An Intelligent Intrusion Detection System for DDoS Attacks Using Deep Neural Networks

D. A Aina ainad@mcu.edu.ng
Department of Computer Science,
College of Computing,
McPherson University,
Seriki Sotayo, Ogun state, Nigeria

J.A Ayeni ja.ayeni@acu.edu.ng
Department of Computer Science,
Faculty of Natural Sciences,
Ajayi Crowther University, Oyo, Nigeria

F. E. Ayo, emmini8168@gmail.com
Department of Mathematical Sciences,
Faculty of Science,
Olabisi Onabanjo University,
Ogun Satte, Nigeria

A. O. Ogunjobi adebayo.ogunjobi@gmail.com
Department of Computer science
College of Natural Sciences
Federal University of Agricultural Sciences, Abeokuta,
Ogun State, Nigeria

Abstract

The emergent dependence on internet-based facilities highlights the exigent need for strong network security, especially in alleviating Distributed Denial-of-Service (DDoS) outbreaks, which seriously disrupt service accessibility and cause significant financial losses. DDoS attacks devastate targeted systems with large volumes of traffic from numerous sources, resulting to downtime and performance dilapidation. Prompt recognition of such attacks remains a serious challenge in cybersecurity. Existing methods often suffer from high false positive rates and inadequate capability to detect the various and complex traffic patterns related with contemporary DDoS attacks, resulting in limited accuracy.

This research work presents an enhanced intrusion detection framework leveraging deep learning techniques for effective identification of DDoS attacks. Three architectures Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU) were used on the CICDDoS2019 dataset sourced from Kaggle. Comparative evaluation shows that the CNN model attained higher performance, exhibiting an accuracy of 99.73%, precision of 99.70%, recall of 99.85%, and F1-score of 99.77%. These outcomes reveal CNN's ability to effectively distinguish between benign and harmful outbreaks while reducing false positives and false negatives. The outcomes validate the effectiveness of deep learning, especially CNN-based models, in exhibiting extremely accurate early exposure of DDoS outbreaks, thus improving network resilience against emerging cyber threats.

Keywords: Distributed Denial-of-Service (DDoS), Deep Learning, Intrusion Detection System (IDS), Convolutional Neural Network (CNN), Network Security, Cybersecurity

I. Introduction

Machine learning (ML) is a branch of artificial intelligence (AI) that involves the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, these systems learn from data and improve their performance over time. Machine learning though powerful, often struggles with high-dimensional data, Complex patterns, and the need for extensive feature engineering, making it less effective for tasks like image and speech recognition [18].

Deep learning, on the other hand, excels in these areas by automatically learning hierarchical features from raw data, handling large volumes of data more effectively, and capturing intricate patterns through its multi-layered neural network architectures.

Distributed Denial of Service (DDoS) attacks are malicious attempts to interfere with a server, service, or network's regular operation by flooding the target or the infrastructure around it with an excessive amount of Internet traffic as shown in figure 1. The efficiency of DDoS attacks arises from their ability to use

several hacked computer systems as sources of attack traffic [11]. One of the earliest DDoS attacks was carried out in 2000 by Michael Calce, also known online as "Mafia boy." He breached the computer systems of many colleges. He launched a DDoS attack using their servers, taking down many websites, including eBay and Yahoo. In 2016, Dyn was hit with a huge DDoS attack that took down major websites and services such as Netflix, PayPal, Amazon, and GitHub. With this, many companies and researchers have shifted their attention in recent years to creating more secure, scalable, and robust networks [9].

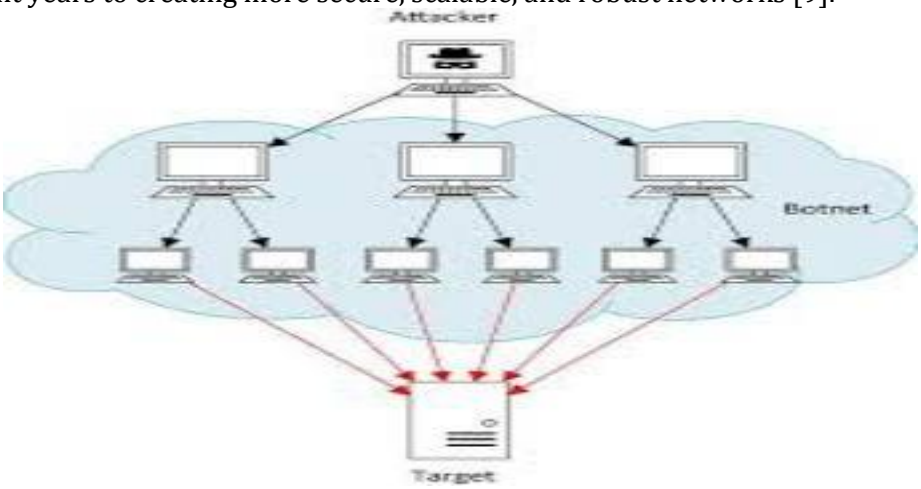


Figure 1: DDoS Process (Source: AlSalel et al., 2024).

Due to network's rapid growth and its direct impact on the interests of the country, businesses, and people, network security has become one of the biggest issues. An effective and efficient network infrastructure serves as the cornerstone of a secure digital environment. It encompasses the hardware, software, and communication protocols that facilitate data transmission and connectivity between devices. Network has always been susceptible to various security threats such as denial of service attacks, worms, port scans and trojans and so on [9]. In recent years, an exponential increase in DDoS attacks was discovered which had incapacitated businesses and organizations in many occasions.

In February of 2020, Amazon Web Services (AWS) suffered a DDoS attack sophisticated enough to keep its incident response teams occupied for several days also affecting customers worldwide [16]. In February of 2021, the EXMO Cryptocurrency exchange fell victim to a DDoS attack that rendered the organization inoperable for almost five hours.

Recently, Australia experienced a significant, sustained, state-sponsored DDoS attack. Belgium also became a victim of a DDoS attack that targeted the country's parliament, police services and universities [8]. Distributed Denial-of-Service (DDoS) attacks pose a significant threat, and early detection is crucial for mitigating their impact. Deep learning offers the best technique in recognizing these outbreak with greater accuracy and efficiency.

There are two key classes of DDoS recognition techniques;

Signature-based detection which depends on pre-defined arrangements or signaturezbxs of identified DDoS outbreaks and the Network movement which is matched alongside these signatures, such that any movement corresponding to a known outbreak pattern is labelled as malicious and Anomaly-based recognition which looks for deviances from regular network movement arrangements [7]. It explores numerous network circulation features like volume of traffic flow, packet size, source IP addresses, and protocol usage.

These outmoded detection techniques scuffle to maintain speed with embryonic outbreak policies. Deep learning provides a potent answer by studying huge volumes of network data. Deep learning models can recognize elusive irregularities revealing DDoS occurrences. This propensity to quickly study and adjust enables them to recognize recent outbreak occurrences early. Deep learning provides an organizations with the means of combating against DDoS outbreaks [15], leading to quicker recognition, enriched network security, and a more dependable user experiences.

The purpose for this study is to develop a deep learning-based technique to identify and alleviate Distributed Denial of Service (DDoS) outbreaks within a Network flow.

