Asian Journal of Research in Computer Science

Volume 17, Issue 4, Page 19-43, 2024; Article no.AJRCOS.112853
ISSN: 2581-8260

Hybrid Approach to Classification of DDoS Attacks on a Computer Network Infrastructure

Enock Quansah Effah [a*], Eric Opoku Osei [a], Maxwell Dorgbefu Jnr. [b] and Abraham Tetteh [c]

[a] Department of Computer Science, KNUST, Kumasi, Ghana.
[b] Information Technology Education Department, AAMUSTED, Kumasi, Ghana.
[c] Mathematics/ICT Department, Bia Lamplighter College of Education, Debiso, Ghana.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2024/v17i4428

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/112853

Original Research Article

Received: 01/12/2023
Accepted: 06/02/2024
Published: 12/02/2024

ABSTRACT

The advancement in technology, its ease of use, and the competitive nature of its deployment in business operations have led to the wide spread of networking systems globally, and Ghana is not an exception. Most business operations and even personal activities are now conducted online leading to increased network connectivity, access to networked resources, and the corresponding cyber-attacks on these network systems. Distributed Denial-of-Service (DDoS) is one of the sophisticated attacks in the cyberspace. In DDOs, the attacker floods the network with massive and unsolicited traffic, causing the network infrastructure to exhaust all its resources in responding to the attacker's request, thereby denying access to legitimate users of such resources. In this study, we designed and implemented a hybrid deep learning model (CRNN-Infusion) for detection and classification of DDoS attacks. Our model utilized the CNN, and RNN models, with the

**Corresponding author: E-mail: songreen18@gmail.com;*

CICDDoS2019 dataset obtained from the Canadian Institute of Cybersecurity (CIC) for its training, with Random Search Hyperparameter Tuning (RSHT) and Feature Selection (FS) techniques for model efficiency and dimensionality reduction. Cybersecurity (CIC) for the model's training, with Random Search Hyperparameter Tuning (RSHT) and FS techniques for model efficiency and dimensionality reduction. The results showed that, our proposed model is a better classifier for DDoS attacks compared to other deep learning (DL) models trained on the same dataset. With the highest accuracy of 98.92%, hybrid deep learning models are suitable for detecting and classifying DDoS attacks on network infrastructures. The findings point out that, with the appropriate choice of feature selection and hyperparameter tuning techniques, hybrid deep learning models perform optimally, with 98.92% accuracy, 99.02% precision, 98.92% recall, and 98.93% F1 score for our proposed model.

Keywords: Convolutional neural network; recurrent neural network; deep neural network; random search hyperparameter tuning.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

CNN:	*Convolutional Neural Network*
CRNN-Infusion:	*Convolutional Recurrent Neural Network Infusion (Proposed Model)*
DL:	*Deep Learning*
DNN:	*Deep Neural Network*
RNN:	*Recurrent Neural Network*
RSHT:	*Random Search Hyperparameter Tuning*

1. INTRODUCTION

With the global wave of digital transformation, communication and information sharing has been the order of the day ranging from individuals using devices to connect to one another, and businesses providing platforms for digital resource sharing among employees and partners. These communication platforms are powered by networks, connecting to servers and network resources to respond to legitimate users' request. As a result, attackers attack these network infrastructures for their personal and monetary gains. Majority of the attacks that are sent to disrupt the free and efficient functioning of these systems are DDoS attacks. DDoS attack intent is to prevent legitimate users of an internet infrastructure from accessing such services [1]. The attackers use several sources that are controlled to generate heavy amount of packet or traffic flow that overwhelms the requested service or system depleting it of the necessary resources to respond to legitimate users. This is because, the service, trying to respond to the several requests leaves it overburdened and malfunctioned causing the denial-of-service to other users. These attacks are usually controlled by instructions from humans (bots and botnets), which are devices that are compromised by malware infection [2]. Cyberattacks, such as DDoS attack is on the increase as more and more sophisticated means of accessing information from the internet is also on the rise [3]. This is the case as some attackers use this as a source for extorting money or merely a disruption of service. The use of networks to enhance business transactions is gaining root as more devices are deployed on the internet for various institutions and stakeholders to have access to readily information anywhere, anytime, over the internet. Ghana, a developing country is investing much into building ICT infrastructure and data centres to make it one of the technological hubs in the sub-Saharan Africa. According to World Bank [4], Ghana's firm's upgrade and the creation of new jobs is to be fostered through digital technology. This technological equipment relies on network for effective communication and functioning. It is therefore expedient to devise ways to respond against these attacks on networks to make them available to the rightful users. Various techniques have been adopted by different researchers to address this menace, ranging from traditional techniques to machine learning techniques as-well-as deep learning techniques. Even though there have been several studies which have delve into the ways of addressing the issue of DDoS attacks, it is still on the rise and needs more critical look into how other efficient models can be deployed to help classify and mitigate against DDoS attacks on a network. Machine leaning models have been deployed by different studies and have yielded good results yet lacks proper implementation plan on live systems. Several machine learning models deployed in this respect include that of Arshi et al. [5], Perez-Diaz et al. [6], and [7], using machine learning algorithms such as Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbour (KNN) and other models to detect DDoS attacks.

Other researchers investigated using deep learning techniques which provides for the extraction of more complex patterns from data to detect if a network traffic is an attack or normal. Examples are studies conducted by Khempetch et al. [8], Lopes et al. [9], and [10] which deployed models such as Dense Neural Network (DNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN) trained on a particular dataset, compare accuracies and rate of detection. From the various models deployed by the various researchers, there is still more to be done in DDoS attacks classification and mitigation as more and more sophisticated attacks are deployed by attackers every day. Nonetheless, few researchers have proposed the use of hybrid model that utilizes more than one DL technique in the detection of DDoS attacks. Examples of such papers are the study by Salmi and Oughdir [10] and Sumathi et al. [11] by combining CNN and LSTM in their study. These studies failed to apply feature selection techniques to include only important features to train the model which could yield much higher accuracy compared to what was achieved. The available dataset utilised in training most of these models do not contain balanced sets in terms of type of attacks at the period of dataset extraction. This paper suggested a model, utilizing a hybrid technique, CNN and RNN with enhanced feature selection and engineering, trained and tested on diverse dataset generated within a prolonged period, to include both old and novel attack types for the classification of DDoS attacks on a network. The study aimed to achieve an optimised performance in the classification of DDoS attacks.

The prevalent nature of DDoS attacks has shifted many researchers' attention into proposing detection and mitigation to these attacks. Filho et al. [12] in their study, utilised signature-based approach to detect DDoS occurrences with traffic extracted from network to make inference and classify it as attack or normal. Their proposed model, "Smart Detection" relied on DDoS signatures for attacks identification on the network. The model's compatibility with existing networks infrastructure guaranteed privacy and early identification of attacks on the network. Their proposed model outperformed other models with a detection rate above 96% and Random Forest achieving a best accuracy of 0.9996% outperforming algorithms such as Decision Tree, AdaBoost and SGD. In a similar study which proposed DDoS attack detection to address limitations in existing detection systems is the work of Pei et al. [13]. They employed the use of feature extraction and trained the model on classifiers such as RF and SVM. RF achieved a higher accuracy with low computation. Performance metrics measured are FP rate, and detection rate. Gupta et al. [14] proposed a framework to reduce dimension in data to enhance the efficiency and accuracy of classifiers of DDoS attacks. The model applied dimensionality reduction techniques to select important attributes that important to DDoS attack identification. The algorithm posed a challenge of processing time, in producing features been the reduced version of the dataset utilised in modelling process. The model proved evidence of efficiency and effectiveness in DDoS classification, and evaluated on precision, recall, false positive, false negative and accuracy. In this study too, RF outperformed other algorithms, obtaining an accuracy rate of 0.99979 on sixty (60) features of the dataset. Ahmed et al. [15] also experimented the use of classifiers as DT, KNN, LR and RF for DDoS attack identification and prevention. This algorithm also proposed a feature importance technique to include only the flow in a traffic that has a better correlation in attack detection. The model achieved between 0.993 – 0.999979 for accuracy. The RF algorithm achieved the best accuracy in this report. According to Malliga et al. [16] in their review of DDoS attack detection and mitigation using DL algorithms, discussed various deep learning techniques utilised in this area such as CNN, RNN, Autoencoder, MLP and datasets that contains various attack types. The study stressed the need to strengthen existing approaches and address data imbalances, which provided the extent to which the fight against DDoS attacks is yielding results. This is just a review and only points out what others have done in DDoS attacks detection and mitigation and do not put out any new study or model for such purpose. Accuracy, precision, recall, F1-score are the performance metrics discussed by this study. In a similar study by Ismail et al. [17], they proposed a model to classify DDoS attacks using

Random Forest and XGBoost techniques. The methodology depended on UNWSP-un-15 dataset, with feature extraction and label encoding and trained and optimised with the RF and XGBoost techniques. The study reported RF as the best predictor which is 100 times faster than other techniques. The accuracy record by the RF technique was 90% which outperformed other models compared in the study. The report did not provide enough information about the composition of the dataset used and makes it very difficult generalizing the findings of this study. In the same way, [18], in their paper proposed a model that dynamically calculate the thresholds of different applications in real-time and using ML to address the challenges of DDoS attacks in SDN environment did not provide any specific values in terms of performance measurement, but only stated that the model achieved a higher performance with the threshold calculation and prediction. The methodology used involves the collection of real-time data and generated threshold value based on the data, the Random Forest Regression is built to predict the threshold value of incoming data and compares the predicted threshold with statistical counts to identify and block DDoS attacks. In this model, the focus is on blocking only the attack traffic which helps to reduce the disruption to legitimate traffic.

Alduailij et al. [19] in their study focused on reducing misclassification error in the DDoS attacks identification in cloud computing. The model used a feature selection technique which are Mutual Information and Random Forest Feature Importance to select the most important features of the dataset, trained and tested using RF, GB, WVE, KNN and LR to identify DDoS attacks. The model achieved a better classification accuracy with reduced misclassification compared to existing models. The use of the feature selection method proved it worth the effort. The Random Forest classifier technique performance outperformed other in terms of accuracy with 0.99993 when 16 features were used, and 0.999977 when 19 features were utilized. Almaraz-Rivera et al. [20] on the other hand, proposed a novel IDS model for the identification and categorisation of DDoS attacks in IoT networks. The model utilized Bot-IoT dataset which addressed the class imbalance problem and using three different feature sets to train ML and DL for classification. The machine learning techniques, DT and RF outperformed other state-of-the-art IDS achieving over 99% in terms of accuracy. This goes to assert the claim made by Ulemale [21] that many state-of-the-art techniques have yielded a positive result in DDoS attack detection. The study reviewed several articles that utilised ML techniques for DDoS attacks detection and compared the performance with other models, emphasizing on the importance of feature selection and ensemble methods. The study only made available findings from several machine and deep learning models and concluded that, machine and deep learning can provide solutions to DDoS attacks, since they showed a greater level of performance in classification metrics. The models reviewed had accuracies between 97.86% to 99.99%.

