuracy and speediness measures into account. The result show that the J48 approach is the most accurate and fastest way for identifying grey hole and black hole attacks. At the same time, the Random Tree method is the most accurate and fastest method for detecting flooding assaults. The J48 approach is the most efficient for speed, requiring an average of 0.54 s of processing time per sample.

Dong *et al.* (2019) did a short survey on DDoS and its effects on the Cloud environment. The effect of DDoS on Cloud could lead to other services that might be using the same hardware. Furthermore, they discuss the increase in bill (EDoS, though the name not used) due to a DDoS attack on the Cloud. They also discuss the VM Neighbor attack (the problem of co-residency) that could arise out of DDoS.

Gupta *et al.* (2017) broke down the different types of DDoS attacks into application layer attacks, protocol attacks and volume-based attacks. They focused on the attacks in the spectrum of Cloud Computing and looked into the common tools that are readily available to perform a DoS or DDoS attack. There are various solutions being provided in the area to reduce the impact of these attacks if not to stop it completely.

Mironeanu *et al.* (2021) proposed a model called as Selective Cloud Egress Filtering (SCEF) where a specified modules are deployed to deal with attacks. Their model focuses on the creation of a Virtual Machine Monitor (VMM) to collect the packet metadata from the VMs and then passing it on to SCEF. The SCEF filters the packets based on attack parameters and helps to stop that particular VM being used as a part of the attack. The model is in its early stages. They have performed a simulation test of their model. However, the model has only focused on ICMP Flood Attacks, DNS Reflection Attacks, SSH Brute Force Attacks and TCP SYN Attacks. The parameters mentioned are not enough to identify more complex attacks using the same protocols that they have added to SCEF. For instance, the parameter to detect

DNS Reflection Attack is a sharp drop in the inbound//outbound DNS packet ratio. While this is in theory can be a good identification of DNS Reflection Attacks, there are other variables like the packet size (query-to-response size), and the structure of the DNS query including parameters like the type, truncation, recursion desired and many more that affect the attack.

Mittal *et al.* (2023) proposed an efficient DL-based DDoS attack detection framework in 5G and B5G environments. The proposed framework is developed by concatenating two differently designed DNN models, coupled with a feature extraction algorithm, i.e. PCC. It is built to detect the DDoS attacks and the type of DDoS attacks encountered. The authors evaluated the proposed framework using four different scenarios over an industry-recognized dataset (i.e. CICDDoS2019 dataset). Results illustrated that the framework could detect DDoS attacks with an accuracy of 99.66% and a loss of 0.011. Furthermore, the proposed detection framework results were compared with the existing approaches, i.e. KNN, SVM, Deep Defense, and CNN ensemble. The proposed framework outperformed all except the CNN ensemble. The CNN ensemble has better precision and recall than the proposed framework. The proposed model has a complex structure so it can take more detection time and thus can affect the model's performance in a real-time scenario.

Sbai and El Boukhari (2020) proposed a DL model DNN (withtwohiddenlayersand6-epochs) to detect data flooding or UDP flooding attack in MANETs, by usingthedatasetCICDDoS2019. The authors trained and evaluated the model with the CICDDoS2019 dataset. The proposed model obtained results that are: Recall: 1, precision: 0.99, F1-score: 0.99, Accuracy: 0.99, which are very promising. In this article, the authors have worked only on the data flood in go UDP flooding attack of the CICDDoS2019 dataset.

Mittal *et al.* (2023) proposed an efficient DL-based DDoS attack detection framework in 5G and B5G environments. The proposed framework is developed by concatenating two differently designed DNN models, coupled with a feature extraction algorithm, i.e. PCC. It is built to detect the DDoS attacks and the type of DDoS attacks encountered. The authors evaluated the proposed framework using four different scenarios over an industry-recognized dataset (i.e. CICDDoS2019 dataset). Results illustrated that the framework could detect DDoS attacks with an accuracy of 99.66% and a loss of 0.011. Furthermore, the proposed detection framework results were compared with the existing approaches, i.e. KNN, SVM, Deep Defence, and CNN ensemble. The proposed framework outperformed all except the CNN ensemble. The CNN ensemble has better precision and recall than the proposed framework. The proposed model has a complex structure so it can take more detection time and thus can affect the model's performance in a real-time scenario.

Alghazzawi *et al.* (2021) proposed the DL model that contains both feature extraction as well as classification processes in its structure. The DNN model consists of an input layer with 69 units, three hidden layers consist of an equal amount of 50 units and two units are used in the output layer. The authors have divided the dataset CICDDoS2019 into two datasets, i.e. Dataset1andDataset2.Dataset1iscategorizedastwo types of traffic: normal and attack traffic. Dataset2 is created to define the types of DDoS attacks. DNN model has nearly 100% accuracy

for DDoS attack detection on Dataset1 and thus the DNN model has achieved the reliable result for early action, suitable for real time scenarios. Also, it successfully classifies DDoS attacks with approximately 95% of accuracy on the Dataset2. The proposed model gives less accuracy in the case of multiclass classification.

Chen *et al.* (2019) proposed a DAD-MCNN (i.e. multichannel CNN) framework to detect DDoS attacks. The number of feature groups decides the number of channels. The authors have split the features into different levels, like packet level, host level, and traffic level. The authors have used the incremental training approach to train MC-CNN. The authors have conducted a sequence of tests over KDDCUP99, CICIDS2017datasetsforbinaryclassificationinboth datasetsandmulticlasscategoryinKDDCUP99only. They also compared MC-CNN with CNN, LSTM (3 layers), and other shallow ML methods (RF, SVM, C4.5, and KNN). The results showed that MC-CNN outperformed the state-of-art methods for all binary and multiclass classification. Further, the authors have also changed the training dataset size and evaluated the CNN and MC-CNN. The results showed that MC-CNN is better in the restricted data set and helpful in building DDoS detection systems when the training data are relatively insufficient. There is no much difference in the results of multichannel and single channel models. Also, the multichannel models will increase the complexity and thus might not be suitable when validated over real-time scenarios.

Cil *et al.* (2021) using the CNN model proposed how to detect DDoS attacks. The Authors compared their proposed model with the classification algorithms like DT, SVM, KNN, and NN over two datasets, i.e. dataset 1 (simulated network traffic) and dataset2 (NSL-KDD). It has been observed that the proposed model performed well compared to the other four classification algorithms such as like DT, SVM, KNN, and NN and gives an accuracy of 99%on both datasets. In this approach one-column padding has been used to convert the data into matrix form. Thus, it can affect the learning of the model.

Haider (2020) proposed a deep CNN framework for the detection of DDoS attacks in Software Defined Networks, and this proposed ensemble mechanism has been evaluated over the CICIDS2017 dataset. This solution is compared with the state-of-the-art DL-based ensembles and hybrid approaches (i.e. RNN, LSTM, RL). The ensemble CNN performed better than other three proposed DL-approaches, but there is a trade-off between their training and testing time. The authors have also compared the proposed ensemble CNN approach with existing competing approaches. The results showed that the ensemble CNN approach outperformed the existing competing approaches. The ensemble CNN has achieved an accuracy of 99.45%. This approach has training and testing times higher than other approaches. Thus, it can affect the mitigation mechanism. Therefore, attacks can cause more damage.

Wang and Liu (2020) proposed an information entropy and DL method to detect DDoS attacks in SDN environment. Thus, the technique uses two level detection for the identification of the attacks. Firstly, the controller will inspect the suspicious traffic through information entropy detection. A CNN model will then execute the detection based on the fine-grained packet to

distinguish among normal traffic and attack traffic. The author shave compared their method with the DNN, SVM, and DT. The CNN achieved higher precision, accuracy, F1-score, and recall among them. The accuracy of it is 98.98%. 1.
The ROC curve of CNN is steeper than DNNs, SVM, and DT. The AUC of CNN is 0.949. There is a need to set the threshold value for the detection method based on information entropy.
Kim *et al*. (2020) developed a CNN-based model to detect DoS attacks using the records of DoS attacks in CSE-CIC-IDS2018andKDDdatasets.Authorshave designed their CNN model considering the number of CLs and kernel size. They evaluated their model by creating 18 scenarios considering hyper parameters, the type of image, i.e. greyscale or RGB, the number of CLs, and the kernel size. The authors have evaluated each scenario for both binary and multiclass classifications. They then suggested optimal scenarios with higher performance. The CNN model is also compared with RNN. The CNN model can identify specific DoS attacks with alike characteristics compared to the RNN model. It has also been found that kernel size in CNN has not significantly impacted both binary and multiclass classification. The pre-processing time of conversion of features to RGB and greyscale images has not been considered, as it matters in real-time validation.
Doriguzzi-Corin *et al.* (2020) has been used to detect DDoS attacks, which helps in, lightweight execution with low processing overhead and detection time. Their unique traffic pre-processing mechanism is designed to feed the CNN model with network traffic for online DDoS attack detection. The authors compared LUCID with Deep Defence 3LSTM over ISCX2012, CIC2017, CSECIC2018, UNB201X and got comparable results. However, the LUCID outperforms 3LSTM in detection time. The performance of LUCID has been compared against state-of-the-art works (Deep Defence, TRIDS, E3ML) and validated on ISCX2012. Also, compared the LUCID with state-of-the-art works (Deep-
GFL, MLP, LSTM, 1D-CNN, 1D-CNN+LSTM) and validated on CIC2017 Dataset. The evaluation results show that the LUCID matches the existing state-of-the-art performance. It has also been demonstrated the suitability of the model in resource-constrained environments. Their work has also proved that LUCID is learning the correct domain information by calculating each feature's kernel activations. The LUCID training time on the GPU developmentboardis40timesfasterthantheauthors' implementation of Deep Defense 3LSTM. The feasibility test has also been done for the proposed approach. The padding has been used for making the size of each flow equal to n. By using padding, the CNN may get affected in learning the patterns. Also, there are trade-offs between accuracy and memory requirements. The pre-processing time has not been calculated as it is important for real-time scenarios.
Assis *et al.* (2021) the authors have proposed an SDN defence system. The defence system detects and mitigates DDoS attacks over the external targeted server and on the controller. The detection module detects attacks. In this module, the authors have used DL-based CNN method to detect DDoS attacks by inspecting the SDN traffic behaviour. The proposed method works in near real-time, as in this study, IP flow data have been extracted and analysed in one-

second intervals to reduce the DDoS effect over genuine users. The proposed CNN approach within the detection module has been compared with the other three anomaly detection approaches, i.e. the LR, the MLP network, and the Dense MLP. The authors have tested the above detection methods over two test scenarios, i.e. the first one uses simulated SDN data, and the second one uses CICDDoS2019 dataset. The overall results showed that CNN is efficient in detecting DDoS attacks for all these test scenarios. A GT-based technique has been applied in the SDN controller to mitigate the attack in the mitigation module. The outcomes showed that the mitigation method efficiently restores the SDN's regular operation. The proposed system operates autonomously to allow the speed of the detection and mitigation processes. The model shows less accuracy for CICDDoS 2019 dataset.