The techniques used in this study are to:

i. Acquire a dataset contains a wide range of DDoS outbreak occurrences (CICDDoS2019 dataset) in a network flow.
ii. carry out a data preprocessing on the acquired dataset so that it can be key-able into the deep learning model.
iii. Design a deep learning model for recognizing the DDoS outbreak.

iv. Evaluate the performance of the model using some selected evaluation metrics.

.

II. Related Works

The evaluation of deep learning in identifying strange network flow by Sabeel et al. [12] informed a machine learning models development, precisely a Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM), to detect a strange Distributed Denial of Service (DDoS) outbreaks. This techniques has to do with allowing the models to learn with preprocessed DoS/DDoS data from the CICIDS2017 dataset and testing them on imitating outbreaks from the ANTS2019 dataset. The models was also made to learn on a combined dataset (CICIDS2017+ANTS2019) and their performance were evaluated on new synthetic attacks. The outcomes showed a substantial advancement in accuracy after subjecting it to more training, with DNN achieving 98.72% accuracy and LSTM achieving 96.15%. Nevertheless, the work failed to consider real-time recognition as a result of this there is need to engage more in exploring more real-world application.

Virupakshar et al. [16] recommended an application for recognizing Distributed Denial of Service (DDoS) outbreaks for OpenStack-based Private Clouds. The motive was to identify link over saturating DDoS outbreaks in OpenStack clouds with the help of machine learning models, including K-Nearest Neighbors (KNN), Decision Trees (DT), Deep Neural Networks (DNN) and Naive Bayes (NB). The models was also evaluated on a dynamic dataset and DNN was found to have highest accuracy and precision (96% on cloud data). Nevertheless, it was discovered that the DNN Precision was lesser on the outdated KDDCUP99 dataset. Moreover, the work did not have facts on Cloud/LAN dataset and also failed to dig deep into the DNN's efficiency alongside wider outbreak forms.

Asad et al. [3] built a software named DeepDetect using a Deep Neural Network (DNN) that will be able to combat against application-layer DDoS outbreaks. They recommended a DeepDetect using a feedforward backpropagation architecture and testing it on the CICIDS2017 dataset for DDoS recognition. They sampled DeepDetect with Random Forest (RF) and DeepGFL algorithms, with DeepDetect to get an F1-score of 0.99 and high accuracy confirmed by an AUC value close to 1. The DeepDetect was set up as a cloud-based web service, concentrating wholly on application layer DDoS outbreaks but failed to explore its efficiency alongside other attack.

Muraleedharan and Janet [19] built a DL based HTTP slow DoS sorting technique on network flow. The purpose was to be able to recognize a slow DoS outbreaks on HTTP traffic through network flow and their model was trained using DoS samples from the CICIDS2017 dataset. The research got 99.61% accuracy in recognizing dissimilar dawdling DoS outbreak forms (Slowloris, SlowHTTP, Hulk, GoldenEye). Nevertheless, the assessment was narrowed to HTTP slow DoS outbreaks and necessitates trying it alongside broader DoS outbreak sorts and datasets.

Sbai and El Boukhari [13] intended to create a Data flooding intrusion detection system for Mobile Ad hoc Networks (MANETs) by means of a deep learning technique. The Deep Neural Network (DNN) was coached with two unseen strata on the CICDDoS2019 dataset and weighed its efficiency, attaining an accuracy (0.99), recall (1.0), F1-score (0.99) and precision (0.99), for data flooding outbreaks. However, the study only focused on data flooding/UDP flooding attacks within the CICDDoS2019 dataset and needs investigation into its effectiveness against other DDoS attack types.

Amaizu et al. [2] anticipated a combined and proficient DDoS outbreak recognition structure for 5G and B5G networks using Deep Learning (DL). Their system was able to merged two Deep Neural Network (DNN) models with a feature extraction algorithm (PCA) and attained 99.66% accuracy and 0.011 loss in DDoS outbreak recognition. Nevertheless, this combined technique can increase the time it will take to recognize outbreak thereby affecting the real-time enactment, and will need improvement for quick execution.

Hasan et al. [13] built a Deep Convolutional Neural Network (CNN) model to identify Burst Header Packet (BHP) flooding DDoS outbreaks in Optical Burst Switching (OBS) networks. This work anticipated a Deep CNN model for DDoS recognitions by means of a smaller dataset with little features and outclassed Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) in multi-class classification. Nevertheless, the dataset had little inadequate amount of occurrences and did not contain all imaginable outbreak patterns, necessitating a further ample dataset for vigorous assessment.

Amma and Subramanian [17] presented the VCDeepFL technique for recognizing DoS outbreaks, this has to do with a two-phase technique with pre-coaching with the help of unsupervised learning (Vector VCNN)

and coaching using supervised learning (FCNN). The VCDeepFL technique was assessed on the NSL KDD dataset and obtained a high accuracy, low false alarm, and enhanced discovery measure against the base classifiers (MLP, SVM) and state-of-the-art outbreak recognition methods. Nevertheless, the work failed to establish research for identifying anonymous outbreaks, and it also engaged an outdated dataset.

Lastly, Shaaban, Abd-Elwanis, and Hussein [15] proposed a Convolutional Neural Network (CNN) model for DDoS outbreak identification and grouping. The work intended to relate the results of their CNN model with existing sorting algorithms. The CNN model was assessed against Decision Trees (DT), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Neural Networks (NN) by means of two datasets: dataset 1 (simulated network traffic) and dataset 2 (NSL-KDD). Outstandingly, the CNN model attained an impressive 99% accuracy on both datasets. Their discoveries established the preeminence of the CNN model above DT, SVM, KNN, and NN classification algorithms in terms of accuracy, the work did not dig deep into the impending effect of one-column lining on model training, which possibly will actually affect the result of the CNN model (Wan, et al. 2021).

III. System Design

As explained in the introductory part, Distributed Denial of Service (DDoS) attacks are malicious attempts to interfere with a server, service, or network's regular operation by flooding the target or the infrastructure around it with an excessive amount of Internet traffic. Though, several researches and analysis have been carried out regarding the subject matter (DDoS Attacks) and its threats on security. At this point Deep learning-based approach is employed to detect and mitigate Distributed Denial of Service (DDoS) attacks within a Network.

a. Data acquisition

A publicly available benchmark dataset for Distributed Denial of Service (DDoS), was used as a secondary dataset. This dataset provides a wider range of attack types therefore providing all the necessary features for training and testing the model to detect DDoS attack.
Dataset Name: CICDDOS2019
Source: https://www.unb.ca/cic/datasets/ids.html
Description: The CICDDOS2019 dataset is a standard benchmark dataset for intrusion detection research, containing a variety of DDoS attack types simulated in a realistic network environment.