Decision Tree (DT) has been mentioned by several studies as one of the algorithms that achieved higher accuracy when used to train the model. According to [7], in their study, compared six machine learning algorithms based on accuracy and processing time to identify the most suitable algorithm for DDoS attack detection, landed on DT as the best classifier. The model used a feature selection method and trained the model using DT, RF, NB, SVM, MLP and KNN. The paper highted the importance of features such as flow bytes, ethernet source and destination address in the identification of DDoS occurrences in SDN environment. The test was conducted in a simulated environment reported that, DT technique better detected DDoS attacks on the network. Same can be said by Sasikumar [22], who in his work proposed an ML model for identifying DDoS attacks and malware with higher accuracy. The study generated traffic using virtual instances in real-time in a private cloud, detecting DDoS attacks by looking at SNMP parameters and applying ML algorithms such as bagging, boosting and ensemble methods. The model yielded the best accuracy with SVM and DT, achieving an accuracy of 0.991 and 0.990 respectfully. This shows the predictability nature of DT as a probabilistic classifier. In another study conducted by Altamemi et al. [23] aimed to mitigate DDoS occurrence in SDN using machine learning for rapid detection employed algorithms such as Logistic regression, NB and DT and utilised of real-time dataset in the building of the model. This provided an up-to-date and realistic data for DDoS attack detection by the model. It achieved its aim of providing an efficient and effective detection and classification of network traffic. It is clear the size of the dataset that was used to train and test the model was small and this can make the model’s generalisation very difficult and unrealistic. The model achieved best prediction accuracy with the Decision Tree algorithm compared with other models in different studies with an accuracy of 99.90%. The Decision Tree again proved it worth in the model proposed by Almaraz-Rivera et al. [20] in their paper that proposed a novel IDS model for identifying and classifying DDoS attacks in IoT networks, which utilized the Bot-IoT dataset, addressing the class imbalance problem used three different feature set to train selected ML and DL algorithms for classification. The model achieved an average accuracy above 99% with Decision Tree outperforming the rest of the techniques. SVM, classification techniques, has proven itself in building predictive and detection models, achieving higher accuracy. SVM technique achieved high accuracy in the model proposed by Singh [24], which aims to develop a solution to identify and alleviate DDoS attacks by building a system that can comprehensively detect the attacks in SDN environment. The methodology involved the training of the model with SVM, KNN, DT, RF, Multi-Layer Perception (MLP) and Gaussian Naïve Bayes (GNB) for the classification of DDoS attacks using proposed dataset that is specifically designed for SDN environment. Even though the specific details on the proposed dataset wasn't explicitly stated, the study reported that, SVM achieved a perfect score in all metrics performing best among other techniques that was compared in the study.

Ghanbari et al. [25] in their study proposed a model which is built-up on an earlier model, Convolutional Neural Network (CNN) technique. The model utilizes a Variance Fractal Dimension Trajectory (DFDTv2) for input data feature extraction, and a discrete wavelet transform (DWT) for data pre-processing and SVM for the post-processing. The DFDTv2 and the DWT comprises the pre-processing of data, CNN training for the data classification as normal or anomalous at the processing stage, with SVM training and testing as the post-processing stage within the framework of the model. Even though the study reported that, SVM achieved the best accuracy within the model, the percentage accuracy (87.35%) can be improved. Prriyadarshini et al. [26] in their study, also used SVM classification technique. The methodology involved the creation of simulated network traffic using rule-based and blacklisting to capture traffic patterns of DDoS occurrence and training the model with the SVM classifier. The report stated effective and efficient in detection in both passive and active attack traffic and could classify non-linear issues with higher accuracy. The paper did not state explicitly the performance measure that was achieved by the model, but stated that, SVM yielded a better accuracy. Azizan et al. [27] in their study came out with a model which was based on ML network IDS and compared the performance among three ML techniques (Decision Jungle (DJ), RF, SVM). This study utilized the knowledge discovery in databased (KDD) and the CICDDoS2017 dataset as the benchmark for evaluation, while testing and evaluating on selected ML classifiers. The performance metrics that was compared were, accuracy, precision and recall with the SVM achieving an accuracy of 98.18% making it superior to the other techniques which recorded an accuracy of 97.76% and 96.50% for RF and DJ techniques respectively. Sahoo et al. [28] in their study, implementing the SVM classification technique in DDoS attack detection combined the SVM with Kennel Principal Component Analysis (KPCA) and a Genetic Algorithm. The model monitors the OpenFlow of switches in SDN network, using KPCA for feature extraction and SVM technique for the classification of network traffic. The combined KPCA, GA and SVM achieved a better result in terms of accuracy than a single SVM and other techniques used for the model building like Random Forest and K-Nearest Neighbour. The proposed model with the SVM achieved 97.04 rate of accuracy compared to single SVM and Random Forest classifier with accuracy 94.41 and 93.31 respectively. The combined model with SVM even though yielded a better overall accuracy, its accuracy on the Smurf DDoS attack was low compared to the other type of attacks. Dasari et al. [1] in their study came out with a method to evaluate the effectiveness of different ML algorithms in identifying different types of DDoS attacks. The paper used CICDDoS2019 dataset containing diverse sets of DDoS attacks and applied modelling on six ML algorithms which are LR, DT, RF, AdaBoost, KNN and NB for the detection of DDoS occurrence. This study didn't report on application of any feature selection technique to the dataset. The model's evaluation was based on classification accuracy, precision, recall, F1-score, with KNN, NB, LR and AdaBoost achieving accuracy above the rate of 0.9967. Prasad et al. [29] addressed the limitations of existing DDoS detection techniques in machine learning by proposing Stochastic Gradient Boosting (SGB). The study reported that a higher prediction accuracy was achieved when the SGB model was trained and fine-tuned on the selected datasets. The model achieved a perfect accuracy with zero misclassifications as compared to machine learning algorithms in DDoS attack detection (KNN, RF, DT). The model is an automatic classifier that automatically classify network traffic flow in real-time. SGB is known for their perfect accuracy and low misclassification as can also be seen in the paper by Narote et al. [30] in their proposed model which developed ML methods for DDoS attacks detection. The model used a hybridised SGB trained and tested on dataset obtained at different times for the mitigation of DDoS attacks on a network. The challenges in the existing models were address with new approaches which involved machine learning and blockchain in this study. Detailed information about the model and its evaluation is scanty, but the paper made it clear that the hybrid SGB model achieved a perfect accuracy with zero misclassification.

In another study by Hariharan et al. [31], proposed the use C5.0 to detect DDoS attacks and compare with state-of-the-art algorithms (Naïve Bayes, C4.5). The methodology deployed in the study involves the simulation of DDoS attack in a virtual environment and capturing the traffic to create a dataset, pre-process to take away features that are redundant and apply the C5.0 classifier. The model has a higher accuracy and fast detection rate, and can detect cloud-based DDoS attacks, the identification only takes place after the damage has been caused. This makes it more problematic in real-time implementation even though it achieved the best and 100% accuracy compared with NB and C4.5 classifiers. Issa and Tiemoman [32] in their model for identification of DDoS attacks in real-time, utilised SDN technologies, bloom filters and machine learning-based behavioural analysis of network traffic. The study developed an IDS based machine learning architecture using Linear Discriminant Analysis (LDA), KNN, SVM to classify normal and attack packets with bloom filters for the storage of known IP addresses of malicious sources. Combining the behaviour analysis, machine learning and SDN technologies helped to improve the model’s performance. The model achieved a better accuracy yet the exact details on accuracy and precision was not reported. Arshi et al. [5] also concluded in their study which discusses the use of different machine learning methods for detecting and analysing network attacks that, MLP achieved a higher precision rate compared with other algorithms which include Decision Trees, SVM and NB. In their work, dataset containing different attack types (UDP Flood, ICMP flood, Smurf attack and HTTP flood was used for modelling process. The use of this comprehensive dataset and the comparison between several machine learning algorithms provided a better insight into the detection of DDoS attacks. Polat et al. [33] in their study used feature selection method together with four machine learning techniques (SVM, ANN, KNN, NB) trained and tested on DDoS attack identification in SDN environment. The FS technique adopted in the study yielded a better result on all the ML techniques deployed with KNN achieving the best accuracy of 98.3%. Dong et al. [34] in their work presented an improved KNN ML algorithms to detect DDoS attacks in SDN using degree of attack for attack identification. The models are DDoS Detection Algorithm based on Degree of Attack (DDADA) and DDoS Detection Algorithm based on Machine Learning (DDAML). The model achieved a better performance in identifying and classifying network traffic compared with other traditional ML algorithms liked SVM, NB, KNN and CIC-SVM. The model achieved a true positive rate of 0.987 and false positive rate of 0.016 for the DDADA technique, 0.994, 0.009 for the DDAML technique. Khempetch et al. [8] in their study proposed the use of DNN AND LSTM to mitigate the risk of DDoS attacks in IoT systems. The study used the CICDDoS2019 dataset and employed feature selection and feature engineering methods to streamline the dataset, trained and tested the model using DNN and LSTM techniques. The study failed to give the computational and training time and resource requirements details of the proposed algorithm, but the study stated that both algorithms achieved a better accuracy in the detection of DDoS attacks with accuracy rate between 0.9993 – 1.00 in both models on all types of DDoS occurrences found in the CICDDoS2019 dataset. In the case of the [35] in a similar study, proposed an Enhanced Deep Sparse Autoencoder (EDSA) based architecture for the detection of DDoS attacks with minimal cost technique. EDSA is used for data extraction and the deep neural network for the classify network traffic. The study reported that the model yielded a higher detection accuracy with reduced percentage of false positives compared to traditional machine learning models. The model yielded an accuracy of 98% with a detection rate of 98.1% and precision, specificity, and false positive rate of 91%, 98% and 1.4 respectively. [36] in their study used DL approach for the identification of DDoS occurrences at the application layer using autoencoders for selecting features and DNN for the attack classification. The model used the CICDDoS2017 dataset and applied feature selection technique on it. This yielded a better performance as compare [35] with a lower false rate. The metrics for the evaluation of the model's performance was the rate of accuracy, detection rate and false positive rate, with the model achieving 99.83%, 99.84% and 0.17 respectively. Shurman et al. [2] proposed two methods for classifying Distributed reflected denial of service attacks in IoT devices. The method used is an IDS (signature-based and anomaly-based) system and a deep learning method that utilize LSTM to train the model using the latest DrDDoS attack dataset. This model has the potential of classifying network traffic irrespective of specific characteristics, recording accuracy of 99.17% against 99.0% and 73.9% in the case of Random Forest for same dataset in different study.

Ingle et al. [37] in their study proposed a three-pattern classification-based for DDoS outbreak identification based on packet flow and ML algorithms to augment the ineffective anomaly-based DDoS attack detection. The model uses LSTM machine learning model to classify network traffic using pattern that is based on time components in the dataset. This model is efficient in that it is scalable, easy to implement and adaptable to new trend of attacks and be able to detect an attack in progress, making it a good model to use in real-time and mission critical applications. The model's performance in terms of accuracy, precision and recall was 99.78%, 98.39% and 99.80% respectively which is an improvement over other models already discussed which used LSTM technique. Shieh et al. [38] in their paper which investigates the impact of Open Set Recognition (OSR) on the DDoS attack identification and to propose a new framework for detection which addresses this problem. The model utilized Bi-LSTM for traffic discrimination and Gaussian Mixture model adopted to differentiate trained models and novel instances and increment learning. The model utilized the CICDDoS2019 and CICDDoS2017 dataset for modelling. The combined ML and DL, with the incremental learning tackles the OSR problem and adopt to evolving DDoS attacks. The study reported 94% accuracy for the proposed model.