Hussain *et al.* (2020) have proposed a method to transform the non-image network traffic into three channel image forms. It has been evaluated on the existing ResNet-18 model, a state-of-the-art CNN model, for detecting the recent DoS and DDoS attacks. The proposed method used the cleaned and normalized features to transform the data into images without using any encoding or transformation techniques. The authors also compared the proposed methodology using ResNet-18 with a state of art solution and outperformed it on the same dataset. The proposed methodology using ResNet-18 achieved 99.99% accuracy in binary class classification. It has also achieved an accuracy of 87.06% for the 11 types of DoS and DDoS attacks on the CICDDoS2019 dataset. The pre-processing time is not calculated for converting non-image data to image data as this is the important metric for real-time validation. Also, the transformation of the original 60*60*3 dimensions into 224*224*3 dimensions has not been described for the input to the ResNet model.

Li *et al.* (2020) proposed a deep learning model to detect DDoS attacks in SDN environment. The model comprises an input, forward recursive, reverse recursive, FC hidden layer, and output layers. RNN, LSTM, and CNN are also used in the model. Thus, the authors have formed four different models that are: LSTM, CNN/LSTM, GRU, 3LSTM. The accuracy of the DDoS attack by the use of the ISCX dataset is 98%. The DDoS attack detection and defence system are built using the ubuntu14. 04 operating system, and the DDoS defence system is verified through real-time DDoS attacks. But tested on only limited types of real-time DDoS attacks that are the Ping Of Death attack, ARP flood inundation attack, SYN flood inundation attack, Smurf attack, and UDP flood inundation attack.

Sajo and Sarjiyus (2025) in their study posited that DoS attacks usually cause a target system to become unresponsive or unavailable by flooding it with too many requests or data Conversely, MitM attacks entail the placement of an attacker in the path of two parties in communication with the intention of intercepting, monitoring, and maybe altering sent data. Because these assaults can imitate real-world traffic patterns, identifying and mitigating them can be difficult, necessitating the development of sophisticated techniques.

What RNN (Recurrent Neural Network) is

A recurrent neural network (RNN) is a deep learning model that is trained to process and convert a sequential data input into a specific sequential data output. Sequential data is data—such as words, sentences, or time-series data—where sequential components interrelate based on complex semantics and syntax rules. An RNN is a software system that consists of many interconnected components mimicking how humans perform sequential data conversions, such as translating text from one language to another. RNNs are largely being replaced by transformer-based artificial intelligence (AI) and large language models (LLM), which are much more efficient in sequential data processing.

How a recurrent neural network work

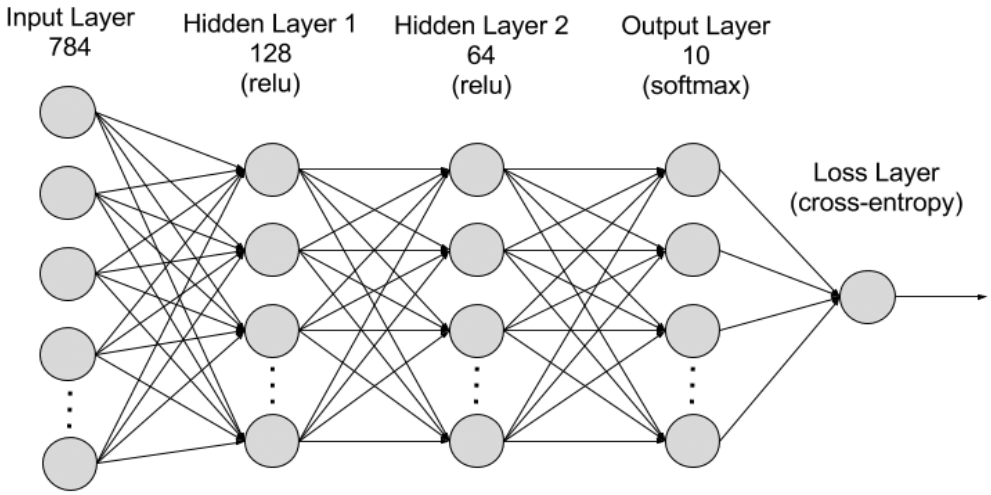


Figure 3.1: The Structure of an RNN.

RNNs are made of neurons: data-processing nodes that work together to perform complex tasks. The neurons are organized as input, output, and hidden layers. The input layer receives the information to process, and the output layer provides the result. Data processing, analysis, and prediction take place in the hidden layer.

The Structure of an RNN

The provided diagram above, illustrates a basic Recurrent Neural Network, showing its three primary components:

1. Input Layer
2. Hidden/Recurrent Layer
3. Output Layer

1. Input Layer

The **Input Layer** consists of nodes (neurons) that take in the initial data. Each node in the input layer represents a feature of the input data. For example, in a time series forecasting task, each input node might represent a value at a specific time step.

In the diagram, there are two input nodes, each connected to every node in the hidden/recurrent layer. This fully connected structure means that every input feature influences every node in the next layer.

2. Hidden/Recurrent Layer

The **Hidden/Recurrent Layer** is the heart of the RNN. Each node in this layer represents a hidden state that captures information from the input and, crucially, from previous hidden states. This recurrent connection allows the network to retain memory over sequences, making it possible to learn temporal dependencies.

Each hidden node in this layer has a self-loop (shown in red in the diagram), indicating that the output of the hidden node at the previous time step is fed back as input to the same node at the current time step. This loop is what gives the RNN its recurrent nature, allowing it to "remember" information from previous inputs.

The diagram also shows connections from the input layer to the hidden layer, indicating that each hidden node is influenced by both the current input and the previous hidden state. This dual influence is key to how RNNs can model sequences.

3. Output Layer

The **Output Layer** is where the final predictions or classifications are made. In the diagram, each node in the hidden layer is connected to every node in the output layer. These connections mean that the output at any time step depends on the entire hidden state vector.

Get Hamid Mujtaba's stories in your inbox

The output layer can vary in complexity depending on the task. For example, in a many-to-one sequence prediction task (such as sentiment analysis), the output might consist of a single node representing the predicted sentiment. In a many-to-many task (such as machine translation), there might be multiple output nodes representing the translated sequence.

How RNNs Process Sequences

The processing of sequences in an RNN can be understood in the following steps:

1. **Initialization**: At the start of the sequence, the hidden states are usually initialized to zero or some small random values.
2. **Forward Pass**: For each time step in the sequence:
    - The network takes the current input.
    - The hidden state is updated using the current input and the hidden state from the previous time step.
    - The output is calculated based on the current hidden state.

Types of recurrent neural networks

RNNs are often characterized by one-to-one architecture: one input sequence is associated with one output. However, you can flexibly adjust them into various configurations for specific purposes. The following are several common RNN types.

One-to-many: This RNN type channels one input to several outputs. It enables linguistic applications like image captioning by generating a sentence from a single keyword.

Many-to-many: The model uses multiple inputs to predict multiple outputs. For example, you can create a language translator with an RNN, which analyzes a sentence and correctly structures the words in a different language.

Many-to-one: Several inputs are mapped to an output. This is helpful in applications like sentiment analysis, where the model predicts customers' sentiments like *positive, negative*, and *neutral* from input testimonials.

Architecture of RNN

For more clear understanding of the concept of RNN, let's look at the unfolded RNN diagram.

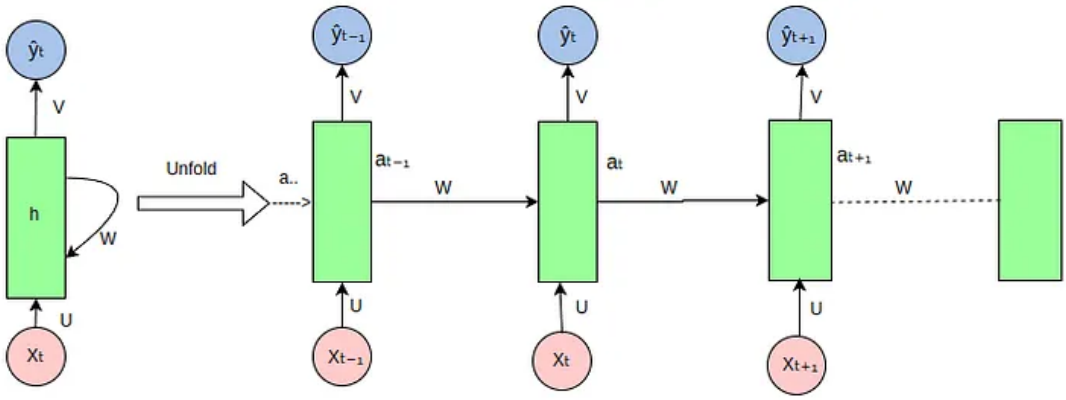


Fig 3.2: RNN Unfolded. Image by Author.

The RNN takes an input vector **X** and the network generates an output vector **y** by scanning the data sequentially from left to right, with each time step updating the hidden state and producing an output. It shares the same parameters across all time steps. This means that, the same set of parameters, represented by U, V, W is used consistently throughout the network. U represents the weight parameter governing the connection from input layer **X** to the hidden layer **h**, W represents the weight associated with the connection between hidden layers, and V for the connection from hidden layer **h** to output layer **y**. This sharing of parameters allows the RNN to effectively capture temporal dependencies and process sequential data more efficiently by retaining the information from previous input in its current hidden state.