Feature extraction- this stage recognizes and describes the attributes mined from the network outbreak data that are appropriate for identifying Distributed Denial-of-Service (DDoS) outbreaks.
The classes of features used in detecting DDoS outbreaks in network flow are as follows:
Flow Features: These attributes seizure data around specific network traffic flows, which are source and destination IP addresses, packet size, number of packets, flow duration, and inter-arrival time between packets. Deviations from distinctive movement form can lead to DDoS outbreaks.
Packet Header Features: The attributes derived from packet headers comprises of protocol type (TCP, UDP, ICMP), source and destination port records, flags (SYN, ACK, FIN), and total length. Investigating these attributes can assist in recognizing apprehensive circulation occurrence accompanying by means of DDoS outbreaks.
Traffic Volume Features: These attributes centers on the whole bulk of network circulation, for instance the entire quantity of packets in a second, total bytes transported per second, and connection rate. Significant changes in these statistics can signal DDoS outbreaks.
Statistical Features: this features addresses the use of other tools. For instance mean, standard deviation, minimum, and maximum values of flow durations, packet sizes, and inter-arrival times. Significant changes in these statistics can signal DDoS activity..
Time-based Features: Features like time stamp, can be helpful in identifying duration of the attack patterns and potential sources of DDoS attacks.

b. Data Preprocessing

To ensure the quality and relevance of the data for training the model, preprocessing steps are performed on both the primary dataset from McPherson University and the CICDDOS2019 Dataset.

i. Handling duplicates and constant values
The initial step involved identifying and removing duplicate features and constant values from the dataset to ensure data quality and prevent redundant or non-informative variables from affecting model performance.

ii. Removing duplicates

Network issues or data collection errors can sometimes lead to duplicate entries. To address this, identical data points were identified using sorting technique and subsequently removed them from the datasets.

iii. Identifying quasi-constant features
Features with very low variance i.e., constant or nearly constant values are removed, as they provide little to no information for model training.

iv. Filtering low information gain features
Features with minimal impact on the target variable (below a threshold of 0.01) are eliminated to reduce noise and improve model performance.

v. Data transformation
To ensure compatibility with the chosen deep learning model, non-numeric features were transformed into numerical representations. This primarily involved one-hot encoding for categorical variables, where each category was mapped to a unique binary vector.

vi. Scaling and normalization
Different features in network traffic data often have varying scales. To address this, standard scaling was applied to the numeric features. This process ensures that each feature contributes equally to the model by normalizing the data to have a mean of 0 and a standard deviation of 1. This prevents features with larger scales from disproportionately influencing the model's learning process.

vii. Data splitting
To examine the efficiency of the model, the CICDDoS2019 dataset was fragmented into three different circles:
Training Set: the 60% of the entire data was used to coach the model.
Validation Set: about 20% of the dataset was used turn the model and prevent over-fitting.
Test Set: The 20% of the dataset was used to weigh the concluding performance of the model on hidden data.

viii. Data Augmentation
In other to have a balance dataset, precisely for the minority class (DDoS attacks), data augmentation was carried out with the help of Synthetic Minority Over-sampling Technique (SMOTE). The Synthetic Minority Oversampling Technique (SMOTE) is majorly acepted as a regular technique for treating unfair data disputes in machine learning. Its acceptance grows from its forthright application and efficient in numerous kinds of glitches.

The SMOTE algorithm always use a controlled approach to produce synthetic models for the minority class. Primarily, a known figure of oversampling occurrences, N is produced, which can have the goal for a well-adjusted class distribution or be dogged through a precise optimization procedure (Chawla et al., [14]. The process encompasses different recursive steps:

An indiscriminate occurrence from the minority sort in the training dataset is designated; The K nearest neighbors (typically 5) of this occurrence are recognized; In generating new synthetic models, N neighbors are indiscriminately selected out of these K occurrences; For every selected neighbor, the dissimilarity amid the feature vectors of the real occurrence and the neighbor is computed.

This dissimilarity is then scaled by an indiscriminate attributes between 0 and 1 and added to the original feature vector, generating a new point along the line segment connecting the original instance and its neighbor. For categorical attributes, one of the two possible values is selected at random.
SMOTE was applied to the minority class to create approximately 900 new synthetic instances of DDoS attacks, balancing the dataset. This augmentation ensured a more even representation of the minority class, reducing the risk of overfitting and enhancing the model's ability to accurately classify DDoS attacks during testing on new data.

b. Model Selection

After the data acquisition and data preprocessing which prepared the dataset gotten from kaggle repository useable for the design model three different model were selected which are CNN, GRU and DNN

i. Convolutional Neural Network (CNN)

CNN was chosen as the foundation of the framework because of its proficiency in recognizing patterns in data, especially within datasets that have high dimensionality as shown in figure 4.
The initial step involved defining the CNN architecture using the Keras library. The model comprised a sequence of layers: A convolutional layer with 64 filters, a kernel size of 3, and ReLU activation function, a max pooling layer with a pool size of 2 for down-sampling and reducing feature dimensionality. an additional convolutional layer with 128 filters, a kernel size of 3, and ReLU activation function, further extracting more complex patterns, max pooling layer with a pool size of 2 for further down-sampling, flattening layer to transform the extracted features into a 1-dimensional vector suitable for fully-connected layers, a dropout layer with a rate of 0.5 for regularization to prevent overfitting, two fully-connected layers: the first with 100 neurons and ReLU activation, and the final output layer with 1 neuron and sigmoid activation for binary classification (normal vs. attack traffic).
The figure 2: illustrates the CNN architecture.

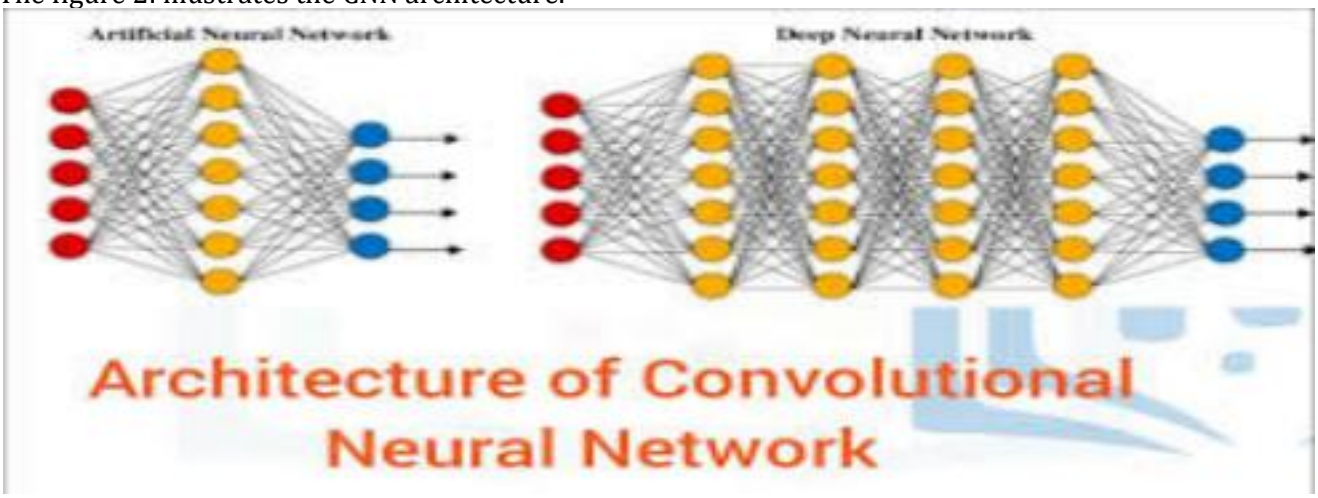


Figure 2: CNN architecture (Ramzan et al., 2023)

Gated recurrent unit (GRU) model
A GRU model was defined using Keras. Similar to LSTMs, GRUs are suitable for sequential data. It begins with an input layer configured to handle sequential data with a shape of
Xtrain.shape[1],1)(X_{train}.shape[1], 1)(Xtrain.shape[1],1), where
Xtrain.shape[1]X_{train}.shape[1]Xtrain.shape[1] represents the sequence length and each time step has one feature. Following the input layer, a GRU layer with 50 units and ReLU activation is employed to capture temporal dependencies and patterns within the sequential data. Lastly, a dense output layer with a single neuron and sigmoid activation function is utilized to produce probabilities.
ii. Deep neural network (DNN) model
The DNN model as shown in Fig 3.4.3, is characterized by its multiple layers of interconnected neurons. This model consists of three layers for binary classification. It starts with a dense layer of 64 neurons using ReLU activation, followed by dropout regularization (rate = 0.5). The second layer has 32 neurons with ReLU activation and another dropout layer (rate = 0.5). The final layer is a dense output layer with a single neuron and sigmoid activation, producing probabilities for binary classification tasks.