Xinlong and Zhibin [39] utilized a hybrid DL approach to classify DDoS attacks on a network, based on management of time information in network traffic flow to accurately classify DDoS attacks. The model used a Hierarchical Temporal Memory (HTM) and LSTM to encode time sequence of incoming data. The HTM utilized hierarchy of regions and columns to differentiate the input and LSTM to handle time sequences of data demonstrating a superior performance over other ML and DL techniques. The model can recognise complex patterns and time sequence, which is a combined ability of both HTM and LSTM technique. The study failed to give enough information on the architecture of the model, but stated that the model achieved 0.977 for accuracy, ROC, precision, recall and F1-score. [6] introduced a new model to address the challenge of mitigating low-rated DDoS attacks in SDN, presented a modular architecture that incorporated machine learning models and IDS. This involves the use of IDS and IPS together with machine learning algorithms to (J48, RF, REP tree, Random tree, SVM, MLP). The model's reliance on algorithms that seek patterns to classify flow traffic may not be efficient as compared to the use of deep learning technique that use hidden layers to capture more informative features. This paper achieved an accuracy: 0.9546, Precision: 0.9501, Recall: 0.9451, F1-Measure: 0.9498 and False rate 0.0052 with MLP, outperforming the other machine learning techniques. Ahmed et al. [40] addressed DDoS attack in the application layer by analysing features of incoming traffic flow to develop a classification model for effective attack detection. The model employed MLP and deep learning to make judgement on the effectiveness of DDoS attack identification. The model achieved highest 98.99% accuracy with MLP algorithm and minimal value of false positives. In their study, [11] introduced a novel approach involving the fusion of RNN and DL methodologies to classify DDoS attacks within a cloud-based setting. The architecture employed LSTM from the RNN family, coupled with an autoencoder-decoder-based technique from the realm of deep learning. To optimize network parameters, a hybrid optimization algorithm was utilized, comprising both Haris Hawk Optimization (HHO) and Particle Swarm Optimization (PSO). The efficacy of this model became evident in its ability to effectively tackle challenges such as delayed convergence, local stagnation, and trapping in both local and global optima – problems commonly encountered in existing DDoS attack detection models. However, it's worth noting that the computational time is prolonged due to the increased number of hidden layers within the neural network. The results showcased a notable accuracy of 0.9953 and an impressive F1-score of 0.9947, underscoring its superior performance. Aktar et al. [41] proposed a DL model in their study using a contractive autoencoder for identifying DDoS attacks. The training of the model involved learning the patterns of the normal network traffic and use stochastic threshold to detect attacks. The model relies on the reconstruction error as the means of anomaly detection which can be misleading and pose challenges in detecting evolving DDoS attacks which closely resembles the normal attack. The various techniques used in the training of the model (Basic Autoencoders, Variation Autoencoders and LSTM) achieved accuracy between 92.45% to 97.58% on all three datasets (CICDDoS2017, CICDDoS2019, NSL-KDD). Elsayed et al. [42] in their study, proposed an IDS based on deep learning (RNN-Autoencoder), detecting DDoS attacks within SDN environment. They referred to this model as DDoSNet. DDosNet is trained in an unsupervised way to extract useful feature representations from an input data and fine tune the training using sampled data to optimise the network. The model provides confidence in the SDN environment due to its accuracy in the detection of DDoS attacks, achieving 0.99 accuracy in all cases. Doriguzzi-Corin et al. [43] on the other hand, developed a DDoS attacks detection system that will analyse and classify live traffic without much processing overhead. The study employed a model LUCID: which is a Lightweight DL approach to the detection of DDoS attacks using CNN to classify traffic flow as attack or normal. The model is noted for its reduced processing overhead, a validated solution on resource constraint hardware and achieved an accuracy of 0.9888 – 0.9987 among the different datasets used to train the model. This study is supported by [10] in their study which sought to develop efficient deep learning-based algorithm for the detection of DDoS attacks in wireless sensor networks, reporting CNN as the best classifier which outperformed the other algorithms with an accuracy rate of 98.79%. The other algorithms, outperformed by the CNN classifier were DNN, RNN and CNN-RNN, trained on the WSN-DS. In this model, FS technique was not applied to identify the most important features in the detection of DDoS attacks which could have improved the accuracy of the model. A new model proposed by Lopes et al. [9] known as CyDDoS, is an integration of IDS framework to effectively detect DDoS attacks while reducing overhead. The model is an integration of feature engineering algorithms and deep neural network algorithms to design effective security mechanism against DDoS attacks. The feature selection process is manual and not based on any algorithm which may suffer efficiency issues when deployed with different dataset. The model outperformed other ML and DL algorithms and that of DDoSNet proposed in the study by Elsayed et al. [42] with an accuracy rate of 0.996. Najafimehr et al. [44] in their study proposed another hybrid machine learning algorithms for the detection of DDoS attacks with the focus on unknown and unprecedented attacks. The methodology deployed is in three phases; unsupervised phase – the used of clustering algorithms to separate the traffic, a cluster analyser phase – which is statistical measures calculated for each cluster and a supervised phase – that is the use of classification algorithms to label the clusters. The model's effectiveness lies in its ability to detect unknown and unprecedented attack traffic. Tekleselassie et al. [3] in a model, developed a novel DDoS attack detection that is expandable and flexible combining deep learning and knowledge graph classification. The model utilized the deep leaning for network traffic classification and the knowledge graph for the expandability of the model. The proposed model achieved an accuracy of 99.93%. Aslam et al. [45] in detecting and mitigating DDoS attacks on IoT devices, a framework named Adaptive Machine Learning SDN Model (AMLSDM) was proposed. This framework employs a multi-layered feed-forward approach to identify DDoS attacks through the analysis of static network traffic features. Classification algorithms used for the traffic classification includes KNN, SVM, NB, LR, RF: using Ensemble Voting. The adaptive machine learning enables the model to adapt to changing networking conditions and improved over time. The proposed model achieved and accuracy of score of 98.5%, which means the model can classify network traffic effectively. Saghezchi et al. [46] during their research, formulated a model that employed information gathered from an actual semiconductor manufacturing facility to create a dataset with labels. This dataset was subsequently used to train eleven distinct machine learning algorithms, comprising both supervised, unsupervised, and semi-supervised methods. Through simulation and subsequent evaluation, it was observed that the supervised learning approach exhibited superior performance compared to the alternative models. Notably, this technique achieved remarkable scores in accuracy, recall, precision, and F1-score, reaching a remarkable accuracy level of 0.999.

Balasubramaniam et al. [47] introduced a novel approach to classify Distributed Denial of Service (DDoS) attacks in cloud computing, utilizing a Deep Learning (DL) method trained by the optimised Gradient Hybrid Leader Optimization (GHLBO) algorithm. The methodology involves integrating gradient descent and hybrid leader-based optimization, and the resulting algorithm trains a Deep Stacked Autoencoder (DSA) for efficient DDoS attack detection. The paper reported achieving a true positive rate (TPR), true negative rate (TNR), and testing accuracy of 0.909, 0.909, and 0.917, respectively, indicating the effectiveness of the proposed GHLBO-based DSA. However, the article could benefit from a more in-depth discussion on computational complexity and specific aspects of feature fusion and data augmentation.

Again, Pandian et al. [48] also proposed a hybrid classification model for the classification of DDoS attack in telecommunication networks, combining neural network and SVM. This two-step approach involves a memory module (CNN) and a learning module (SVM). The CNSVM allows for more accurate attack detection and classification than using only SVM or CNN. The CNSVM achieved an accuracy of 89.50%. DDoS attacks on network infrastructure is still a challenge to organisations as lasting solution has not been deployed to eradicate the menace. The embracing nature of technology and the use of networks and networking devices to achieve 24/7 availability and global recognition, has made networks a lifeline to many businesses. ML and DL algorithms has been deployed in different studies to mitigate against DDoS attacks on network resources and infrastructure. From the review conducted, which involves the analysis of about 45 articles which are sampled from 2019 to 2023, has proved that a headway has been made in the use of machine and deep learning techniques for DDoS attack identification and alleviation. In the case of Prasad et al. [29], a perfect accuracy is achieved with zero misclassification making it a better performance by all standards. Hitherto, the reliance on the SGB pose a challenge and limitation in detecting DDoS attack patterns as more features cannot be extracted from data unlike deep neural networks in the case of novel and evolving attacks. Deep learning models have come to limelight with promising outcomes in the fight against DDoS attacks. The deep neural networks, with their capability to learn complex patterns in data makes them very effective models to deploy. In this review, [10] proposed model is one such model which showed a good classification accuracy with CNN classifier. Nonetheless, the hybrid model deployed in that study could yield an optimal outcome if feature selection is done for dimensionality reduction on the dataset coupled with extensive hyperparameter tuning. Techniques for feature selection are ways to select important features that have higher correlation on the desire output. This can serve as an opportunity for other researchers as gap that can be filled, asking the question; how hybrid deep learning technique can be used to effectively tackle DDoS occurrences considering the weakness in the datasets utilized, and how that could be overcome through feature selection and feature engineering with hyperparameter tuning techniques. Based on the review presented, enhanced feature selection, feature engineering and hyperparameter tuning techniques can be used to address some of the challenges identified in most of the studies presented.

2. METHODOLOGY

2.1 Description of Dataset

This paper makes use of the DDoS attack detection dataset provided by the Canadian Institute of Cybersecurity's DDoS 2019 dataset (CICDDoS2019), accessible on the institute's official website. Within this dataset, there is a compilation of both reflection-based and exploitation-based Distributed Denial of Service (DDoS) attacks, as categorized by Sharafaldin et al. [49]. The reflection-based DDoS attacks conceal their origins by leveraging authentic third-party elements. These attacks are executed through the application layer protocols TCP and UDP. The assortment of attacks documented in this dataset encompasses MSSQL, SSDP, NTP, TFTP, DNS, LDAP, NETBIOS, and SNMP. In addition to this, the exploitation-based DDoS attacks can also be executed within the application layer using the transport layer protocols TCP and UDP. Among these are SYN flood, UDP flood, and UDP Lag attacks. The dataset is grouped into csv files based on the type of attack which is a labelled dataset presented in the Table 1. This is in a categorical series with labels for each of the categories of attack type.

The dataset was made up of 83 features, consisting of various traffic statistics together with the label of that record or traffic entry. The features used were determined using a feature selection technique (discussed in subsequent sessions) to include only the important features to train the model for the classification of DDoS attack network traffic. Table 2 is a feature description of the dataset.

The features, detailed in Table 2, forms the network traffic statistics that was collected, and analysed to classify a network traffic as either a benign or DDoS attack.