At each time step t, the hidden state a_t is computed based on the current input x_t , previous hidden state a_{t-1} and model parameters as illustrated by the following formula:

$$a_t = f(a_{t-1}, x_t; \theta) \text{ — — — — — (1)}$$

It can also be written as,

$$a_t = f(U * X_t + W* a_{t-1} + b)$$

where,

- a_t represents the output generated from the hidden layer at time step t .
- x_t is the input at time step t.
- θ represents a set of learnable parameters (weights and biases).
- U is the weight matrix governing the connections from the input to the hidden layer; $U \in \theta$
- W is the weight matrix governing the connections from the hidden layer to itself (recurrent connections); $W \in \theta$
- V represents the weight associated with connection between hidden layer and output layer; $V \in \theta$
- A_{t-1} is the output from hidden layer at time t-1.
- b is the bias vector for the hidden layer; $b \in \theta$
- f is the activation function.

For a finite number of time steps T=4, we can expand the computation graph of a Recurrent Neural Network, illustrated in Figure 3, by applying the equation (1) T-1 times.

$a_4 = f(a_3, x_4; \theta)$ — — — — — (2)

Equation (2) can be expanded as,

$a_4 = f(U * X_4 + W* a_3 + b)$

$a_3 = f(U * X_3 + W* a_2 + b)$

$a_2 = f(U * X_2 + W* a_1 + b)$

The output at each time step t, denoted as y_t is computed based on the hidden state output a_t using the following formula,

$\hat{y}_t = f(a_t; \theta)$ — — — — — (3)

Equation (3) can be written as,

$\hat{y}_t = f(V * a_t + c)$

When t=4, $\hat{y}_4 = f(V * a_4 + c)$

Where,

- $\hat{Y}_t$ is the output predicted at time step t.
- V is the weight matrix governing the connections from the hidden layer to the output layer.
- c is the bias vector for the output layer.

Number of Layers in RNN

The number of layers refers to how many recurrent (hidden) layers are stacked in the network.

a. **Single-layer RNN**
   - Has one recurrent hidden layer that connects to the input and output.
   - Simpler and faster to train, but limited in its ability to capture complex or hierarchical features.

- Used for simple time-series or short text tasks.

b. **Multi-layer (Stacked or Deep) RNN**

- Has two or more recurrent layers**,** where each layer's output is passed to the next.
- Each higher layer captures more abstract and long-range dependencies**.**
- Common setups:
  - **2–4 layers:** Typical for NLP or sequential modelling.
  - **More than 4:** Used in deep architectures (e.g., speech recognition, video understanding).

Trade-off: Improves performance but increases computational cost and risk of over fitting.

Activation Functions in RNN

Activation functions introduce non-linearity, enabling RNNs to model complex temporal relationships.

Typical usage:

- Hidden layers: tanh or sometimes ReLU
- Gates (in LSTM/GRU): sigmoid
- Output layer: softmax (for classification) or linear (for regression)

Dropout Rates in RNN

Dropout is a regularization technique used to prevent over fitting by randomly dropping units (neurons) during training.

How Dropout Works:

- At each training step, some neurons are "dropped" (temporarily ignored) with a probability equal to the dropout rate**.**
- This prevents the network from relying too heavily on specific neurons.

Typical Dropout Rates:

Layer Type	***Typical Range***	***Notes***
Input Layer	0.1 – 0.3	Light dropout to preserve signal integrity
Recurrent (hidden) Layers	0.2 – 0.5	Applied to prevent over fitting in time steps
Output Layer	0.1 – 0.3	Optional; depends on task complexity

In Keras/TensorFlow**,** you often see:

If.keras.layers.SimpleRNN(128, dropout=0.3, recurrent_dropout=0.2)

Here:

- Dropout=0.3 → drops 30% of input connections
- Recurrent_dropout=0.2 → drops 20% of recurrent connections between time steps

Data Collection

The datasets CICIDS 2017**,** CSE-CIC-IDS 2018, NSL-KDD and UNSW-NB15 used for this study were obtained from publicly available and widely recognized cyber security repositories. The CICIDS 2017 and CSE-CIC-IDS 2018 datasets were sourced from the Canadian Institute for Cyber Security (CIC) which provides realistic and labelled network traffic captured in controlled test bed environments simulating both benign and malicious activities, including DoS and MitM attacks. The NSL-KDD dataset which is an improved version of the original KDD CUP'99 dataset, was downloaded from the University of New Brunswick repository ensuring benchmark comparability with existing intrusion detection research. Additionally, the UNSW-NB15 dataset was accessed from the Australian Centre for Cyber Security (ACCS) offering balanced modern traffic with contemporary attack scenarios. These sources were selected because they are peer-reviewed, widely cited in academic research, and provide structured, labelled data essential for training and evaluating machine learning models for network intrusion detection.

Analysis of the Existing System

The potential drawback of the current system is that they rely on static rules or signatures to identify attacks, which attackers using novel or unidentified attack techniques can easily attack the system. These drawbacks can be addressed by an RNN-based approach, which would learn patterns of typical behaviour in real-time and identify deviations from those patterns as anomalies, doing away with the need for static rules or signatures. The existing system has false negative rate, where the metric counts the instances in which the system fails to detect malicious traffic. An example of a false negative would be when a DoS attack is launched against a system, but the system's detection mechanism misses it. It also has false positive rate, where the metric counts the instances in which the system incorrectly flags legitimate traffic as malicious. For instance, if a firewall is configured to block traffic from a certain IP address but that IP address is actually associated with a legitimate user that would be a false positive. Detection latency also exists in the existing system because, statistic quantifies the time lag between the launch of an attack and the system's ability to detect it. The detection latency would be 10 minutes, for instance, if a DoS attack was launched against a system and it took the system that long to detect the attack.

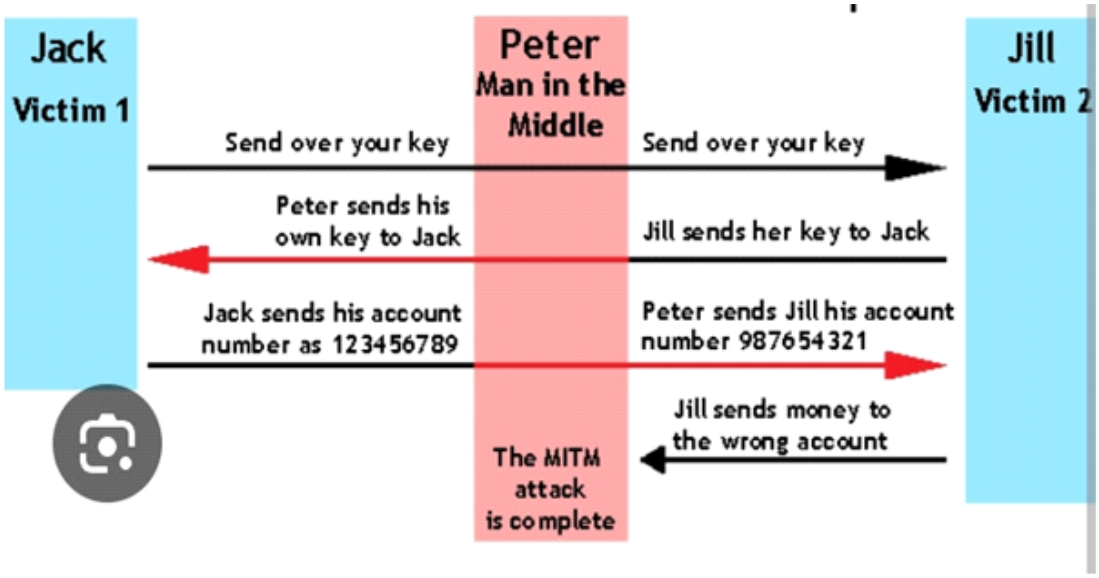


Figure 3.3 Structure of the existing model

Overview of the New System

The newly designed system is a real-time, layered Cybersecurity framework built to detect and mitigate Denial-of-Service (DoS) and Man-in-the-Middle (MitM) attacks with high accuracy and low latency. At its foundation, the system captures raw traffic from the server using packet and flow exporters, which are streamed into a scalable message bus for reliable ingestion. These streams are processed by feature extraction modules that transform packet and session characteristics such as packet rate, byte volume, source diversity, duplicate IP-MAC mappings, and ARP anomalies into structured features that highlight abnormal behaviors. The features are then passed to an RNN-based inference server (LSTM/GRU) that classifies events as benign, DoS, or MitM, with confidence scores normalized and thresholded for decision-making. A policy engine contextualizes these predictions with asset sensitivity and network rules, allowing it to automatically block malicious sources, quarantine compromised switch ports, or escalate suspicious activity to security analysts. All actions, predictions, and traffic metadata are logged in an event store for auditing, replay, and future refinement, while observability components monitor system performance to ensure real-time responsiveness. By integrating automated mitigation with explainable alerts, the system not only safeguards critical infrastructure from high-impact attacks but also empowers human operators with contextual intelligence to make informed security decisions.