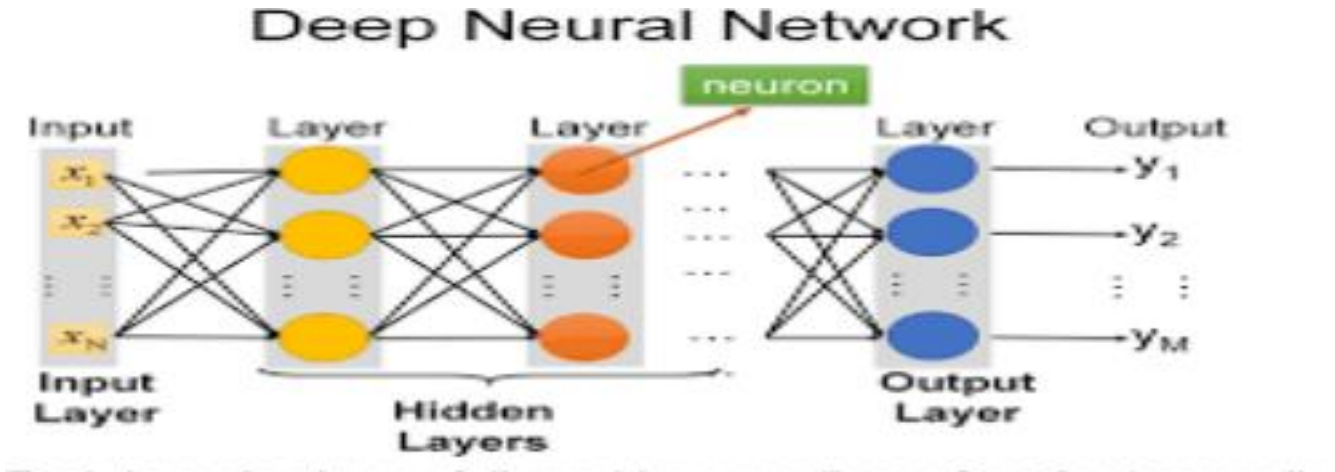


Figure 3: DNN Architecture (Kuma, 2020)

i. Hyperparameter training

In deep learning models, hyperparameter tuning has to do with identifying the best set of factors to improve network enactment and efficiency. This procedure has to do with methodically analyzing various hyperparameter figures or classes, coaching and assessing the network for each configuration, and choosing the set of hyperparameters that produced the best result on a endorsement set or through cross-validation. The precise figure of these factors can differ subject to the requirements and dataset. The configuration parameters used for model training are presented in Table 1

ii. Learning rate

The learning rate parameter explicate the step size for each repetition as the model reaches the minimum of the loss function. Recognizing the best learning speed necessitates investigating by means of numerous measure. The work used the Adaptive Moment Estimation (Adam) technique to decide the learning speed for the models, attaining a learning speed of 0.001 provided the best optimization [5].

iii. Activation functions

This study utilized the Rectified Linear Unit (ReLU) activation function. The ReLU function enabled the model to learn complex features within the network's hidden layers. Compared to other activation functions such as sigmoid and tanh, ReLU demonstrated greater efficiency in this context [6]

Table 1:. Model Parameterizatio

Parameter	CNN	DNN	GRU
Input Shape	(60, 53, 1)	(53,)	(60, 53)
Number of Layers	2 Conv, 1 Dense	4 Dense	2 GRU, 1 Dense
Units/Filters	Conv: 32, 64 Dense: 128	Dense: 128, 64, 32, 16	GRU: 50, 100 Dense: 128
Filter Size	3x3	N/A	N/A
Pooling Layers	MaxPooling (2x2)	N/A	N/A
Dropout Rate	0.5	0.5	0.5
Activation Function	ReLU	ReLU	ReLU
Epochs	10	10	10
Number of Batch Size	32	32	32

iv. Early Stopping

Early stopping is a technique where the training of the model halts when its performance does not improve after a predetermined number of epochs. This method tracks the validation loss, with a minimum change threshold of 0.001. If the validation loss fails to decrease by at least
0.001 over five consecutive epochs, he training process terminates early. [7]

v. Optimizers

The Adam optimizer is an optimization algorithm that combines the RMSprop and AdaGrad techniques. It adjusts the learning rates based on the first and second moments of the gradients, effectively preserving learning rates for each parameter. By dynamically altering the learning rate during training, the Adam optimizer efficiently updates the model's weights [6]

v. Batch size

Batch size refers to the number of training samples the model processes in each iteration during training. Research indicates that larger batch sizes result in more stable gradients and training models, whereas smaller batch sizes can lead to faster training but with less stability and accuracy. Batch sizes generally start at 32 and can go higher. In this study, experiments were conducted with a batch size of 32. [7] The defined models were then compiled using the Adam optimizer, binary cross-entropy loss function, and accuracy metric for evaluating performance during training. Finally, the training process started. The preprocessed training data (X_train) were reshaped to incorporate the time dimension (samples, features, channels) for compatibility with each model. The model was trained on the reshaped X_train data along with the corresponding labels (y_train). A validation set (X_val, y_val) was used to monitor the model's performance while going through coaching and reducing overfitting. The coaching progression loped for 10 RFM epochs, by means of a set size of 32 models processed at a time. This iterative process allowed the models to learn the underlying patterns in the training data and adjust its internal parameters to improve classification accuracy. Figure 5 shows the overall flow of the methodology.

D. Model Evaluation

Various metrics were used to evaluate the effectiveness of the model such as test accuracy, precision, F1 score, recall and also confusion matrix shown in Table 2

i. Accuracy

This measures the overall correctness of the model in predicting both classes (DDoS and non-DDoS).

Formula: Accuracy = $\frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$

ii. Precision

This indicates the proportion of true DDoS attacks among the instances predicted as DDoS.

Formula: Precision = $\frac{True\ Positives}{True\ Positives+False\ Positives}$

iii. Recall

This measures the proportion of actual DDoS attacks that were correctly predicted. Formula:

Recall = $\frac{True\ Positives}{True\ Positives+False\ Neggatives}$

iv. F1-Score

Harmonic mean of precision and recall, providing a single metric to evaluate model performance.