Table 1. Description of the CICDDoS2019 dataset

DDoS ATTACK TYPE	DATASET ENTRIES	Index	Training Set 70%	Testing set 30%	Proportion
SYN	356496	12	249547	106949	31.53%
TFTP	227223	13	159056	68167	20.10%
DrDoS_NTP	129285	4	90499	38786	11.43%
BENIGN	113065	0	79145	33920	10%
Portmap	42606	11	29824	12782	3.77%
LDAP	41801	8	29260	12540	3.69%
UDP	33695	14	23586	10109	2.98%
UDP-lag	33377	15	23364	10013	2.95%
DrDoS_DNS	30618	1	21433	9185	2.71%
MSSQL	25280	9	17696	7584	2%
DrDoS_UDP	19413	7	13589	5824	1.72%
DrDoS_MSSQL	18054	3	12638	5416	1.60%
NetBIOS	16252	10	11376	4876	1.44%
DrDoS_NetBIOS	15363	5	10754	4609	1.36%
DrDoS_LDAP	14508	2	10156	4352	1.28%
DrDoS_SNMP	13563	6	9494	4069	1%
WebDDoS	51	16	11	5	0.00451%
TOTAL	1,130,650		791428	339186	100.00%

Table 2. Feature set of the CICDDoS2019 dataset

Feature No.	Feature Name	Feature Description
1	Source Port	the traffic source network connection port number
2	Destination Port	traffic destination network connection port number
3	Protocol	the communication protocol used in the network
4	Timestamp	the date time stamp of the network flow
5	Flow Duration	Network flow duration
6	Total Fwd Packets	Total number of packets in the forward direction
7	Total Backward Packets	Total number of packets in the backward direction
8	Total Length of Fwd Packets	total length of packets in the forward direction
9	Total Length Bwd Packets	Total length of packets in the backward direction
10	Fwd Packet Length Max	Maximum length of packet in the forward direction
11	Fwd Packet Length Min	Minimum length of packet in the forward direction
12	Fwd Packet Length Mean	the average length of packet in the forward direction
13	Fwd Packet Length Std	the standard deviation of packet length in the forward
14	Bwd Packet Length Max	the maximum length of packets in the backward direction
15	Bwd Packet Length Min	the minimum length of packets in the backward direction
16	Bwd Packet Length Mean	the average length of packets in the backward direction

Feature No.	Feature Name	Feature Description
17	Bwd Packet Length Std	Standard deviation of packet length in the backward
18	Flow Bytes/s	the rate of flow in bytes per second
19	Flow Packets/s	the rate of flow in packets per second
20	Flow IAT Mean	the average inter-arrival time between two consecutive flows
21	Flow IAT Std	Standard deviation of inter-arrival times between flows
22	Flow IAT Max	the maximum inter-arrival time between flows
23	Flow IAT Min	the minimum inter-arrival time between flows
24	Fwd IAT Total	Total inter-arrival time in the forward direction
25	Fwd IAT Mean	Average inter-arrival time in the forward direction
26	Fwd IAT Std	Standard deviation of inter-arrival times in the forward direction
27	Fwd IAT Max	Maximum inter-arrival time in the forward direction
28	Fwd IAT Min	the minimum inter-arrival time in the forward direction
29	Bwd IAT Total	Total inter-arrival time in in the backward direction
30	Bwd IAT Mean	Average inter-arrival time in the backward direction
31	Bwd IAT Std	Standard deviation of the inter-arrival time in the backward direction
32	Bwd IAT Max	the maximum inter-arrival time in the backward direction
33	Bwd IAT Min	the minimum inter-arrival time in the backward direction
34	Fwd PSH Flags	the number of push flags set in the forward direction
35	Bwd PSH Flags	the number of push flags set in the backward direction
36	Fwd URG Flags	the number of urgent flags set in the forward direction
37	Bwd URG Flags	the number of urgent flags set in the backward direction
38	Fwd Header Length	the total length of headers in the forward direction
39	Bwd Header Length	the total length of headers in the backward direction
40	Fwd Packets/s	Rate to packet in the forward direction
41	Bwd Packets/s	Rate of packets in the backwards direction
42	Min Packet Length	the minimum length of packets
43	Max Packet Length	the maximum length of packets
44	Packet Length Mean	the average of length of packets
45	Packet Length Std	the standard deviation of length of packets
46	Packet Length Variance	the variance of the length of packets
47	FIN Flag Count	the number of FIN flags set
48	SYN Flag Count	the number of synchronization flags set
49	RST Flag Count	the number of RST flags set
50	PHS Flag Count	the number of push flags set
51	ACK Flag Count	the number of acknowledgement flags set
52	URG Flag Count	the number of urgent flags set
53	CWE Flag Count	the number of CWE flags set
54	ECE Flag Count	the number of ECE flags set
55	Down/Up Ratio	the download to upload ratio
56	Average Packet Size	the average of packets size
57	Avg Fwd Segment Size	The average size of forward segments
58	Avg Bwd Segment Size	the average size of backward segments
59	Fwd Header Length	the length of the header in the forward direction
60	Fwd Avg Bytes/Bulk	The average number of bytes per bulk forward packet
61	Fwd Avg Packets/Bulk	The average number of packets per bulk forward packet
62	Fwd Avg Bulk Rate	The average bulk forward rate
63	Bwd Avg Bytes/Bulk	The average number of bytes per bulk backward packet

Feature No.	Feature Name	Feature Description
64	Bwd Avg Packets/Bulk	The average number of packets per bulk backward packet
65	Bwd Avg Bulk Rate	the average bulk backward rate
66	Subflow Fwd Packets	The number of subflow packet in the forward direction
67	Subflow Fwd Bytes	the number of subflow bytes in the forward direction
68	Subflow Bwd Packets	The number of backward subflow packets
69	Subflow Bwd Bytes	The number of backward subflow bytes
70	Init_Win_bytes_forward	The initial window size in bytes in the forward direction
71	Init_Win_bytes_backward	The initial window size in bytes in the backward direction
72	Act_data_pkt_fwd	The number of actual data packets in the forward direction
73	Min_seg_size_forward	The minimum segment size in the forward direction
74	Active Mean	The average time of active connections
75	Active Std	The standard deviation of active connections
76	Active Max	The maximum time of active connections
77	Active Min	The minimum time of active connections
78	Idle Mean	The average time of idle connections
79	Idle Std	The standard deviation of idle connections
80	Idle Max	The maximum time of idle connections
81	Idle Min	The minimum time of idle connections
82	Inbound	Indicates inbound or outbound of the network flow
83	Label	The classification of the network flow

2.2 Proposed Methodology

The proposed methodology deployed in this study focused on two key areas in Machine Learning (ML), which are data preprocessing and modelling phase. The Fig. 1 shows an overview of the proposed methodology adopted by the study.

2.2.1 Data preprocessing

Data pre-processing was applied to clean the data for the modelling process (data cleaning, data transformation and normalization and data splitting).

Data cleaning technique was performed to improve dataset quality and remove noise, and handle missing values, making the dataset standard to enhance the performance of the model. This handled missing data, noise, feature scaling using normalization or standardization, and feature engineering.

Data transformation was also carried out to convert the raw data into a form that can be handled by the model and for easy analysis. The attack types in the dataset were converted into numeric values with Benign, DrDoS_DNS, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NTP, DrDoS_NetBIOS, DrDoS_SNMP, DrDoS_UDP, LDAP, MSSQL, NetBIOS, Portmap, SYN, TFTP, UDP, UDP-Lag and WebDDoS as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16 respectively. It is worth noting that, the researchers delve deep into the preprocessing activities as it’s a way of ensuring dataset quality in training the model for better performance.

Data splitting was carried out to split dataset into training and testing sets as one of the central preprocessing activities that has influence on the model’s performance. This is so because, when a model is trained on a particular dataset and is tested with different dataset in terms of features, the results will not be as accurate, and performance will be affected than where same dataset is in both cases. The dataset was split into 70% training and 30% testing set as exhibited in Table 1 above.

2.2.2 Modelling

This involved the process where the data was modelled using the CRNN-Infusion model proposed in this study. The dataset was trained on the model by initially using the CNN model to train on the dataset, without the output of the CNN, the RNN model is directly stack on it using the Gated Recurrent Unit (GRU) which provides two gates; update and reset gates, controlling the flow of information within the network and enables the model to selectively update and reset the hidden state at each time step. The GRU has fewer gates and parameters thereby making it faster to train compared with LSTM. The proposed CRNN-Infusion model introduces input layer with a 1D convolution with the feature size of the dataset and a hidden layer with filter size of 64, kernel size of 3 and a relu activation function. The RNN part was implemented with the addition of the GRU input layer of 64 filters, together with the CNN output layer and a relu activation function, a hidden layer with 64 filters is added and finally, an output layer with filter size equivalent to the dataset classes, compiled on a categorical cross entropy, an Adam optimization and evaluated on accuracy metrics.

The initial model's performance was recorded, and an optimization technique applied using Hyper Parameter Tuning (HPT) for selecting appropriate parameters that is best fit and produced a better results or outcome. This includes the selection of number of layers, the activation function technique employed and the epochs (number of times the model passes through the dataset) that should be carried out for the model to achieve better prediction between underfitting and overfitting. The technique employed was the Random Search Hyperparameter Tuning Technique (RSHPT). The results obtained using RSHPT was compared with an initial outcome without FS and HPT to ascertain the extent to which the algorithm's performance has improved based on the hyper parameter tuning. Once an optimization was achieved, the result was analysed and evaluated. In the modelling process, a feature selection was applied using various techniques in picking out important attributes for improving the model's performance. The modelling process hyperparameters, model architecture, and the algorithm used is summarised in Table 3, Fig. 2 below.