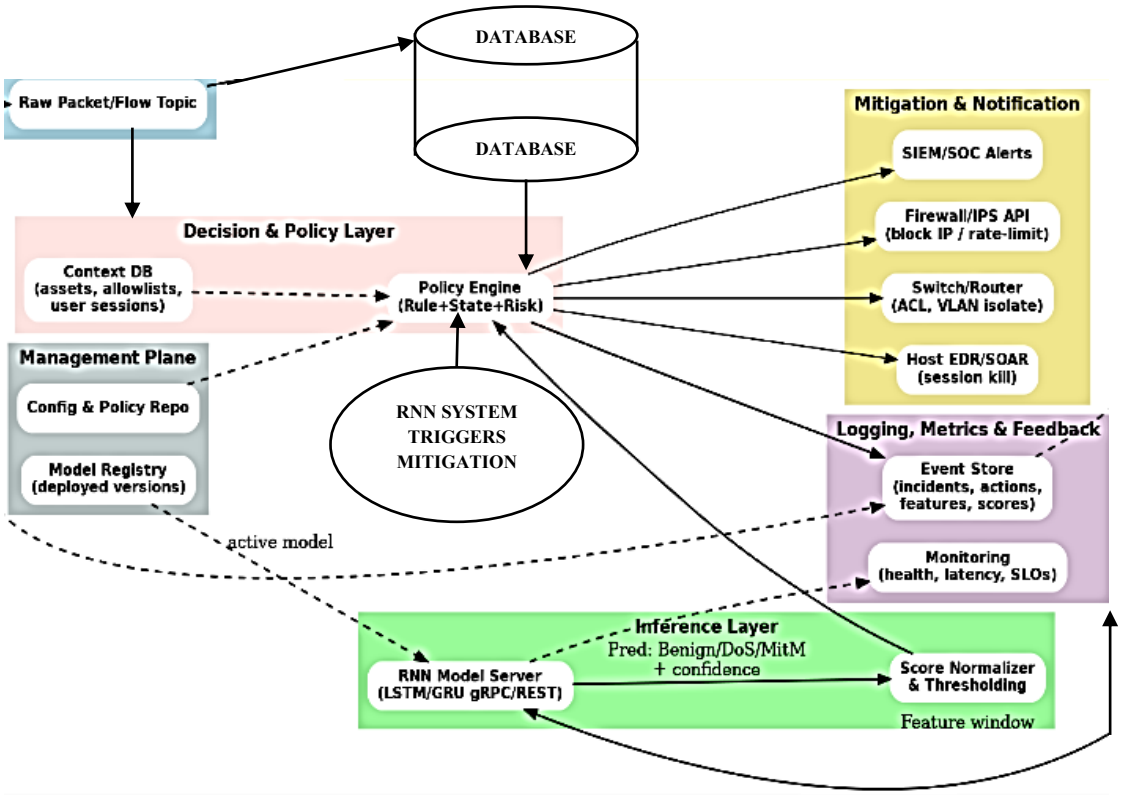


Figure 3.2 Proposed System Model

System Design

The system design employs standard modeling tools to visualize and explain the functionality of the proposed framework. Use Case, Activity, and Sequence diagrams are utilized to represent user interactions, workflows, and component communications within the system. Data Flow Diagrams (DFD) and Flowcharts are applied to illustrate the logical flow of processes and data across different system levels. Finally, the Entity–Relationship (ER) diagram and database design are used to structure and organize data storage for efficient system operation.

UML Use Case Diagram

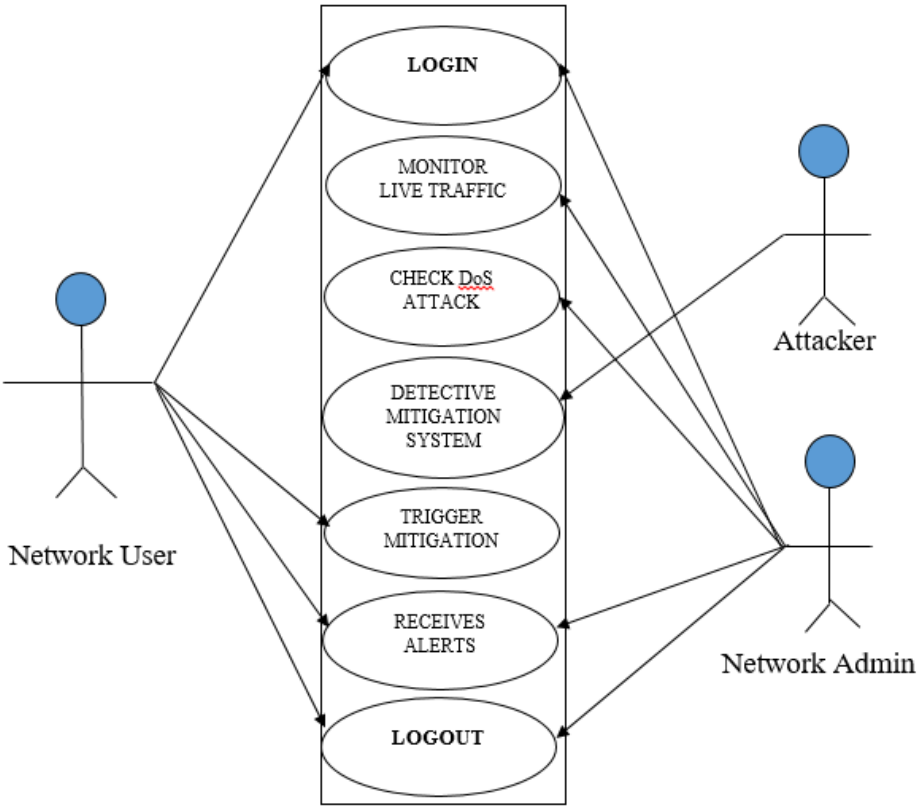


Figure 3.3: UML Use Case for the System

The use case diagram in figure 3.3 illustrates the interaction between two main actors—Security Analyst and Network Admin with the security monitoring system. Both users can log in, monitor live traffic, check for DoS and MitM attacks, trigger mitigation actions, receive alerts, review logs and feedback, and log out. This setup highlights their shared responsibilities in ensuring continuous threat detection, response, and system oversight.

Results and Discussions

This section presents the empirical findings from the evaluation of the proposed RNN-based intrusion detection system, detailing the model's performance against the key metrics of accuracy, precision, recall, and F1-score. The results are systematically presented through tables and graphs, followed by a comprehensive discussion that interprets these outcomes, validates them against the study's objectives, and situates the model's performance within the broader context of contemporary academic research.

High F1 Score

The F1 score is calculated by taking the harmonic mean of recall and precision.

Implication:

- The model successfully strikes a balance between false positives and false negatives.
- It works effectively on datasets that are unbalanced, where precision by itself can be deceptive.

A high F1 score indicates:

High precision: there are few false positives when the model misclassifies benign (normal) data as assaults.

High recall: the majority of real attacks are effectively detected by the model (low false negatives).

In terms of cyber security, this means that:

- The model is dependable for automated responses and real-time threat detection;
- The detection system can precisely identify attack patterns while reducing false alarms.

AUC (Area under the ROC Curve) is high. The model's ability to differentiate between classes—such as attack and normal traffic—is gauged by the AUC-ROC.

Implication: Across all decision thresholds, the model gives actual attacks a better ranking than non-attacks.

- Excellent reparability is indicated by a high AUC (around 1.0), which shows that the model can distinguish between harmful and legitimate behaviour.
- It demonstrates the model's resilience to threshold modifications, which is helpful when operational thresholds change.

In terms of cyber security, this means that:

- The model is flexible and generalizes well to new or changing attack patterns.
- It can accurately identify intrusions across a range of sensitivity levels.

Overall conclusion: A high F1 and AUC indicate that the model is balanced, accurate, and has good generalization, which is a crucial characteristic for identifying evolving or zero-cyber threats of the day.

Even if overall performance stays high, a small decrease in precision for Man-in-the-Middle (MitM) attack detection has significant ramifications for real-world deployment. What it signifies and why it matters are as follows:

1. **Definition of a Precision Drop:** Precision quantifies the proportion of identified assaults that are MitM attacks.
   - A decrease in precision indicates that the model is generating more false positives; occasionally, it misidentifies innocuous or typical traffic as a MitM assault.
2. **Real-World Implications:**
   (a) Additional False Alerts: More bogus MitM alerts may be sent to network managers, which can lead to alert fatigue and operators ignoring or missing critical warnings. That gradually erodes confidence in the detecting system.
   (b) Operational Overhead: Security teams might waste resources by spending more time looking into unrelated problems, and automated defensive systems

might mistakenly identify secure communications as threats, which would interfere with services or connections.

(c) Impact on the User Experience. False alarms may disrupt valid encrypted connections in IoT or business settings. For instance, obstructing secure TLS/SSL communications may result in brief delays or failures in communication.

(d) Model Adjustment required: The precision loss indicates that retraining with more representative data or adjusting the threshold may be necessary for the model. It might also be a sign of data drift, which is the occurrence of novel or unobserved network behaviors that resemble MitM patterns.

3. **A Fair Perspective:** A small decrease in accuracy is not always detrimental, particularly if:
   • Recall is strong (the model still recognizes the majority of actual MitM attacks).
   • Human analysts can control the false positive rate.
   • Feedback loops are incorporated into the system to retrain and gradually lower false positives.

Results

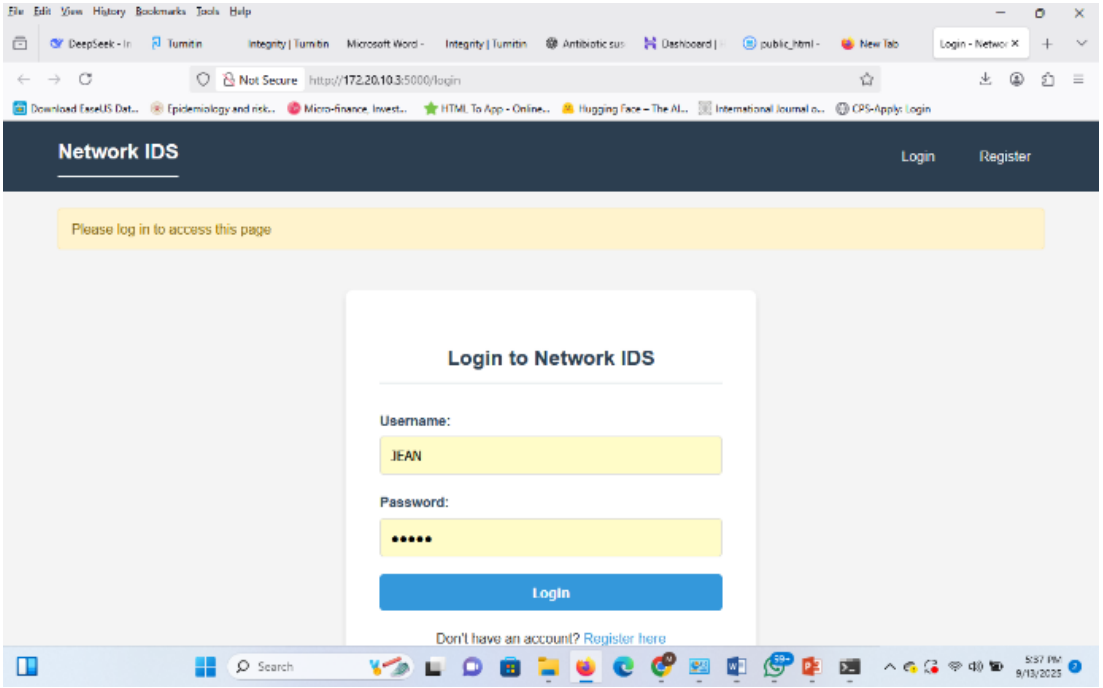


Figure 4.1: User Login

Figure 4.1 depicts the user authentication interface, which serves as the critical entry point for system access control. This interface implements a secure credential-based authentication protocol, requiring users to input a unique identifier and a confidential passphrase. Its primary

function is to verify user identity and enforce authorization policies, ensuring that only privileged users with valid credentials can access the protected administrative functionalities of the system, thereby maintaining the confidentiality and integrity of the Network monitoring environment.