Formula: F1-Score = $2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall}$

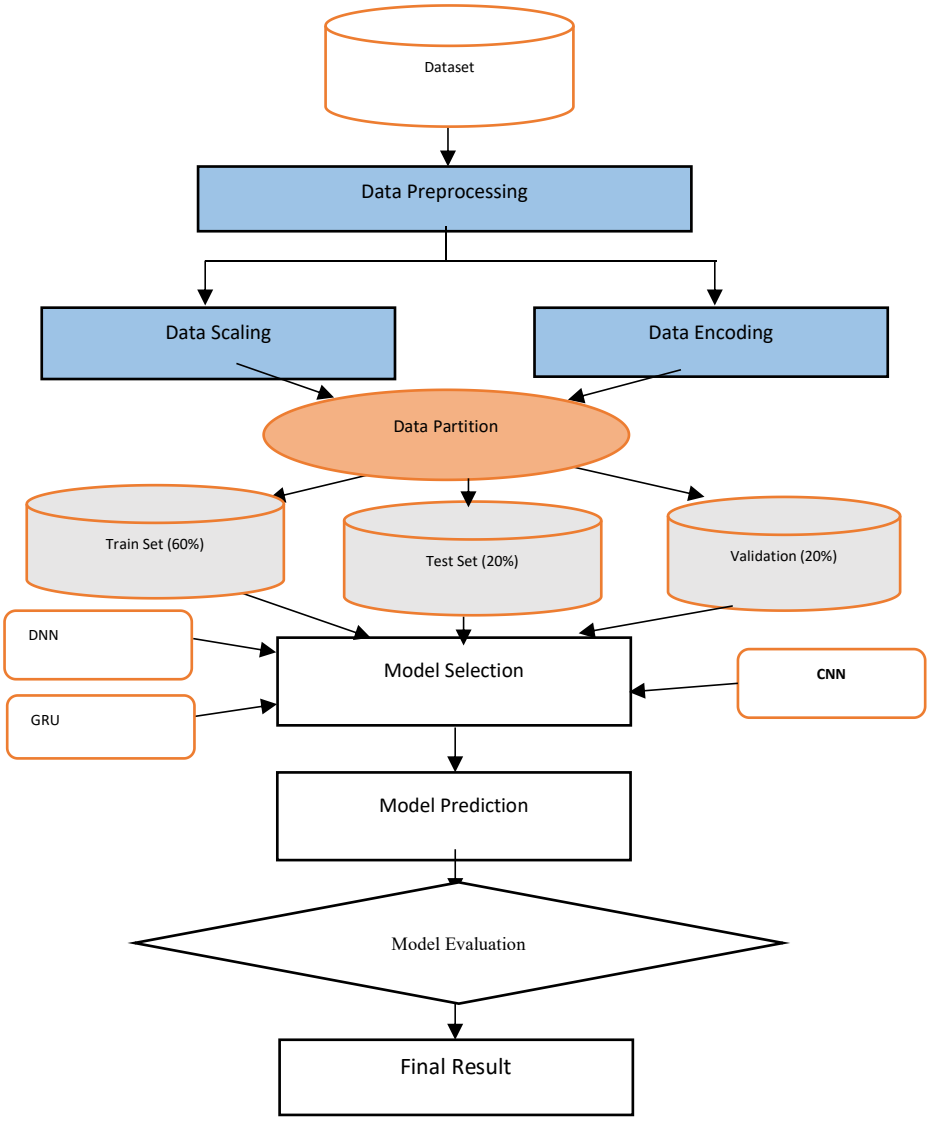


Figure 4: Methodology flow diagram

Figure 4 shows the flow diagram of the methodology describing various steps by step process taken in designing a deep learning machine model for detecting and mitigating DDoS attack. The flow diagram starts with dataset, and the dataset used in this study is a secondary dataset. This dataset was preprocessed so that it can be useable for training in the designed model. The data preprocessing includes data encoding and data scaling. After data preprocessing the data was splited into two categories 80% for training and

20% for testing; after which the training was implemented using three different deep learning models namely: CNN, GRU and DNN, for predictions. The quality of the prediction was evaluated using some selected matrix's which are Accuracy, Precision, Recall and F1 Score.

IV. Results

Evaluation metrics were collected via the confusion matrix as shown in Figure 5. The parameters of the confusion matrix are True Positive (TP), which denotes accurately recognized malicious traffic, and True Negative (TN), which denotes correctly identified benign traffic. False
Positive (FP) denotes malicious traffic that is incorrectly recognized as benign traffic, and False Negative (FN) denotes benign traffic that is mistakenly identified as malicious traffic.

a. Confusion Matrix

A confusion matrix is a table that provides a summary of a machine learning model's performance on a test dataset. It shows the number of correct and incorrect predictions made by the model. This matrix is frequently used to evaluate the effectiveness of classification models, which predict categorical labels for each input instance. (AlSalel et al., 2024)

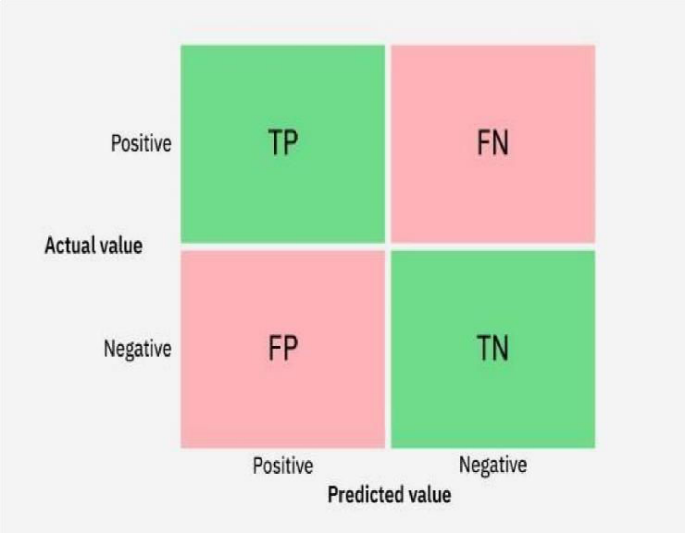


Figure 5: Confusion Matrix

b. System Requirements, Tools, Software libraries and Frameworks

Hardware Requirements
Processor: Hp Spectre Intel Core i7-11800H CPU @ 2.50GHz
RAM: 16 GB DDR4
Storage: 1 TB SSD
Graphics: NVIDIA GeForce RTX 3060 (6GB VRAM) — utilized for GPU-accelerated deep learning training
Operating System: Windows 11 / Kali OS, 20.04 LTS (dual boot)

Software Requirements
Programming Language used: Python 3.10
Deep Learning Framework: TensorFlow 2.10 & Keras
Libraries and Tools:
- Matplotlib / Seaborn (was used for picturing)
- Keras (for high-level neural network API)
- Pandas & NumPy- was used for data management, analysis and numerical computing
- Scikit-learn was used for data preprocessing in this work

Jupyter Notebook / Google Colab / VS Code: was the interactive development environment (IDE) used for running the python code and for documentation
Anaconda: was used for environment and package management
CUDA Toolkit 11.7: For enabling GPU support during training

c. Model Evaluation

This research work was carried out using the CICDDoS2019 dataset and it gives a very positive outcomes for binary classification DDoS detection using CNN, DNN, and GRU respectively.