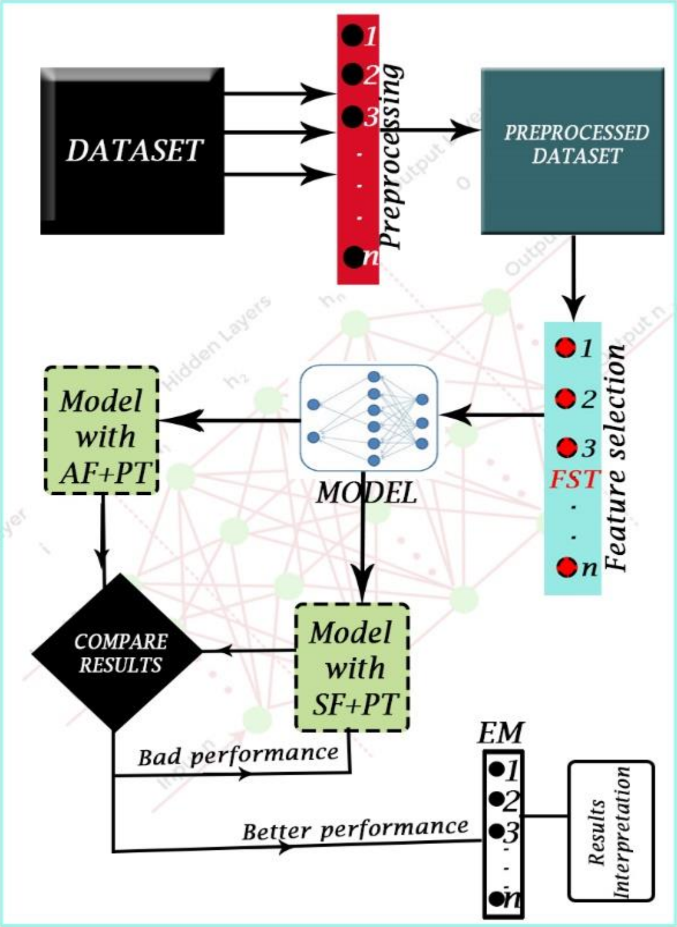


Fig. 1. Overview of the proposed methodology (CRNN-Infusion)

Table 3. CRNN-Infusion model hyperparameter

Hyperparameter	Value
Epoch	25
Activation Function	Relu, Softmax
Batch size	128
Loss Function	Categorical cross entropy
Optimization algorithm	Adam
Learning rate	0.001 (default)
Verbose	1

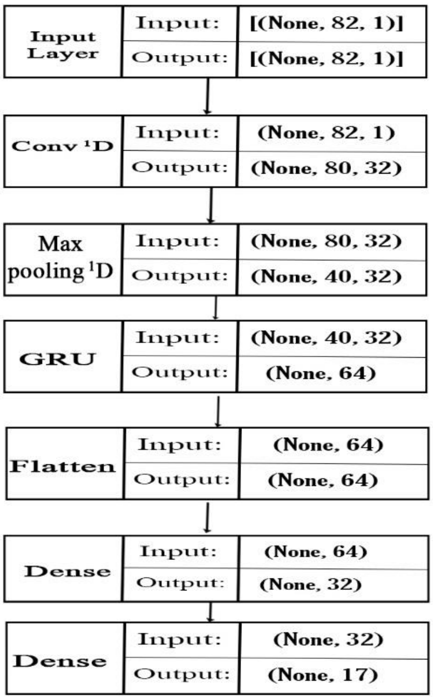


Fig. 2. Proposed CRNN-Infusion model architecture

2.2.3 Simulation environment

In conducting the simulation for performance evaluation of the proposed CRNN-Infusion, the model was implemented in Python 3.10.9 using the TensorFlow and Keras environment on a device with 16GB RAM, Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz 2.71GHz with a 64Bit Windows operating system. The Fig. 2 above was the architecture of the proposed CRNN-Infusion model.

2.2.4 Modelling algorithm

DDoS Attack Detection Algorithm

- ***Input** the CICDDoS2019 dataset (Training & Testing set)*
- ***Output** Classification results: accuracy, precision, recall, F1 score, and Confusion matrix.*

```
Begin: Data preprocessing
        X2 = Data cleaning (X1)
End

Begin: Feature Selection

F1 = dframe.corr()
F2 = mutual_info(x,y)
F3 = feature_importance(x,y)
SF = CorrMIRFFI(F2,F2,F3)
End

Begin: Training and Classification
Train the CRNN-Infusion classifier using CICDDoS2019 training set.
```

```
    Testing dataset CICDDoS2019 are put into
    the trained CRNN-Infusion classifier to
    detect attacks
        .
    End
```

Return *classification report.*

2.2.5 Feature Selection (FS)

In this study, the feature selection techniques employed includes the Correction Analysis, Mutual information, and Random Forest Feature Importance (RFFI). These FS techniques were selected due to their explorative nature, and discretised feature importance value.

Correlation Analysis: The range of the correlation sample spans from -1 to 1. Positive correlation signifies higher values in one variable correspond to higher values in another, while negative correlation indicates that higher values in one variable correspond to lower values in another [50]. This is mathematically expressed as r = (Σ ((x - mean(x)) * (y - mean(y)))) / (n * std(x) * std(y)), where Σ represents the sum of all data points, x and y denote the two features, mean(x) and mean(y) denote the means of x and y respectively, and std(x) and std(y) represent the standard deviations of x and y respectively, with n representing the number of data points.

Mutual Information: This quantifies the volume of information that a variable imparts about a target variable. Features with higher mutual information are regarded as crucial features. The mathematical representation is I(X: Y) = ΣΣ p(x, y) log(p(x, y) / (p(x) * p(y))), where Σ encompasses the summation of all conceivable values of x and y, p(x,y) stands for the joint probability mass function of x and y, and p(x) and p(y) denote the individual probability mass functions of x and y respectively. This approach was implemented using the mutual_info_classif method found in the sklearn library for Python.

Random Forest Feature Importance (RFFI): In this study, it was executed using the RandomForestClassifier from the sklearn library in Python.

The application of these feature selection techniques is illustrated in the appendix section of this report.

2.2.6 Model Performance Evaluation (MPE)

In the context of this study the metrics used is that of the classification task metrics looking at the nature of the model deployed. CRNN-Infusion is a classification model and the use of metrics for the evaluation of the model include Accuracy, Precision, Recall, F1 Score and Confusion matrix.

2.2.6.1 Accuracy

Accuracy is calculated as the ratio of samples correctly classified to the sample total predicted, mathematically represented as $\boldsymbol{A} = \frac{\boldsymbol{CP}}{\boldsymbol{TP}}$. This provides a general overview of the correctness and appropriateness of how a model can perform on a given dataset. In the case of this study, the accuracy was used as measure, but its interpretation would be much dependent on the TP and FP rates, to check if the accuracy is not a way the model is exhibiting overfitting or underfitting.

2.2.6.2 Precision

It focuses on measuring the actual predictions correctly made in a dataset. When used with accuracy in an unbalanced data, precision clarifies the inconsistencies in terms of how accurate the model is since it puts into account only the positive instances that were correctly predicted hence given a representation of correctly predicted percentage on any dataset. This is calculated as: $P = \frac{TP}{TP+FP}$, where TP is True Positive and FP is False Positives. It is focused on reducing false positives as possible, providing a better model's performance representation. Precision is used in this study as a means of performance measure of the CRNN-Infusion model.

2.2.6.3 Recall

The recall on the other hand, is focused on reducing false negatives as possible by focusing on the ratio of instances predicted correctly (true positives) to the actual positive occurrence (true positive + false negative). This is given mathematically as $R = \frac{TP}{TP+FN}$. This metric was utilized in this model to check the rate of false negatives in the model's prediction.

2.2.6.4 F1 Score

This is a combined metric value of precision and recall. This score ranges from 0 to 1 with 1 interpreted as perfect precision and recall and 0 being a poor performance. This is mathematically represented as $F1 - score = 2 * \frac{P*R}{P+R}$, where P for precision and R for recall. The model made use of this metric as a measure to find out how balanced the model performed in minimising false positives and false negatives.

2.2.6.5 Confusion Matrix (CM)

The CM, a performance measure, furnishes a comprehensive breakdown of a model's predictions compared to the actual class labels, facilitating the comprehension of the model's performance. It takes the form of a square matrix, where each row signifies the actual class labels, and each column signifies the predicted class labels. This matrix offers an overview of the model's True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These values underpin the computation of performance metrics like accuracy, precision, recall, and F1 score, as previously discussed in the report's preceding sections. This metric was also applied in evaluating the proposed model's analysis.

3. RESULTS AND DISCUSSION

The modelling process was carried out in two sessions, with the initial session comprising of basic parameters without any FS technique applied to data to select important features to classify DDoS attacks on a network traffic. The CRNN-Infusion model was then trained on the dataset without FS (Initial) and the outcome of the model recorded, and the model trained again with FS and RSHPT (Optimised).

The CRNN-Infusion initial modelling recorded 97.38% accuracy, 96.38% precision, 97.38% recall and 97.78% for F1-score on all classes. Whereas the optimised modelling recorded 98.92% accuracy, 99.02% precision, 98.92% recall and 98.93% F1 score. The models' initial and optimised performance is summarized in the Table 4 and Fig. 3 for all class instances in the dataset. The Fig. 3 shows the model's accuracy given the number of epochs as against the cross-entropy loss on the initial modelling. From Table 4, the model's performance based on precision rate, recall rate and F1 score is between the range of 0% - 100% for the various classes in the database.

This result depicts that the CRNN-Infusion model can classify DDoS attacks achieving a 98.92% rate of accuracy, 99.02% precision rate, 98.92% recall, and 98.93% F1-score on the testing dataset. This is very evident that, the proposed CRNN-Infusion model can detect DDoS attacks on a network infrastructure as the result achieved for this objective has a higher rate of detection and classification accuracy above 90%.

3.1 Comparison of Model's Performance Using Accuracies

The results presented above in both the initial and optimised modelling, depicts a better performance when the optimisation technique was applied. The initial model obtained an accuracy of 97.38%, with other accuracies as precision, recall and F1 score of 96.38%, 97.38% and 96.78% respectively for weighted averages and 84%, 87% and 85% respectively for macro averages been the measures of how well the model performed in the classification of the various network attack types. This, compared to the results when the RSHT and FS optimisation technique was used in the training of the model yielded a positive outcome as the optimised model yielded an accuracy of 98.92% with precision, recall and F1 score of 99.02%, 98.92% and 98.93% respectively for weighted averages and macro averages of 92% on all metrics. From these results, the model was optimised by 2.74%, 1.58% and 2.22% on weighted averages for precision rate, recall rate and F1 score respectively with macro average optimised by 9.5%, 5.75% and 7.23% respectively with an overall accuracy improvement of 1.58%. This is less but significant figure as far as appropriate classification of network traffic is concerned.

This is a very welcoming performance which provides the means of classifying these attacks and ensuring appropriate mitigation strategies deployed. Summary of the models' performance between the initial modelling and the optimised modelling is depicted in the Fig 3.

The summary of the findings from the initial and the optimised modelling show that the model was successful in the optimisation process, which enhanced the model's accuracy rate of classification.

3.2 Comparison of Models' Performance Using Confusion Matrix

The confusion matrix helps to ascertain the extent to which each of the classes were correctly (True Positive (TP) and True Negative (TN)) or wrongly (False Positive (FP) and False Negative (FN)) classified. It was ascertained from the results of the initial modelling from Table 5 that, out of the total test Benign samples of 28868, 28835 (99.88%) were TP with only 33 (0.12%) as FP, DrDoS_DNS recorded 8096 (99.56%) TP with 36 samples (0.44%) as FP rate, DrDOS_LDAP with 3857 samples (99.71%) TP with 11 samples (0.29%) FP. DrDoS_MSSQL, DrDOS_NTP, DrDoS_NetBIOS, recorded 4621 (96.79%), 34746 (99.16%), 4026 (98.53%), and 3473 (95.25%) TP rates with 153 (3.21%), 118 (0.34%), 60 (1.47%), and 173 (4.75%) as FP respectively. For DrDoS_SNMP, DrDoS_UDP, LDAP, MSSQL, NetBIOS, Portmap, and Syn; the TP rates were 5273 (99.62%), 11230 (99.88%), 6852 (99.92%), 4332 (99.90%), 11401 (99.19%), 86306 (98.77%) constituting the TP rate with 20 (0.38%), 13 (0.12%), 5 (0.08%), 4 (0.10%), 92 (0.81%) and 1067 samples (1.23%) were the FP rates respectively. With TFTP, UDP, UDP_Lag and WebDDoS; the model's initial performance on confusion matrix yielded the following: 61195 samples (98.77%), 9117 (99.89%), 8020 (88.85%) and 0 (0%) TP rates with 495 (0.81%), 10 (0.11%), 1057 (11.65%) and 19 (100%) were the FP rates recorded by the model.