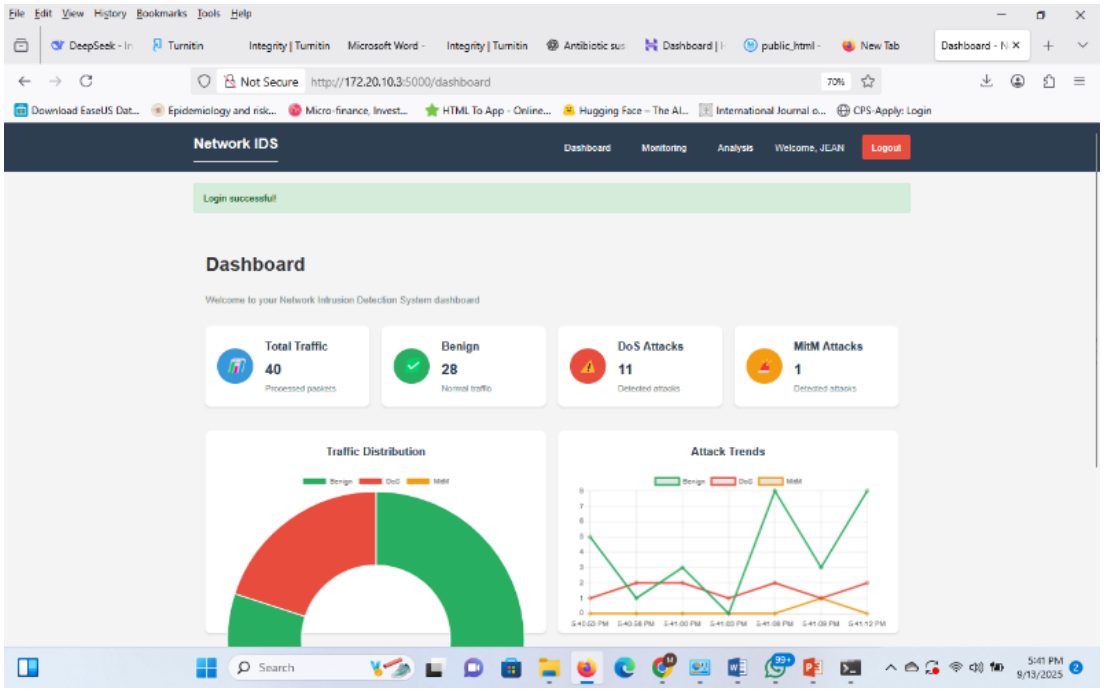


Figure 4.2: System Dashboard

Figure 4.2 illustrates the primary system dashboard, which functions as a centralized graphical user interface (GUI) for network security situational awareness. It provides a high-level, aggregated visualization of key performance indicators (KPIs) and system health metrics through widgets, charts, and summary statistics. This dashboard offers security personnel an at-a-glance overview of the operational status, including current threat levels, traffic volume, and alert summaries, facilitating rapid assessment and informed decision-making for subsequent analytical or investigative actions.

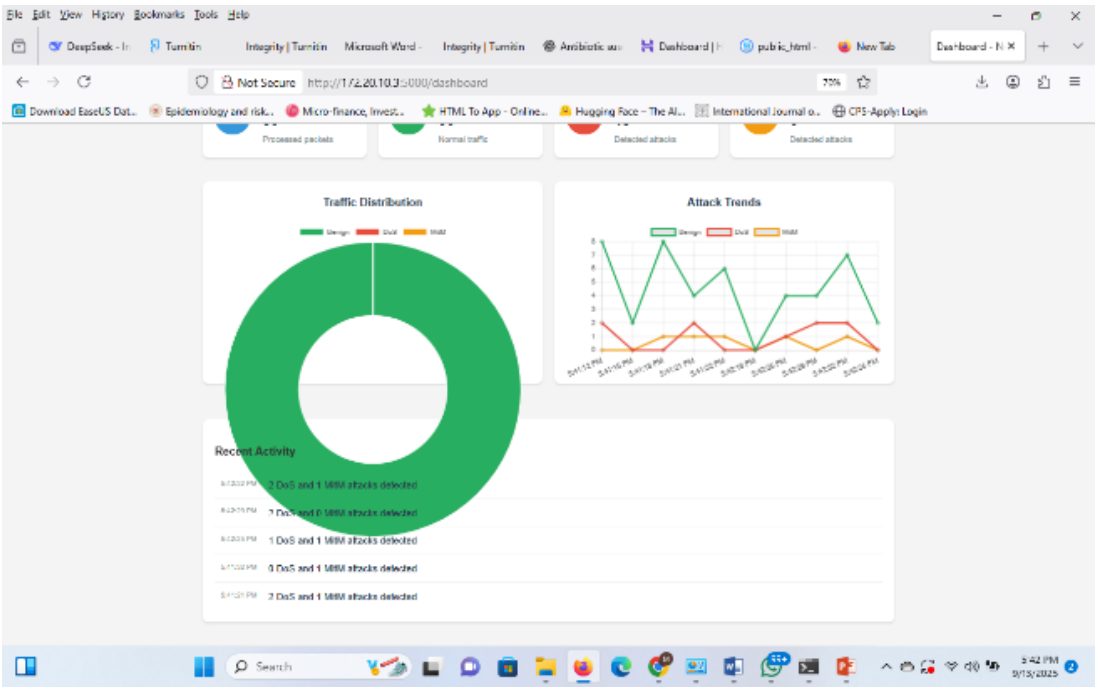
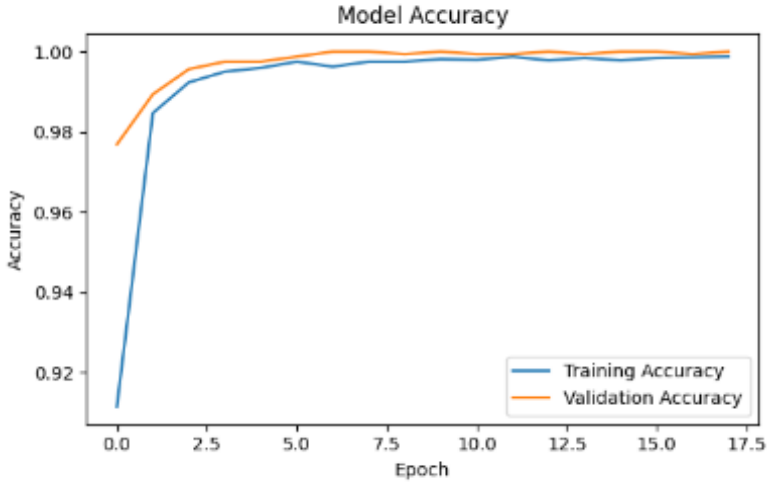


Figure 4.3: Traffic trend Stats

Figure 4.3 presents a temporal analysis of network traffic statistics, visualized through time-series data plots such as line charts or histograms. This analytical view is essential for conducting longitudinal traffic analysis, enabling the identification of patterns, trends, and anomalies in network flow data over a selected timeframe. By examining metrics like packet count, bandwidth utilization, and session duration, analysts can perform baseline modelling to distinguish normal operational behaviour from potential deviations that may indicate malicious activity or performance degradation.

Performance Evaluation



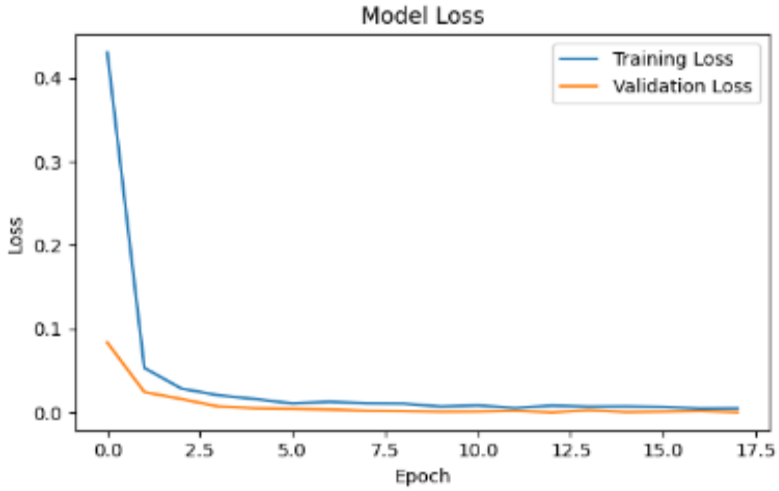


Figure 4.5: Model training Epoch graph

Figure 4.5 illustrates training and validation loss trajectories decreasing and converging to a low value, indicating stable optimization and successful convergence without obvious over fitting (training and validation losses remain close throughout). The smooth descent suggests appropriate hyper parameter settings and learning dynamics.

Table 4.1 Classification report

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
DoS	0.997	0.985	0.991	332.0
MitM	0.982	0.997	0.99	335.0
Normal	0.997	0.994	0.995	333.0

Table 4.1 reports very high per-class metrics: DoS (Precision=0.997, Recall=0.985, F1=0.991, Support=332), MitM (Precision=0.982, Recall=0.997, F1=0.990, Support=335), and Normal (Precision=0.997, Recall=0.994, F1=0.995, Support=333). These values indicate balanced, near-optimal performance across the three classes with only tiny class-specific differences; in particular, MitM shows slightly higher recall but slightly lower precision relative to DoS, indicating a small trade-off between missed attacks and false alarms. (See Table 4.1)

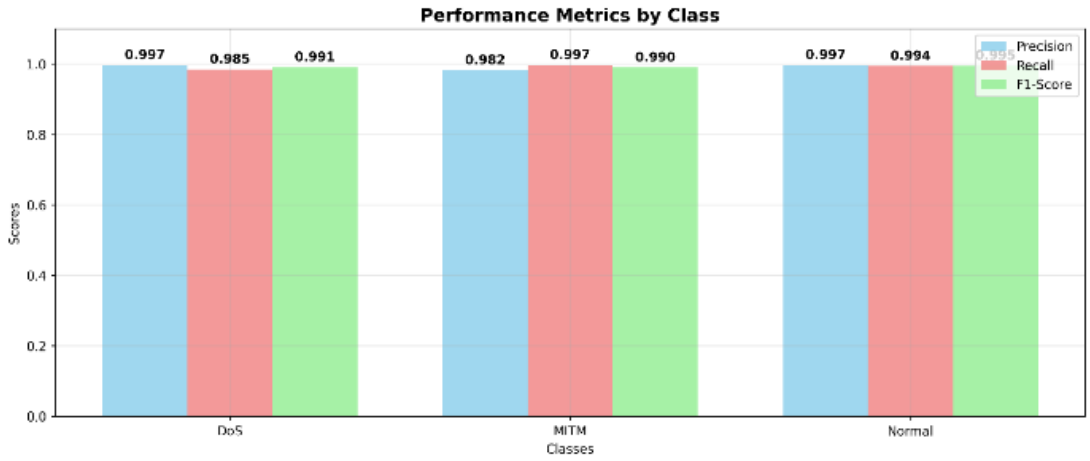


Figure 4.6: Model Performance Metrics by class

Figure 4.6 visualizes precision, recall and F1 for each class and corroborates Table 4.1: bars for all three metrics lie in the high-performance band (≈0.98–0.997). The chart makes class-level differences easily visible (e.g., marginally lower precision for MitM).