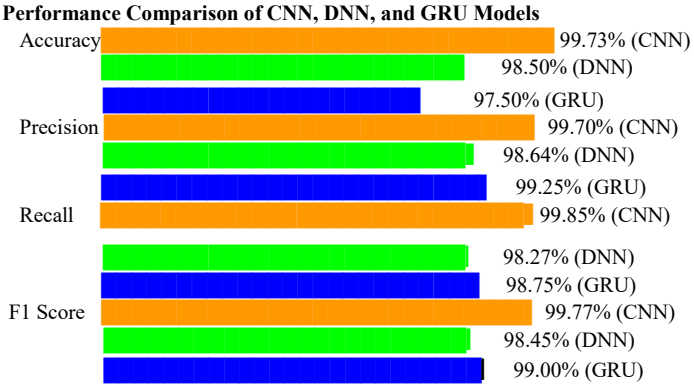


Figure 6: Performance Comparison of CNN, DNN, and GRU Models

The Convolutional Neural Network (CNN) appears as the groundbreaker between the three models used, with the highest scores in all metrics except Precision, where GRU clasps a slim lead. The CNN gives a remarkable accuracy of 99.73%, signifying a very small error rate in categorizing both DDoS attacks and benign traffic. Which is additionally sustained by the high Precision (99.70%) and Recall (99.85%) values, signifying the model efficiently reduces mutually false positives (benign traffic classified as attacks) and false negatives (attacks classified as benign). The F1 score of 99.77% encapsulates this robust complete performance.

Table 2: Performance results for the classification

Performance Measure	CNN	DNN	GRU
Accuracy	99.73%	98.50%	97.50%
Precision	99.70%	98.64%	99.25%
Recall	99.85%	98.27%	8.75%
F1 Score	99.77%	98.45%	99.00%

The Gated Recurrent Unit (GRU) model is the model closely behind the CNN with a reputable accuracy of 97.50%, and its Precision (99.25%) is a bit higher than the CNN, signifying that it yields faintly rarer false positives. Nevertheless, the Recall (98.75%) is a bit lower, indicating that GRU may fail to identify a more actual DDoS occurrences when placed side by side with the CNN. The F1 score of 99.00% shows the trade-off amid precision and recall.

The Deep Neural Network (DNN) exhibits the lowest performance among the three models.
While its accuracy (98.50%) is still good, it falls short of the CNN and GRU. Additionally, both Precision (98.64%) and Recall (98.27%) are lower, suggesting the DNN struggles to correctly identify both DDoS attacks and benign traffic to the same extent as the other models. The F1 score of 98.45% reflects this overall weaker performance.

The CNN confusion matrix as shown in figure 7, indicates a very high number of True Positives (TP) and True Negatives (TN), signifying accurate identification of both attack and benign traffic. The minimal False Positives (FP) and False Negatives (FN) further solidify the CNN's effectiveness.

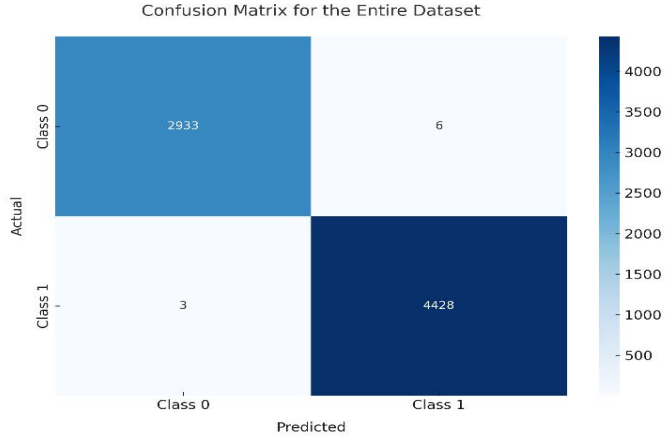


Figure 7: Confusion Matrix for CNN

The GRU confusion matrix as indicated in figure 8 shows a similar pattern to the CNN with a high TP and TN. However, the presence of a slightly higher FN (37) compared to the CNN suggests the GRU might miss a few more DDoS attacks.

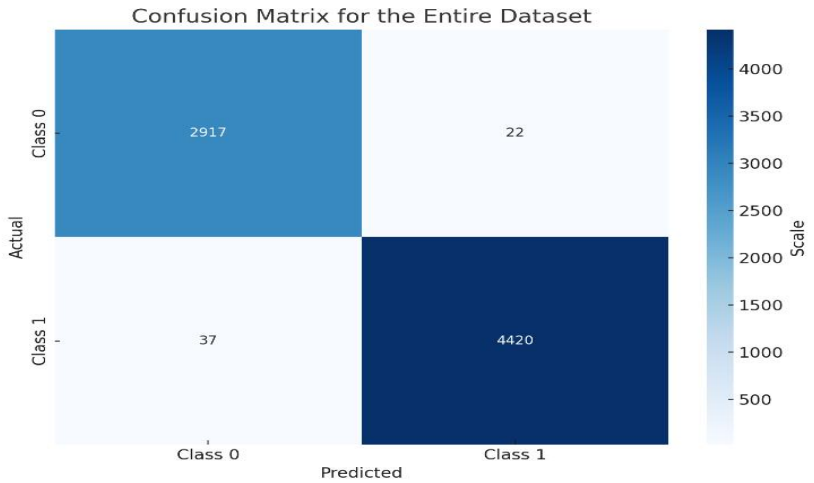


Figure 8: Confusion Matrix for GRU

The DNN's confusion matrix as shown in figure 9 reveals a lower TP and TN compared to the other models, indicating a higher number of misclassifications. The higher number of FPs (40) suggests the DNN might incorrectly classify some benign traffic as attacks. Additionally, the higher FN (51) implies the DNN misses a substantial number of actual DDoS attacks.