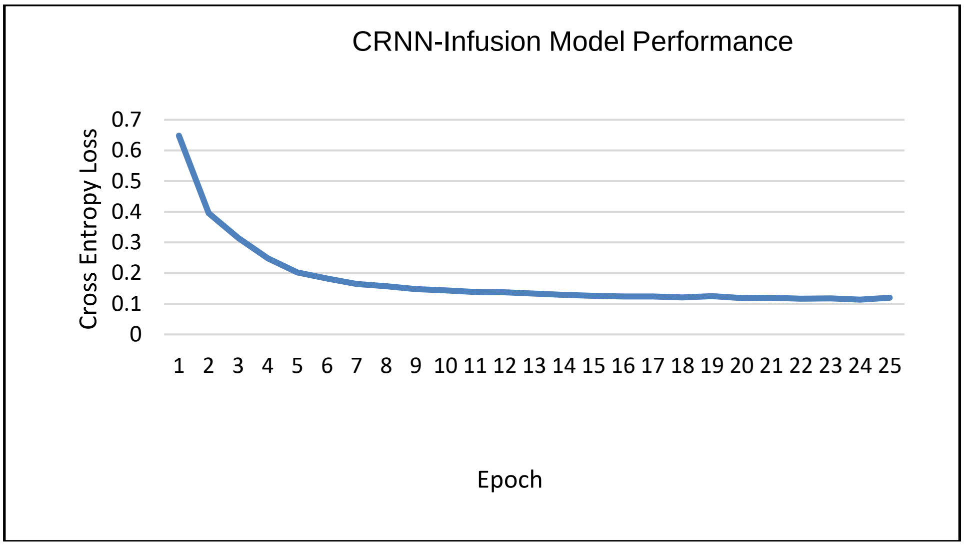
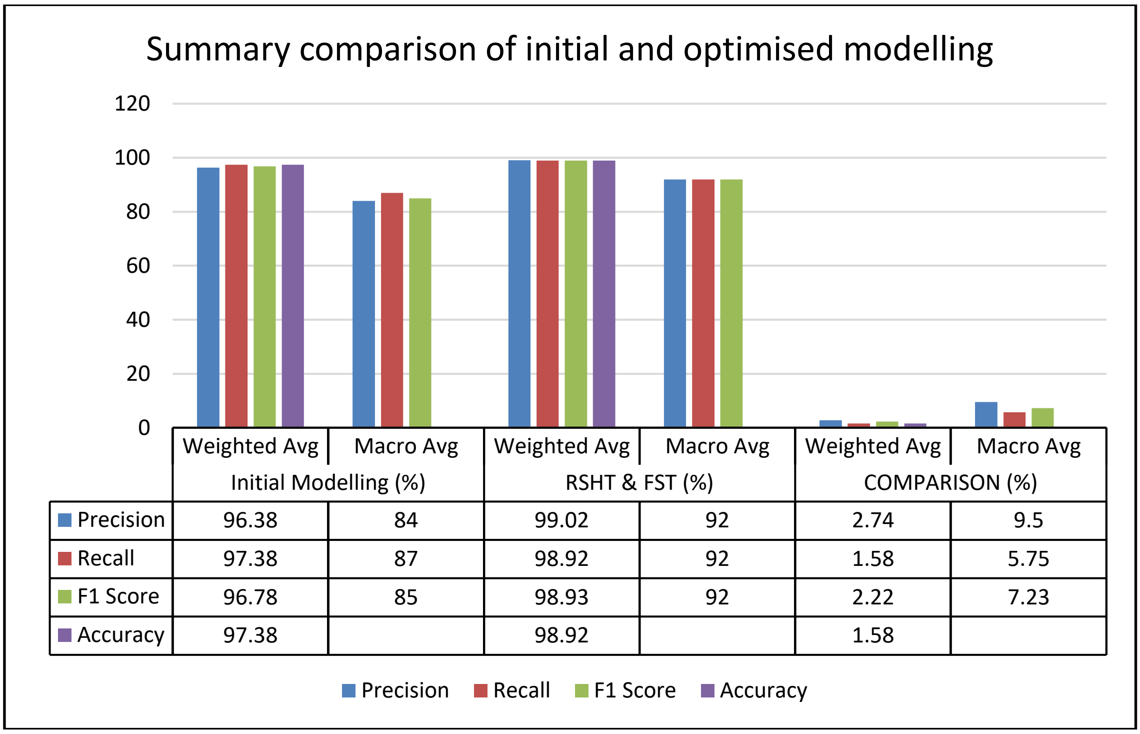


Fig. 3. Plot of loss Vs epoch for initial modelling

Table 4. Summarised Performance of Algorithm on the Testing Set for Initial Optimised Modelling (I – Initial, O – Optimised)

Type of Attack	Precision		Recall		F1 score		Accuracy	
Session	I	O	I	O	I	O	I	O
Benign	0.99	0.99	1.00	1.00	0.99	0.99		
DrDoS_DNS	1.00	1.00	1.00	0.99	1.00	1.00		
DrDoS_LDAP	0.99	0.99	1.00	1.00	0.99	1.00		
DrDoS_MSSQL	0.98	0.99	0.97	0.99	0.97	0.99		
DrDoS_NTP	1.00	0.99	1.00	1.00	1.00	0.99		
DrDoS_NetBIOS	0.95	0.96	0.99	0.99	0.97	0.97		
DrDoS_SNMP	0.97	1.00	0.95	0.95	0.96	0.98		
DrDoS_UDP	0.83	1.00	1.00	0.90	0.91	0.95		
LDAP	1.00	1.00	1.00	1.00	1.00	1.00		
MSSQL	1.00	1.00	1.00	0.96	1.00	0.98		
NetBIOS	0.00	1.00	0.00	0.91	0.00	0.95		
Portmap	0.72	0.97	0.99	1.00	0.84	0.98		
Syn	1.00	1.00	0.99	0.99	0.99	0.99		
TFTP	1.00	1.00	0.99	0.99	1.00	1.00		
UDP	1.00	1.00	1.00	1.00	1.00	1.00		
UDPLag	0.85	0.83	0.88	1.00	0.87	0.91		
WebDDoS	0.00	0.00	0.00	0.00	0.00	0.00		
Overall	0.9638	0.9902	0.9738	0.9892	0.9778	0.9893	0.9738	0.9892



	Weighted Avg	Macro Avg	Weighted Avg	Macro Avg	Weighted Avg	Macro Avg
	Initial Modelling (%)		RSHT & FST (%)		COMPARISON (%)	
🟦 Precision	96.38	84	99.02	92	2.74	9.5
🟥 Recall	97.38	87	98.92	92	1.58	5.75
🟩 F1 Score	96.78	85	98.93	92	2.22	7.23
🟪 Accuracy	97.38		98.92		1.58	

Fig. 4. Summary of model's initial performance and the optimised performance

The optimised technique (RSHT & FST) showed a positive improvement based on the confusion matrix report from Table 6. Among the rates for Benign was 28849 samples (99.936%) TP with only 19 samples (0.066%) classified as FP. This is an improvement on the initial modelling metric obtained for this class of network traffic. The remaining network traffic attacks saw significant improvement after the optimisation technique implementation as DrDoS_DNS, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NTP, DrDoS_NetBIOS, DrDoS_SNMP, LDAP, MSSQL, NetBIOS, Portmap, Syn, TFTP, UDP, UDP_Lag and WebDDoS recorded TP rates and FP rates in their respective order as follows; 8074 (99.287%), TP and 58 (0.713%) FP, 3867 (99.141%) TP and 1 (0.026%) FP, 4733 (99.629%) TP and 41 (0.859%) FP, 34748 (98.629%) TP and 124 (0.356%) FP, 4030 (95.447%) TP and 56 (1.371%) FP, 3480 (90.327%) TP and 166 (9.553%) FN, 4781 (90.327%) TP and 512 (9.673%) FP, 11240 (99.973%) TP with 3 (0.027%), 6605 (96.325%) TP with 252 (3.675%) of FP, 3947 (91.029%) of TP with 389 (8.971%) FP, 11454 (99.661%) TP with 39 (0.339%) FP, 86380 (98.863%) with 993 (1.137%) FP, 612001 (99.920%) TP with 491 (0.080%) FP, 9111 (99.825%) TP with 16 (0.175%) as FP, 9068 (99.901%) TP with 9 (0.099% as FP rates and WebDDoS did not have any correct classification with 19 (100%) FP rates which was the only class where its TP rate and FP rates did not improve in both the initial model performance and the RSHT & FST model’s performance.

3.3 Proposed Model’s Performance with other Models

The study after achieving optimal performance with the proposed model (CRNN-Infusion) made a comparison with other hybrid CNN + RNN models and models modelled on the CICDDoS2019 dataset. The Table 7 shows the report of the proposed model comparison with other models.

The comparison of the proposed CRNN-Infusion with other models in the field of deep learning, or models that utilised the CICDDoS2019 dataset as depicted in the Table 7 above, which forms the baseline for the model’s performance evaluation, is a confirmation that the proposed model was effectively optimised for DDoS attacks identification on a network. The Table 7 shows that, the proposed model had the best performance of 98.92% accuracy followed closely by [10] CNN model with 98.75%.