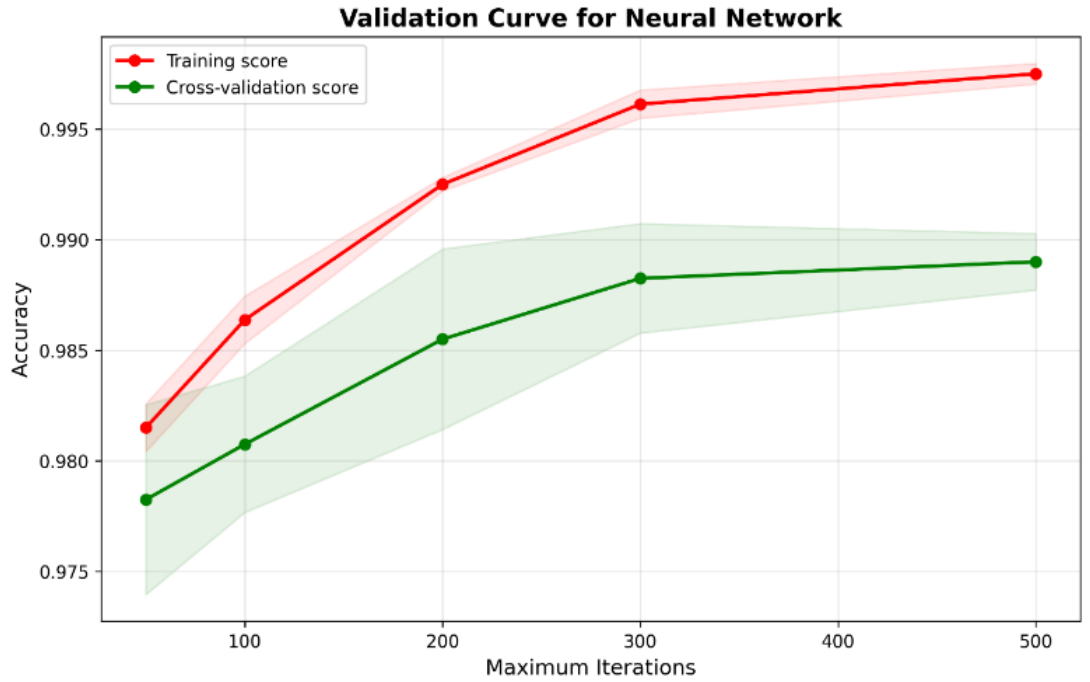
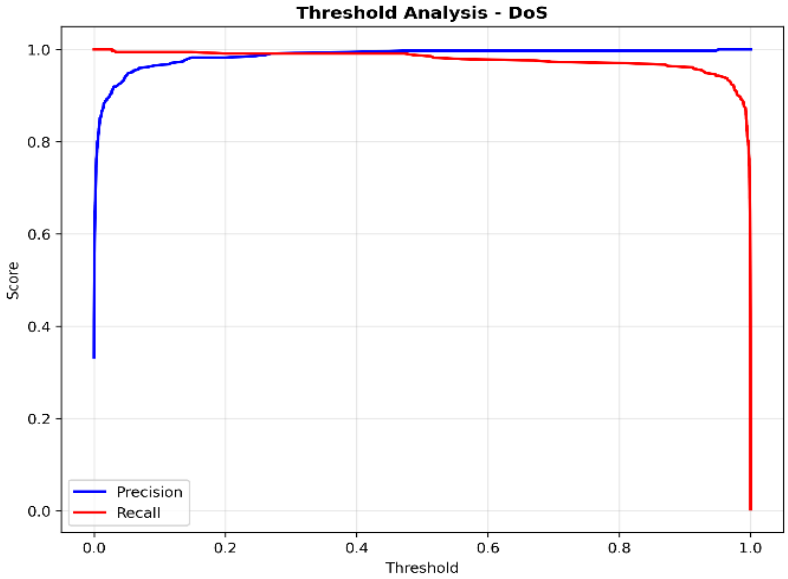
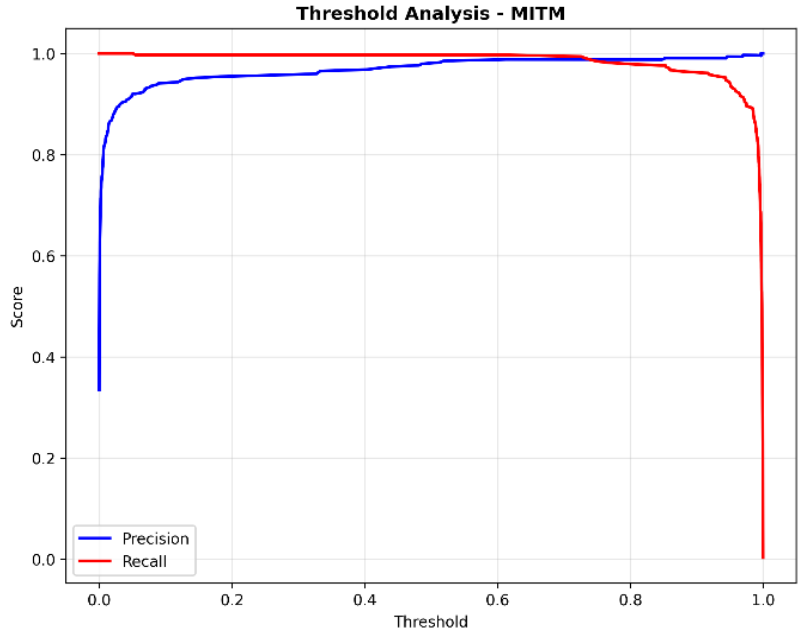


Figure 4.7: Validation Curve for Neural Network

Figure 4.7 shows validation performance (e.g., accuracy or F1) across parameter/epoch configurations; the curve indicates consistent improvement followed by plateau, which suggests model capacity is appropriate and additional tuning yields diminishing returns.


Threshold Analysis - DoS
Precision
Recall
Score
Threshold
0.0
0.2
0.4
0.6
0.8
1.0


Threshold Analysis - MITM
Precision
Recall
Score
Threshold
0.0
0.2
0.4
0.6
0.8
1.0

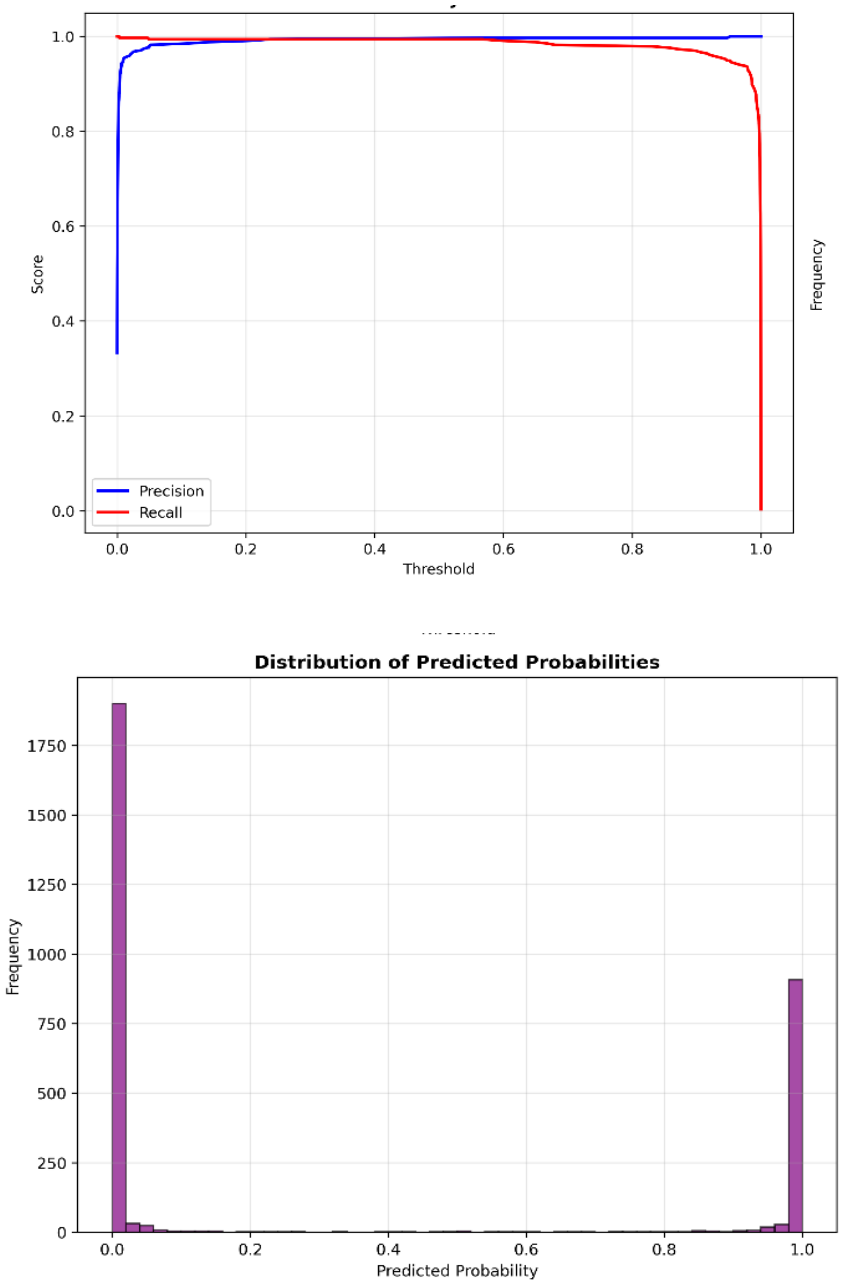


Figure 4.8: Threshold Analysis and distribution of predicted probabilities

Figure 4.8 presents predicted-probability distributions and threshold analysis: the plot highlights score reparability between classes and shows how varying the classification threshold will trade precision vs. recall. The distribution indicates a generally well-calibrated classifier with distinct peaks for confident predictions.