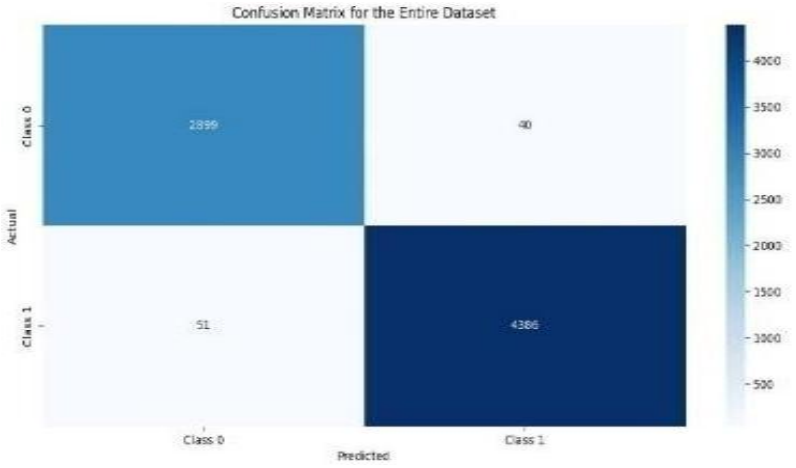
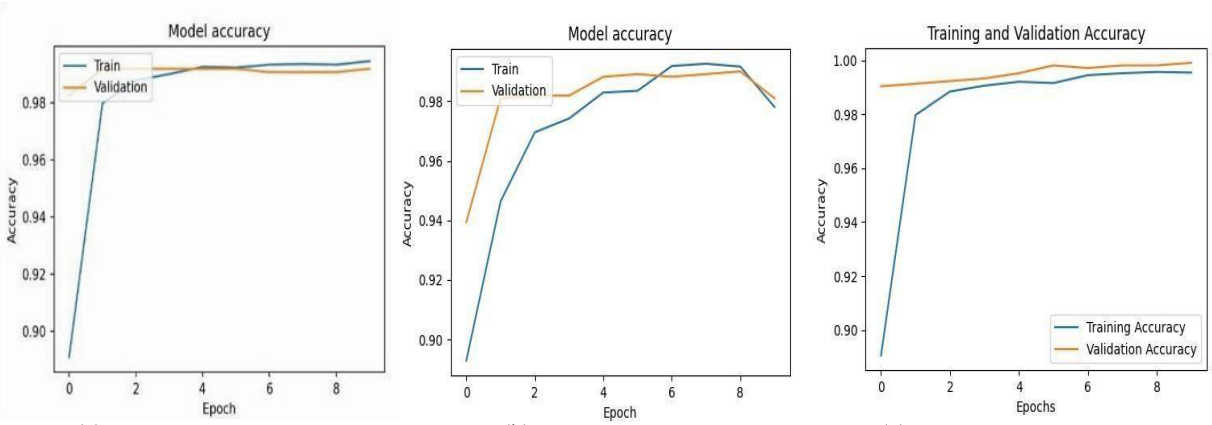


Figure 9: Confusion Matrix for DNN

Since the CNN model demonstrated the highest performance metrics, including precision, recall, and F1 score, it will be utilized to test and validate the McPherson University dataset. This choice ensures that the most effective model is employed to achieve accurate classification and reliable detection of DDoS attacks within the university's network traffic data. The Figure 10 below displays the Model Accuracy Graph for CNN, GRU and DNN.



(a) CNN Accuracy (b) GRU Accuracy (c) DNN Accuracy

Figure 10: Model Accuracy Graphs of (a) CNN (b) GRU (c) DNN

Table 3: Evaluation Table with Existing System Results

Study / Model	Dataset(s)	Accuracy (%)	F1-Score	Key Strengths	Limitations
Proposed Model	CICDS2019 Extended Dataset	99.85	0.985	High accuracy; robust multi-class detection	Real-time evaluation and efficiency yet to be explored fully
Sabeel et al. [12]	CICIDS2017, ANTS2019	98.72 (DNN), 96.15 (LSTM)	N/A	Multi-dataset training; synthetic attacks	No real-time detection tested
Virupakshar et al. [16]	OpenStack Cloud Data, KDDCUP99	96.00 (DNN)	N/A	Applied on dynamic OpenStack environment	Weak results on outdated dataset; limited attack diversity
Asad et al. [3]	CICIDS2017	High (AUC ≈ 1)	0.99	Cloud deployment; compared with RF, DeepGFL	Focused only on application-layer DDoS
Muraleedharan & Janet [19]	CICIDS2017	99.61	N/A	High accuracy on HTTP slow DoS variants	Narrow scope; lacks attack variety
Sbai & El Boukhari [13]	CICDDoS2019	99.00	0.99	Strong metrics for UDP flooding	Only evaluated on data flooding attacks

Validation on McPherson University Dataset

The effectiveness of the Convolutional Neural Network (CNN) is validated on the enhanced McPherson University network traffic, aiming to demonstrate the model's high performance in accurately detecting Distributed Denial of Service (DDoS) attacks. To address the class imbalance present, we implemented the Synthetic Minority Over-sampling Technique (SMOTE). This technique was used to generate 990 additional instances of DDoS attacks, thus balancing the dataset.

The model shows excellent performance with a test accuracy of 98.68%, indicating it correctly classified nearly 99% of samples. It's precision of 0.96 means that 96% of positive predictions were accurate, and the F1 Score of 0.98 highlights a strong balance between precision and recall. The confusion matrix values in Figure 11 (1601 TP, 40 FP) and (5 FN, 1005 TN) reveals that the model has a high number of true positives and true negatives, with a very low false negative rate, suggesting few missed positive cases. The quick training time of 4 milliseconds per step further underscores the model's efficiency. The CNN model's performance on the McPherson University dataset is very similar to its performance on the CICDDOS2019 dataset, indicating that the model generalizes well across different datasets. Overall, the metrics reflect a highly effective model.

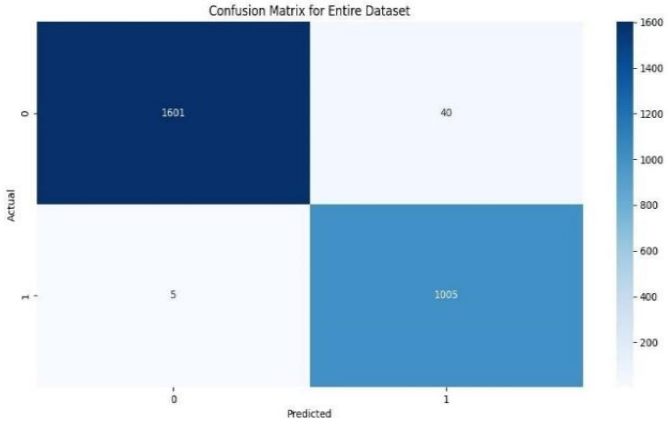


Figure 11 Confusion Matrix of the Validation

V. Discussion

In this work, deep learning techniques were analyzed to see how well it can detect Distributed Denial-of Service (DDoS) attacks on the McPherson University network. Accurately identifying DDoS attacks amidst normal network traffic remains a complex task. While researchers have developed effective deep learning methods for DDoS detection, these methods often struggle to adapt to the constantly evolving tactics of attackers. Attackers are unceasingly developing new approaches and initiating unique, never-before-seen (zero-day) outbreaks with diverse circulating methods, making surviving detection technique ineffective. (Mittal et al., 2022)

The objective of this work was to know if deep learning models could assist recognizing and lessen DDoS outbreaks in McPherson University network. While the specifics purposes were:

1. To develop and train a deep learning model using the CICDDOS2019 dataset to identify DDoS outbreaks and then test its effectiveness on McPherson University network.
2. To evaluate the model's efficiency by determining its accuracy, precision, recall, and F1score on the McPherson University network data. The key phase in this work is acquiring data, preprocessing it, choosing applicable factors, training the models, and evaluating their performance. The outcomes revealed that deep learning approaches can effectively differentiate amid usual and malicious network activities.

The three models created produced an encouraging results, attaining considerable accuracy, precision, and recall metrics. Despite the delays in getting the dataset and the inadequate attribute within the dataset, the deep learning approach proved instrumental in identifying patterns that indicates DDoS outbreaks inside multifaceted network movement.

a. Contribution to Cybersecurity

This research work discussed the application of deep learning for DDoS recognition, a moderately novel and encouraging technique in the field of cybersecurity. Studying more and assessing the efficiency of deep learning models, this work contributes appreciated knowledge and improvements to the larger body of knowledge in DDoS mitigation techniques.