Table 5. Confusion Matrix on the Proposed CRNN-Infusion Model (Initial Modelling)

Benign	**28835**	**6**	**0**	**0**	**7**	**6**	**0**	**1**	**0**	**0**	**0**	**4**	**0**	**0**	**1**	**8**	**0**
DrDoS_DNS	3	**8096**	6	0	22	2	0	0	0	1	0	1	0	0	1	0	0
DrDoS_LDAP	0	0	**3857**	10	0	0	0	0	0	0	0	0	1	0	0	0	0
DrDoS_MSSQL	2	0	30	**4621**	0	5	113	1	0	0	0	0	0	1	0	1	0
DrDoS_NTP	108	0	0	1	**34746**	0	0	0	0	0	0	0	0	5	0	4	0
DrDoS_NetBIOS	7	0	0	49	0	**4026**	3	1	0	0	0	0	0	0	0	0	0
DrDoS_SNMP	2	0	0	7	0	164	**3473**	0	0	0	0	0	0	0	0	0	0
DrDoS_UDP	10	0	0	0	0	0	9	**5273**	0	0	0	0	0	0	0	1	0
LDAP	0	0	0	0	3	0	0	0	**11230**	10	0	0	0	0	0	0	0
MSSQL	3	1	0	0	0	0	0	0	0	**6852**	0	0	1	0	0	0	0
NetBIOS	2	1	0	0	1	0	0	0	0	0	**4332**	0	0	0	0	0	0
Portmap	38	5	2	5	0	19	0	0	10	5	0	**11401**	0	0	2	6	0
Syn	123	0	0	17	0	5	0	29	0	0	0	0	**86306**	1	0	892	0
TFTP	25	0	0	0	1	0	0	2	0	0	0	0	2	**61195**	0	467	0
UDP	6	0	0	0	0	1	0	1	2	0	0	0	0	0	**9117**	0	0
UDPLag	15	0	0	0	0	1	0	1029	0	0	0	0	0	12	0	**8020**	0
WebDDoS	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table 6. Confusion Matrix of the CRNN-Infusion Model (Optimised Modelling)

Benign	**28849**	**4**	**0**	**0**	**11**	**3**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
DrDoS_DNS	4	**8074**	5	0	9	0	0	0	0	1	0	0	0	0	39	0	0
DrDoS_LDAP	0	0	**3867**	0	0	0	0	0	0	0	0	0	1	0	0	0	0
DrDoS_MSSQL	1	0	29	**4733**	0	0	0	0	0	0	0	11	0	0	0	0	0
DrDoS_NTP	116	0	0	0	**34748**	0	0	0	0	8	0	0	0	0	0	0	0
DrDoS_NetBIOS	4	0	0	50	0	**4030**	0	0	0	0	0	0	2	0	0	0	0
DrDoS_SNMP	2	0	0	0	0	164	**3480**	0	0	0	0	0	0	0	0	0	0
DrDoS_UDP	3	0	0	0	0	0	3	**4781**	0	0	0	0	2	0	0	504	0
LDAP	2	0	0	0	0	0	0	0	**11240**	0	1	0	0	0	0	0	0
MSSQL	2	0	0	0	250	0	0	0	0	**6605**	0	0	0	0	0	0	0
NetBIOS	0	0	0	0	0	0	0	0	0	0	**3947**	389	0	0	0	0	0
Portmap	38	0	0	0	1	0	0	0	0	0	0	**11454**	0	0	0	0	0
Syn	78	0	0	14	0	0	0	3	0	0	0	0	**86380**	0	0	898	0
TFTP	25	0	0	0	0	0	0	0	0	0	0	0	0	**612001**	0	466	0
UDP	0	0	0	0	5	0	0	1	0	7	0	3	0	0	**9111**	0	0
UDPLag	7	0	0	0	0	0	0	0	0	0	0	0	1	1	0	**9068**	0
WebDDoS	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table 7 Proposed CRNN-Infusion Model Compared with Other Models

Reference	Dataset	Algorithm	Metrics			
			Accuracy	**Precision**	**Recall**	**F1 Score**
Sindian and Sindian [35]	CICDDoS2019	SAE & DNN	98%	91%	NA	NA
Shieh et al. [38]	CICDDoS2019	BI-LSTM-GMM	94%	87.2%	99.9%	97.6%
Alghazzawi et al. [51]	CICDDoS2019	CNN + BiLSTM	94.52%	94.74	92.04%	93.44%
Xinlong and Zhibin [39]		HTM & LSTM	97.7%	97.20	97.92%	97.72%
Salmi and Oughdir [10]	WSN-DS	CNN+RNN	96.50%	85.17%	84.50%	81.87%
Salmi and Oughdir [10]	WSN-DS	CNN	98.75%	94.86%	92.97%	93.72%
Aktar and Nur [41]	CICDDoS2019	BAE, VAE & LSTM	92.45% – 97.58 %	NA	NA	NA
Proposed Model	CICDDoS2019	CRNN-Infusion	98.92%	99.02%	98.92%	98.93%

4. CONCLUSION, RECOMMENDATION AND FUTURE WORKS

DDoS attack is on the rise as institutions, businesses and individuals are turning their focus to the use of technology as the main drive in the discharge of most obligations. Leveraging on the effectiveness of these technological tools posed the danger of been attacked in the cyberspace if proper measures are not put in place to detect and mitigate these attacks. One of such common attacks is the DDoS attacks. To mitigate DDoS attacks, early detection is of importance as well as the proper classification of the attack type to activate the required mitigation measure without disruption to legitimate network traffic. This study proposed a hybrid DL model (CRNN-Infusion) with effective hyperparameter tuning and feature selection technique to detect and effectively classify DDoS attack traffic on unseen DDoS attack types. Even though, the model did not achieve a perfect accuracy, it showed an exceptional performance in the classification of varied DDoS attack types and showed a higher classification accuracy.

The proposed model lives up to the standard of the classification task as the model was trained on a multiclass dataset with varied network attack types, yet the following areas can be investigated in the classification of DDoS attacks on a network.

In the contest of feature selection or dimensionality reduction, other FS techniques could be employed to determine the best and important features that could reduce the selected features and further reduce model's dimensionality.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Dasari BK, Davarakonda N. Detection of Different DDoS Attacks Using Machine Learning Classification Algorithms," Ingénierie des Systèmes d'Information. 2021;26(5):461-468,
2. Shurman M, Khrais R, Yateem A. DoS and DDoS Attack Detection Using Deep Learning and IDS," The International Arab Journal of Information Technology. 2020;17(4A):655-661,
3. Tekleselassie H. A Deep Learning Approach for DDoS Attack Detection Using Supervised Learning," MATEC Web of Conferences. 2021;348:01012-01019.
4. World Bank. The Key to Creating More Jobs in Ghana: Driving Technological Transformation of Micro-, Small and Medium-sized Enterprises," 24 November 2021. Available:https://www.worldbank.org/en/news/feature/2021/11/24/the-key-to-creating-more-jobs-in-ghana-driving-technological-

transformation-of-micro-small-and-medium-sized-enterprises.

5. Arshi M, Nasreen M, Madhavi K. A Survey of DDoS Attacks Using Machine Learning Techniques, E3S Web of Conferences. 2020;184:01052-01058.
6. Perez-Diaz AJ, Valdovinos AI, Choo RKK, Zhu D. A Flexible SDN-Based Architecture for Identifying and Mitigating Low Rate DDoS Attacks Using Machine Learning, IEEE Access. 2020;8:155859-155872.
7. Le TD, Dao HM, Nguyen TLQ. Comparison of machine learning algorithms for DDoS attack detection in SDN, Informatsionno-upravliaiushchie sistemy [Information and Control Systems]. 2020;3:59-70.
8. Khempetch T, Wuttidittachotti, DDoS attack detection using deep learning, IAES International Journal of Artificial Intelligence (IJ-AI). 2020;10(2):382-388.
9. Lopes OI, Zou D, Ruambo AF, Akbar S, Yuan B. Towards Effective Detection of Recent DDoS Attacks: A Deep Learning Approach," Security and Communication Networks. 2021;5710028-5710041.
10. Salmi S, Oughdir L. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network, Journal of Big Data. 2023;17-42.
11. Sumathi S, Rajesh R, Lim S. Recurrent and Deep Learning Neural Network Models for DDoS Attack Detection, Journal of Sensors. 2022;8530312-8530332.
12. Filho LdSF, Silveira AFF, Brito MdAJ, Vargas-Solar G, Silveira FL. Smart Detection: An Online Approach for DoS/DDoS Attack," Security and Communication Networks. 2019;15.
13. Pei J, Chen Y, Ji W. A DDoS Attack Detection Method Based on Machine," Journal of Physics: Conf. Series. 2019;1237:032040.
14. Gupta S, Grover D, AlZubi AA, Sachdeva N, Baig WM, Singla J. Machine Learning with Dimensionality Reduction for DDoS Attack Detection," Computers, Materials & Continua. 2021; 72(2):2665-2682.
15. Ahmed SS, Shet RS. A Study of Machine Learning Algorithms for DDoS Detection, International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2021;9(6):174-178.
16. Malliga S, Nandhini SP, Kogilavani VS. A Comprehensive Review of Deep Learning Techniques for the Detection of (Distributed) Denial of Service Attacks, Information Technology and Control. 2022;180-215.
17. Ismail IM, Mohmand H, Hussain KA, Ayaz U, Ullah M, Zakarya A, Ahmed M, Raza UI. Rahman and M. Haleem, A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks, IEEE Access. 2022;10:21443-21454.
18. Sanjeetha R, Kanavalli A, Gupta A, Pattanaik A, Agarwal S. Real-time DDoS Detection and Mitigation in Software Defined Networks using Machine Learning Techniques, International Journal of Computing. 2022;21(3):353-359.
19. Alduailij M, Khan WQ, Tahir M, Muhammad S, Alduailij M, Malik F. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method, Symmetry. 2022;14:1095-1109.
20. Almaraz-Rivera GJ, Perez-Diaz AJ, Cantoral-Ceballos AJ. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models, Sensors. 2022;22:3367-3384.
21. Ulemale T. Review on Detection of DDOS Attack using Machine Learning," International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2022;10; (3):764-768.
22. Sasikumar H. DDoS Attack Detection and Classification using Machine Learning Models with Real-Time Dataset Created, International Journal of Recent Technology and Engineering (IJRTE). 2021;9(5):145-153.
23. Altamemi JA, Abdulhassan A, Obeis TN. DDoS attack detection in software defined networking controller using machine learning techniques, Bulletin of Electrical Engineering and Informatics. 2022;11(5):2836-2844.
24. Singh KA. Machine Learning in OpenFlow Network: Comparative Analysis of DDoS Detection Techniques, The International Arab Journal of Information Technology. 18;2.
25. Ghanbari M, Kinsner W. Detecting DDoS Attacks Using Polyscale Analysis and Deep Learning, International Journal of Cognitive Informatics and Natural Intelligence. 2020;14:1.
26. Prriyadarshini AM, Devi RS. Detection of DDoS Attacks Using Supervised Learning Technique," Journal of Physics:

Conference Series. 2020;1716:012057-012069.