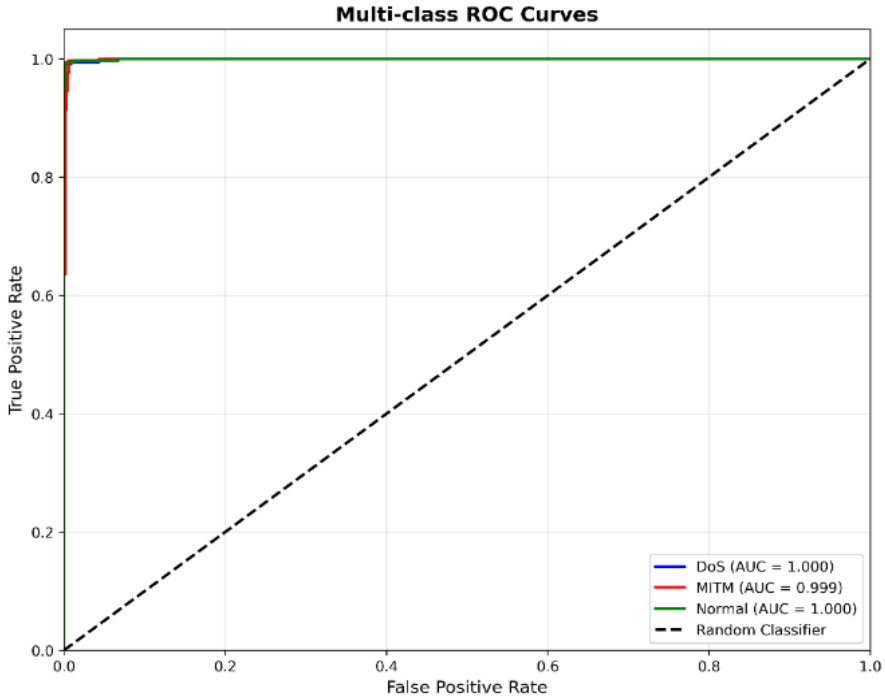


Figure 4.9: Multi-class ROC Curves

Figure 4.9 contains ROC curves for the three classes (one-vs-rest). The curves lie well above the chance diagonal, indicating strong discrimination power and high AUC values for each class.

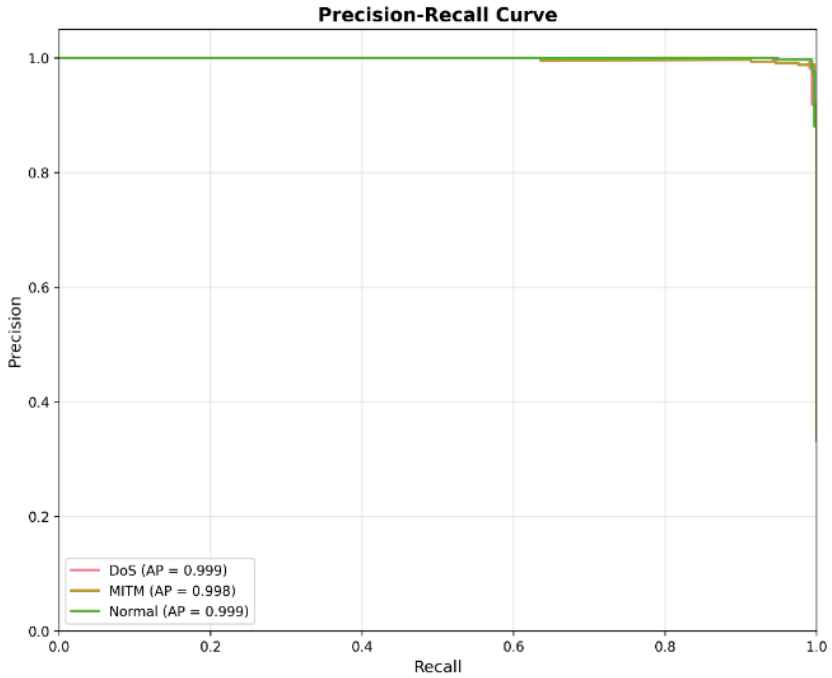


Figure 4.10: Precision Recall Curves

Figure 4.10 shows PR curves for each class; these curves remain near the upper-right region, confirming high precision across a wide span of recall values — especially important under class imbalance.

Conclusion

In conclusion, this study successfully designed, implemented, and validated an optimized RNN-based intrusion detection system that demonstrates state-of-the-art performance in accurately and reliably detecting DoS and MitM attacks, thereby making a significant contribution towards enhancing proactive cyber defense mechanisms. The model was trained and evaluated on a combination of publicly available, benchmark cyber security datasets—such as CICIDS2017, CSE-CIC-IDS2018, and UNSW-NB15—alongside additional attack traffic to capture diverse characteristics of benign and malicious network flows. This provided a realistic and comprehensive basis for assessing the system's detection capability across varied network conditions.

Performance evaluation of the RNN model yielded highly promising results, with accuracy, precision, recall, and F1-scores consistently ranging between 0.989 and 0.992. These findings confirm the system's robustness in distinguishing between normal traffic and malicious activity with minimal false positives and false negatives. The balance between precision and recall highlights the model's effectiveness in both avoiding unnecessary alerts and ensuring comprehensive detection of attacks. The training process demonstrated gradual and stable convergence, with learning curves for accuracy and loss indicating the absence of severe over fitting and strong generalization capacity. This implies that the system is well-suited to handle previously unseen traffic patterns, which is essential for real-world deployment where novel attack variations frequently emerge.

Finally, future researchers should employ a hybrid dataset approach, as done in this study, to ensure model robustness and generalizability across different network environments.

References

Abdullahi Wabi, A.; Idris, I.; Mikail Olaniyi, O.; Joseph, A.; Surajudeen Adebayo, O. Modeling DDOS attacks in sdn and detection using random forest classifier. *J. Cyber Secur. Technol.* **2023**, 1–14.

Abrar, I.; Ayub, Z.; Masoodi, F.; Bamhdi, A.M. A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. In Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 18–20 September 2020; pp. 919–924.

Ahalawat, A.; Dash, S.S.; Panda, A.; Babu, K.S. Entropy based DDoS detection and mitigation in OpenFlow enabled SDN. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.

Ahmad, F., Adnane, A., Franqueira, V. N., Kurugollu, F., & Liu, L. (2018). Man-in-the-middle attacks in vehicular ad-hoc networks: Evaluating the impact of attackers' strategies. *Sensors*, *18*(11), 4040.

Ahmad, S.; Mir, A.H. SDN Interfaces: Protocols, Taxonomy and Challenges. *Int. J. Wirel. Microwave Technol.* **2022**, *12*, 11–32.

Ahmad, Z., et al. (2023). A novel deep learning framework for intrusion detection in IoT networks. *Computers and Electrical Engineering*, 110, 108853. https://doi.org/10.1016/j.compeleceng.2023.108853

Ahmed, C. M., Zhou, J., & Mathur, A. P. (2018). *Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS.* In *Proceedings of the 34th Annual Computer Security Applications Conference* (pp. 566–581). Association for Computing Machinery.

Ahmed, F., Sumra, I. A., & Jamil, U. (2024). A Comprehensive Review on DDoS Attack in Software-Defined Network (SDN): Problems and Possible Solutions. *Journal of Computing & Biomedical Informatics*, *7*(01).

Aizuddin, A. A., Atan, M., Norulazmi, M., Noor, M. M., Akimi, S., & Abidin, Z. (2017, January). DNS amplification attack detection and mitigation via sFlow with security-centric SDN. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication* (pp. 1-7).

Al Zubaidi, A. (2021). *Measuring the level of cyber-security awareness for cybercrime in Saudi Arabia*. *Heliyon, 7*(1), e06016.

Al-Abassi, A.; Karimipour, H.; Dehghantanha, A.; Parizi, R.M. An Ensemble deep learning-based cyber-attack detection in industrial control system. *IEEE Access* 2020, *8*, 83965–83973.

Alashhab, A. A., Zahid, M. S. M., Azim, M. A., Daha, M. Y., Isyaku, B., & Ali, S. (2022). A survey of low rate ddos detection techniques based on machine learning in software-defined networks. *Symmetry*, *14*(8), 1563.

Al-Dunainawi, Y.; Al-Kaseem, B.R.; Al-Raweshidy, H.S. Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment. *IEEE Access* 2023, *11*, 106733–106748.

Algarni, A. M., Thayananthan, V., & Malaiya, Y. K. (2021). *Quantitative Assessment of Cybersecurity Risks for Mitigating Data Breaches in Business Systems*. *Applied Sciences, 11*(8), Article 3678.

Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M. Z. (2021). Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences*, *11*(24), 11634.

Alhaidari, F.A.; AL-Dahasi, E.M. New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 3–4 April 2019; pp. 1–6.

Alheeti, K. M. A., Iftikhar, S., Khan, D., Al-Madani, D., Fatima, K., (2022). *An intelligent detection of malicious intrusions in IoT based on machine learning and deep learning techniques*. *Comput. Sci. J. Moldova, 30*(3), 288-307

Alhijawi, B.; Almajali, S.; Elgala, H.; Salameh, H.B.; Ayyash, M. A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets. *Comput. Electr. Eng.* 2022, *99*, 107706.

Ali, M., Benamrane, F., Luong, D. K., Hu, Y. F., Li, J. P., & Abdo, K. (2019, September). An AI based approach to secure SDN enabled future avionics communications network against DDoS attacks. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)* (pp. 1-7). IEEE.

Ali, S., & Li, Y. (2019). Learning multilevel auto-encoders for DDoS attack detection in smart grid network. *IEEE Access*, *7*, 108647-108659.

Ali, S., Rehman, S. U., Imran, A., Adeem, G., Iqbal, Z., & Kim, K. I. (2022). Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection. Electronics 2022, 11, 3934.

Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine learning techniques to detect a DDoS attack in SDN: A systematic review. *Applied Sciences*, *13*(5), 3183.

Ali, T.E.; Chong, Y.W.; Manickam, S. Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Appl. Sci.* **2023**, *13*, 3183.

Aljuhani, A. (2021). Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access*, *9*, 42236-42264.

Almomani, I. M., & Alenezi, M. (2018). Efficient Denial of Service Attacks Detection in Wireless Sensor Networks. *J. Inf. Sci. Eng.*, *34*(4), 977-1000.

Al-Rabiaah, S. The "Stuxnet" Virus of 2010 As an Example of A "APT" and Its "Recent" Variances. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–5.

Alsaedi, A., et al. (2024). TönT: Transformer-based model for Internet of Things intrusion detection. *Scientific Reports*, *14*, 8340. https://doi.org/10.1038/s41598-024-58900-3

Alubaidan, H.; Alzaher, R.; AlQhatani, M.; Mohammed, R. DDoS Detection in Software-Defined Network (SDN) Using Machine Learning. *Int. J. Cybern. Inform.* 2023, *12*, 93–104.

Alzubaidi, M., Hasan, K. N., & Meegahapola, L. (2022, November). Identification of the Impact of Wind Speed and Load Uncertainties on Short-term Voltage Stability. In *2022 IEEE PES 14th Asia-Pacific Power and Energy Engineering Conference (APPEEC)* (pp. 1-7). IEEE.

Amin, S.; Litrico, X.; Sastry, S.; Bayen, A. Cyber security of water SCADA systems- part II: Attack detection using enhanced hydrodynamic models. *IEEE Trans. Control Syst. Technol.* 2013, *21*, 1679–1693.

Anand, P., Singh, Y., Singh, H., Alshehri, M. D., & Tanwar, S. (2022). SALT: transfer learning-based threat model for attack detection in smart home. *Scientific Reports*, *12*(1), 12247.

Asad, M., Asim, M., Javed, T., Beg, M. O., Mujtaba, H., & Abbas, S. (2020). Deepdetect: detection of distributed denial of service attacks using deep learning. *The Computer Journal*, *63*(7), 983-994.

Asad, M., Moustafa, A., & Yu, C. (2020). *A critical evaluation of privacy and security threats in federated learning.* Sensors, 20(24), 7182.

Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.A.A.; Jilani, S.F. Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors* 2022, *22*, 2697.

Assis, M. V., Carvalho, L. F., Lloret, J., & Poença Jr, M. L. (2021). A GRU deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, *177*, 102942.

Aupetit, M., Zhauniarovich, Y., Vasiliadis, G., Dacier, M., & Boshmaf, Y. (2016, October). Visualization of actionable knowledge to mitigate DRDoS attacks. In *2016 IEEE symposium on visualization for cyber security (VizSec)* (pp. 1-8). IEEE.

Azzedin, F. (2023). Mitigating Denial of Service Attacks in RPL-Based IoT Environments: Trust-Based Approach. *IEEE Access*, *11*, 129077–129089. https://doi.org/10.1109/access.2023.3331030

Banitalebi Dehkordi, A.; Soltanaghaei, M.R.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* 2021, *77*, 2383–2415.

Bastola, S.B.; Shakya, S.; Sharma, S. Distributed Denial of Service Attack Detection on Software Defined Networking Using Deep Learning. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017.

Behal, S.; Singh, J. Detection and Mitigation of DDoS attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions. *Comput. Sci. Rev.* **2020**, *37*, 100279.

Bezemskij, A., Loukas, G., Gan, D., & Anthony, R. J. (2017). *Detecting cyber-physical threats in an autonomous robotic vehicle using Bayesian networks* [Conference paper].

Bhardwaj, A., Mangat, V., & Vig, R. (2020). Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud. *IEEE Access*, *8*, 181916-181929.

Bhayo, J., Hameed, S., & Shah, S. A. (2020). An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT). *IEEE Access*, *8*, 221612-221631.

Bierbrauer, D.A.; Chang, A.; Kritzer, W.; Bastian, N.D. Cybersecurity Anomaly Detection in Adversarial Environments. *arXiv* **2021**, arXiv:2105.06742.

Cao, J.; Xu, M.; Li, Q.; Sun, K.; Yang, Y.; Zheng, J. Disrupting SDN via the data plane: A low-rate flow table overflow attack. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Niagara Falls, ON, Canada, 22–25 October 2017; Springer: Cham, Switzerland, 2017; pp. 356–376.

Carvalho, R.N.; Bordim, J.L.; Alchieri EA, P. Entropy-based DoS attack identification in SDN. In Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 20–24 May 2019; IEEE: New York, NY, USA, 2019; pp. 627–634.

Catak, F. O., & Mustacoglu, A. F. (2019). Distributed denial of service attack detection using autoencoder and deep neural networks. *Journal of Intelligent & Fuzzy Systems*, *37*(3), 3969-3979.

Chen, J., Yang, Y. T., Hu, K. K., Zheng, H. B., & Wang, Z. (2019, February). DAD-MCNN: DDoS attack detection via multi-channel CNN. In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing* (pp. 484-488).

Chen, J.; Zheng, X.; Rong, C. Survey on software-defined networking. In Proceedings of the Second International Conference on Cloud Computing and Big Data in Asia, Huangshan, China, 17–19 June 2015; Springer: Cham, Switzerland, 2015; pp. 115–124.

Choi, W., Chang, Y. S., Jung, Y., & Song, J. (2018). Low-Power LoRa signal-based outdoor positioning using fingerprint Algorithm. *ISPRS International Journal of Geo-Information*, *7*(11), 440.

Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*, *169*, 114520.

Cui, Y.; Yan, L.; Li, S.; Xing, H.; Pan, W.; Zhu, J.; Zheng, X. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *J. Netw. Comput. Appl.* **2016**, *68*, 65–79.

Deepa, V.; Sivakumar, B. Detection of DDoS Attack using Multiple Kernel Level (MKL) Algorithm. In Proceedings of the 2022 International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 12–13 February 2022; IEEE: New York, NY, USA, 2022; pp. 1–5.

Deshpande, A., Pedamkar, O., Chaudhary, N., & Borde, S. (2021). Detection of phishing websites using Machine Learning. *International Journal of Engineering Research & Technology (IJERT)*, *10*(05).

Dong, S., Abbas, K., & Jain, R. (2019). A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. *IEEE Access*, *7*, 80813-80828.

Dong, S.; Sarem, M. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access* **2019**, *8*, 5039–5048.

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del-Rincon, J., & Siracusa, D. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Transactions on Network and Service Management*, *17*(2), 876-889.

Dover, J.M. *A Denial of Service Attack against the Open Floodlight SDN Controller*; Dover Networks LCC.: Edgewater, MD, USA, 2013.

Dover, J.M. *A Denial of Service Attack against the Open Floodlight SDN Controller*; Dover Networks LCC.: Edgewater, MD, USA, 2013.

El Naqa, I.; Murphy, M.J. What Is Machine Learning? Springer: Berlin/Heidelberg, Germany, 2015; pp. 3–11.

ElSayed, M. S., Le-Khac, N. A., Albahar, M. A., & Jurcut, A. (2021). A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique. *Journal of Network and Computer Applications*, *191*, 103160.

Elsayed, N., Maida, A. S., & Bayoumi, M. (2020). Reduced-gate convolutional long short-term memory using predictive coding for spatiotemporal prediction. *Computational Intelligence*, *36*(3), 910-939.

Fahad, A. M., Alomary, F. O., Alwadi, H. F., & Albusayli, R. R. (2018). Ontology-based cyber security policy implementation in Saudi Arabia. *Journal of Information Security*, 9(4), Article ID 88030.

Fouladi, R.F.; Ermiş, O.; Anarim, E. A DDoS attack detection and defense scheme using time-series analysis for SDN. *J. Inf. Secur. Appl.* 2020, *54*, 102587.

Gondim, J. J., de Oliveira Albuquerque, R., & Orozco, A. L. S. (2020). Mirror saturation in amplified reflection Distributed Denial of Service: A case of study using SNMP, SSDP, NTP and DNS protocols. *Future Generation Computer Systems*, *108*, 68-81.

Guozi Sun Jiang, W.; Yu, G.U.; Danni, R.E.N.; Huakang, L.I. DDoS attacks and flash event detection based on flow characteristics in SDN. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; IEEE: New York, NY, USA, 2018; pp. 1–6.

Gupta, B. B., & Dahiya, A. (2017). *Distributed Denial of Service (DDoS) Attacks: Classification,*

Javeed, D., Khan, M. T., Ahmad, I., Iqbal, T., Badamasi, U. M., Ndubuisi, C. O., & Umar, A. (2020). *An efficient approach of threat hunting using memory forensics*. *International Journal of Computer Networks and Communications Security*, 8(5), 37–45

Kim, K., Alfouzan, F. A., & Kim, H. (2021). *Cyber-Attack Scoring Model Based on the Offensive Cybersecurity Framework. Applied Sciences, 11*(16), 7738.

Liu, J., et al. (2023). *Prompt injection attacks and defenses in LLM-integrated applications* [Preprint / conference paper]. Semantic Scholar / paper record.

Premkumar, M., & Sundararajan, T. V. P. (2020). *DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocessors and Microsystems, 79*, 103278.

Sabeel, U., Heydari, S. S., Mohanka, H., Bendhaou, Y., Elgazzar, K., & El-Khatib, K. (2019). *Evaluation of deep learning in detecting unknown network attacks.* In *Proceedings of the 2019 International Conference on Smart Applications, Communications and Networking (SmartNets)* (pp. 1–6). IEEE.

Sajo, A & Sarjiyus, O. *Neural Network for Mitigating Denial of Service (DoS) and Man in the Middle (MitM) Attacks in Cyberspace. International Journal of Science Research and Technology Vol. 8 No. 9 E-ISSN 3026-8796 P-ISSN 3026-8095, 2025.*

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). *Toward generating a new intrusion detection dataset and intrusion traffic characterization.* In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)* (pp. 108–116). ScitePress.

Singh, J., & Behal, S. (2020). Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Computer Science Review, 37*, Article 100279.

Thakur, K., & Kumar, S. (2021). *Cybersecurity threats and defense mechanisms in modern networks. International Journal of Computer Applications*, 183(22), 15–21.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust intelligent malware detection using deep learning. *IEEE Access, 7*, 46717–46738.