The efficacious application of a deep learning model for DDoS identification in McPherson University can serve as a appreciated case study for other institutions of learning and administrations facing related cybersecurity problems. This work creates room for the broader acceptance of deep learning approach in network safety, eventually producing a saver and strong digital environment.

b. Conclusion

The study recommends that applying this new development can significantly advance the university's cybersecurity strenght. Future work may involve tunung the model, applying it in real-world situations, and evaluating its performance in alleviating DDoS attack through different network environments. This work may serve as a benchmack for McPherson University to improve its digital security organization through state-of-the-art technique.

c. Recommendations to McPherson University ICT

The outcome of this research work depict that McPherson University ICT can take the following practical approach to improve its cybersecurity against DDoS occurrences.

i. Regular Model Updates

They must ensure a regular updates of the deep learning model using newer data. This approach will aid the model to be more active in identifying emerging DDoS outbreak patterns and adapting to variations in network performance at any period of time.

ii. Explore Hybrid Models:

They can source for Hybrid models that can combine deep learning with other techniques. Incorporating deep learning with traditional statistical techniques or rule-based systems can hypothetically widen the model's proficiencies. This method will increase the model accuracy in identifying multifaceted DDoS outbreak patterns and reducing false positives.

d. Future Works

Looking at the future advances, the plan was to improve the efficiency and real-world application of deep learning in recognizing DDoS outbreaks, eventually consolidation cybersecurity status through different segment and situations;

i. Addressing Evolving Attack Techniques - Adversarial Learning

Adversarial learning provides a encouraging approach in other to move with the ever-changing techniques of DDoS outbreaks, This approach has to do with training deep learning models to recognize and change to new outbreak methods produce by malicious occurrences; this is done by repeatedly subjecting the model to mimicked outbreak instances, adversarial learning make stronger the capability to develop systems that can work with multifarious and sprouting DDoS attack.

ii. Scaling and Deployment

Also, it is necessary to make sure that the DDoS recognition models would be able to efficiently weighbridge larger networks within an organizational setup. This has to do with trying the models in experimental projects and before deploying them for use in other environs. This will assist one to know how thriving they can work in changing situations and network oodles.

iii. Collaborative Defense Mechanisms - Federated Learning:

Federated learning gives a collective method to improving DDoS resistance through circulated networks. Not like the existing central approaches, federated learning enables each network connectors to individually train a Deep Learning models through their personal data; where merely there accumulated updates out of these models will be shared on the central server, protection data confidentiality. This distributed approach assists various networks to communally optimize model accuracy and compliance without degrading profound information, in so doing improving the efficiency of DDoS recognition and alleviation effect.

References

[1] I. AlSaleh, A. Al-Samawi, and L. Nissirat, “Novel machine learning approach for DDoS cloud detection: Bayesian-based CNN and data fusion enhancements,” *Sensors*, vol. 24, no. 5, p. 1418, 2024. [Online]. Available: https://doi.org/10.3390/s24051418

[2] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, “Composite and efficient DDoS attack detection framework for B5G networks,” *Computer Networks*, vol. 188, p. 107871, 2021. [Online]. Available: https://doi.org/10.1016/j.comnet.2021.107871

[3] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, “DeepDetect: Detection of distributed denial of service attacks using deep learning,” *The Computer Journal*, 2019. [Online]. Available: https://doi.org/10.1093/comjnl/bxz064

[4] M. Badsha, “A beginner’s guide to understanding DDoS attacks & how to protect your site,” *Host SEO*, Jul. 4, 2019. [Online]. Available: https://blog.hostseo.com/a-beginners-guide-to-understanding-ddosattackshow-to-protect-your-site/

[5] J. Brownlee, Understand the dynamics of learning rate on deep learning neural networks. Machine Learning Mastery, 2020. [Online]. Available: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deeplearning-neural-networks/

[6] J. Brownlee, Gentle introduction to the Adam optimization algorithm for deep learning. Machine Learning Mastery, Jan. 13, 2021. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[7] J. Brownlee, “Difference between a batch and an epoch in a neural network,” Machine Learning Mastery, Aug. 15, 2022. [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/. [Accessed: Apr. 4, 2023].Cil, A. E.,

[8] K. Yildiz and A. Buldu, “Detection of DDoS attacks with feed forward based deep neural network model,” Expert Systems with Applications, vol. 169, p. 114520, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.114520

[9] K. Some, “The history, evolution and growth of deep learning,” Analytics Insight, Oct. 30, 2018. [Online]. Available: https://www.analyticsinsight.net/deep-learning/the-history-evolution-andgrowth-of-deep-learning

[10] A. Kumar, “Deep learning explained in layman’s terms,” DZone, Oct. 8, 2020. [Online]. Available: https://dzone.com/articles/deep-learning-explained-simply-in-layman-terms-dat

[11] M. Mittal, K. Kumar, and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” Soft Computing, 2022. [Online]. Available: https://doi.org/10.1007/s00500-021-06608-1

[12] M. Mittal, K. Kumar, and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” Soft Computing, vol. 27, no. 18, pp. 13039–13075, 2022. [Online]. Available: https://doi.org/10.1007/s00500-021-06608-1

[13] O. Sbai and M. El Boukhari, “Data flooding intrusion detection system for MANETs using deep learning approach,” Proceedings of the 2020 International Conference on Computing, Control, Networking, Electronics, and Embedded Systems Engineering (ICCCNEE), 2020. [Online]. Available: https://doi.org/10.1145/3419604.3419777

[14] M. Ramzan, M. Shoaib, A. Altaf, S. Arshad, F. Iqbal, N. K. Castilla, and I. Ashraf, “Distributed denial of service attack detection in network traffic using deep learning algorithm,” Sensors, vol. 23, no. 20, p. 8642, 2023. [Online]. Available: https://doi.org/10.3390/s23208642

[15] A. R. Shaaban, E. Abd-Elwanis, and M. Hussein, “DDoS attack detection and classification via convolutional neural network (CNN),” IEEE Xplore, Dec. 1, 2019. [Online]. Available: https://doi.org/10.1109/ICICIS46948.2019.9014826

[16] K. B. Virupakshar, M. Asundi, K. Channal, P. Shettar, S. Patil, and D. G. Narayan, "Distributed denial of service (DDoS) attacks detection system for OpenStack-based private cloud," Procedia Computer Science, vol. 167, pp. 2297–2307, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.03.282

[17] K. W. Wan, C. H. Wong, H. F. Ip, D. Fan, P. L. Yuen, H. Y. Fong, and M. Ying, "Evaluation of the performance of traditional machine learning algorithms, convolutional neural network and Auto ML Vision in ultrasound breast lesions classification: a comparative study," Quantitative Imaging in Medicine and Surgery, vol. 11, no. 4, p. 1381, 2021.

[18] A. Wilson and M. R. Anwar, "The future of adaptive machine learning algorithms in high-dimensional data processing," International Transactions on Artificial Intelligence, vol. 3, no. 1, pp. 97–107, 2024.

[19] C. Muraleedharan and B. Janet, "Deep learning-based classification of HTTP slow DoS attacks using flow data," Procedia Computer Science, vol. 171, pp. 1126–1135, 2