27. Azizan HA, Mostafa AS, Mustapha A, Foozy MFC, Wahab AHM, Mohammed and A. B. Khalaf, A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems, Annals of Emerging Technologies in Computing (AETiC). 2021;5(5):201-208.
28. Sahoo SK, Tripathy KB, Naik K, Ramasubbareddy S, Balusamy B, Khari M, Burgos D. An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks, IEEE Access. 2020;8:132502-132513.
29. Prasad DM, Babu PV, Amarnath C. Machine Learning DDoS Detection Using Stochastic Gradient Boosting, International Journal of Computer Sciences and Engineering. 2019;7(4):157-167,
30. Narote A, Zutshi V, Potdar A, Vichare R. Detection of DDoS Attacks using Concepts of Machine Learning, International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2022;10(VI):390-403.
31. Hariharan M, Abhishek HK, Prasad GB. DDoS Attack Detection Using C5.0 Machine Learning Algorithm, I.J. Wireless and Microwave Technologies. 2019;1:52-59.
32. Issa T, Tiemoman K. propose a method for detecting and mitigating Distributed Denial of Service (DDoS) attacks in real-time by using Software Defined Network (SDN) technologies, Bloom filters, and machine learning-based behavioral analysis of network traffic, International Journal of Advanced Computer Science and Applications (IJACSA). 2019;10(9)406-412.
33. Polat H, Polat O, Cetin A. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models," Sustainability. 2020;12:1036-1051.
34. Dong S, Sarem M. DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," IEEE Access. 2020;8:5039-5048.
35. Sindian S, Sindian S. An Enhanced Deep Autoencoder-based Approach for DDoS Attack Detection, WSEAS Transactions on Systems and Control. 2020;15:716-724.
36. Tennakoon C, Fernando S. Deep learning model for distributed denial of service (DDoS) detection, International Journal of Advanced and Applied Sciences. 2021;9(2):109-118.
37. Ingle A, Gour A, Kshirsagar K. DDoS Attack Detection Algorithms Based on Pattern Classification and Machine Learning, Journal of University of Shanghai for Science and Technology. 23(2):132.
38. Shieh CS, Lin WW, Nguyen TT, Chen CH, Horng MF, Miu D. Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model," Applied Sciences. 2021;11:5213-5225.
39. Xinlong L, Zhibin C. DDoS Attack Detection by Hybrid Deep Learning Methodologies," Security and Communication Networks. 2022;7866096-7866103.
40. Ahmed S, Khan AZ, Mohsin MS, Latif S, Aslam HM, Adil M, Najam Z. Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron, Future Internet. 2023;15:76-99.
41. Aktar S, A. Nur Y. Towards DDoS attack detection using deep learning approach," Computers & Security. 2023;129:103251.
42. Elsayed SM, Le-Khae NA, Soumyabrata D, Jurcut DA. DDoSNet: A Deep-Learning Model for Detecting Network Attacks," IEEE; 2020.
43. Doriguzzi-Corin R. Miller S. Scott-Hayward S. Martinez-del-Rincon J, Siracusa D. LUCID: A Practical Lightweight Deep Learning Solution for DDoS Attack Detection, IEEE Transactions on Network and Service Management. 2020;2971776-2971790.
44. Najafimehr M, Zarifzadeh S, Mostafavi S. A hybrid machine learning approach for detecting unprecedented DDoS attacks, The Journal of Supercomputing. 2021;78:8106-8136.
45. Aslam M, Ye D, Tariq A, Asad M, Hanif M, Ndzi D, Chelloug AS, AM. Elaziz AAM. Al-Quaness and F. S. Jilani, "Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT, Sensors. 2022;2697-2724.
46. Saghezchi BF, Mantas G, Violas AM, OdMA. Duarte and J. Rodriquez, In this paper, the data is collected, and efficient ML is built 4.0 CPPSs," Electronics. 2022;11:602-615.
47. Balasubramaniam S, Vijesh Joe C, Sivakumar TA, Prasanth A, Kavitha V,

Satheesh Kumar K, Dhanaraj RK. Optimization Enabled Deep Learning-Based DDoS Attack Detection in Cloud Computing, International Journal of Intelligent Systems. 2023;2039217 – 2039233:02.

48. Pandian PA, Smys S. DDOS ATTACK DETECTION IN TELECOMMUNICATION NETWORK, Journal of Ubiquitous Computing and Communication Technologies (UCCT). 2019;1(1):33-44.

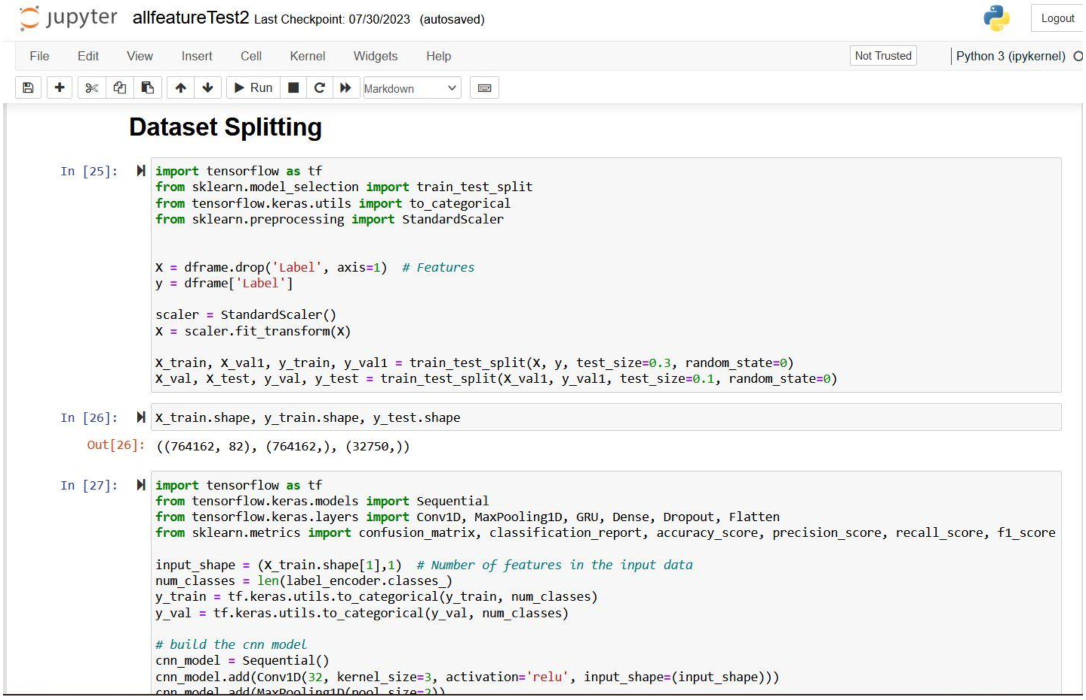
49. Sharafaldin I, Lashkari HA, Hakak S, Ghorbani AA. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, in 2019 International Carnahan Conference on Security Technology, Chennai, India; 2019.

50. LaMorte WW. Correlation and Linear Regression," 07 October 2021. Available:https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_correlation-regression/bs704_correlation-regression2.html.

51. Alghazzawi D, Bamasag O, Hayat O, Ashgar ZM. Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," Applied Sciences. 2021;11634-11656.

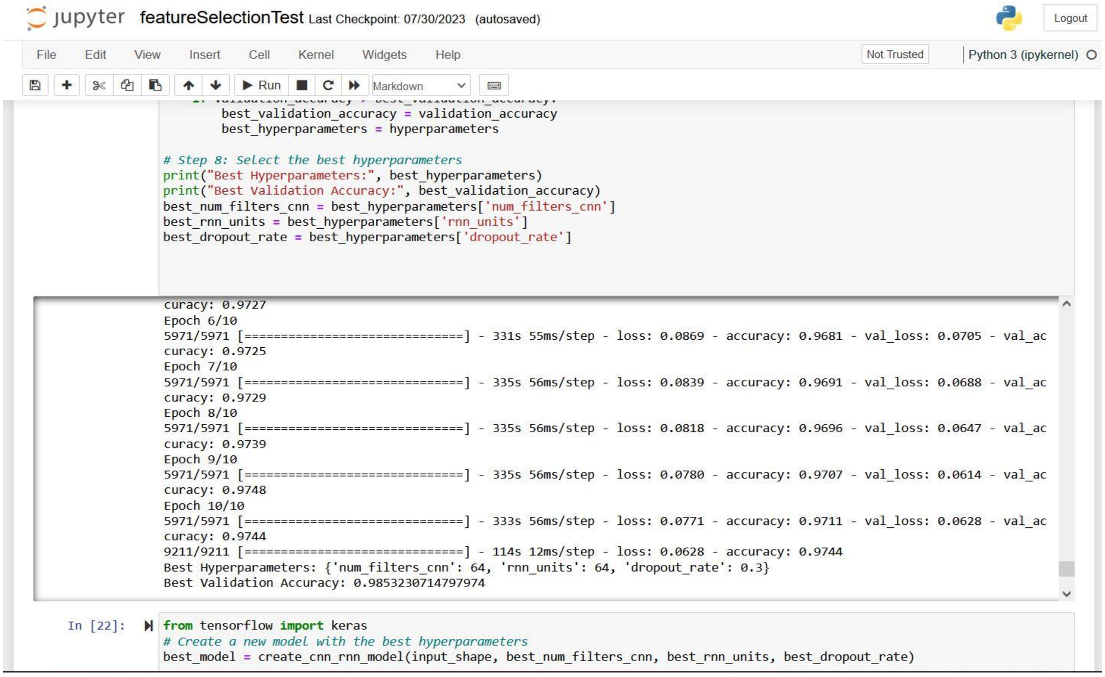
APPENDIX A

MODEL IMPLEMENTATION (INITIAL AND OPTIMISED)


jupyter allfeatureTest2 Last Checkpoint: 07/30/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown

Dataset Splitting

```
In [25]: import tensorflow as tf
         from sklearn.model_selection import train_test_split
         from tensorflow.keras.utils import to_categorical
         from sklearn.preprocessing import StandardScaler


         X = dframe.drop('Label', axis=1)  # Features
         y = dframe['Label']

         scaler = StandardScaler()
         X = scaler.fit_transform(X)

         X_train, X_val1, y_train, y_val1 = train_test_split(X, y, test_size=0.3, random_state=0)
         X_val, X_test, y_val, y_test = train_test_split(X_val1, y_val1, test_size=0.1, random_state=0)

In [26]: X_train.shape, y_train.shape, y_test.shape

Out[26]: ((764162, 82), (764162,), (32750,))

In [27]: import tensorflow as tf
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Conv1D, MaxPooling1D, GRU, Dense, Dropout, Flatten
         from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score, f1_score

         input_shape = (X_train.shape[1],1)  # Number of features in the input data
         num_classes = len(label_encoder.classes_)
         y_train = tf.keras.utils.to_categorical(y_train, num_classes)
         y_val = tf.keras.utils.to_categorical(y_val, num_classes)

         # build the cnn model
         cnn_model = Sequential()
         cnn_model.add(Conv1D(32, kernel_size=3, activation='relu', input_shape=(input_shape)))
         cnn_model.add(MaxPooling1D(pool_size=2))
```


jupyter featureSelectionTest Last Checkpoint: 07/30/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown

```
            best_validation_accuracy = validation_accuracy
            best_hyperparameters = hyperparameters

    # Step 8: Select the best hyperparameters
    print("Best Hyperparameters:", best_hyperparameters)
    print("Best Validation Accuracy:", best_validation_accuracy)
    best_num_filters_cnn = best_hyperparameters['num_filters_cnn']
    best_rnn_units = best_hyperparameters['rnn_units']
    best_dropout_rate = best_hyperparameters['dropout_rate']
```

```
curacy: 0.9727
Epoch 6/10
5971/5971 [==============================] - 331s 55ms/step - loss: 0.0869 - accuracy: 0.9681 - val_loss: 0.0705 - val_ac
curacy: 0.9725
Epoch 7/10
5971/5971 [==============================] - 335s 56ms/step - loss: 0.0839 - accuracy: 0.9691 - val_loss: 0.0688 - val_ac
curacy: 0.9729
Epoch 8/10
5971/5971 [==============================] - 335s 56ms/step - loss: 0.0818 - accuracy: 0.9696 - val_loss: 0.0647 - val_ac
curacy: 0.9739
Epoch 9/10
5971/5971 [==============================] - 335s 56ms/step - loss: 0.0780 - accuracy: 0.9707 - val_loss: 0.0614 - val_ac
curacy: 0.9748
Epoch 10/10
5971/5971 [==============================] - 333s 56ms/step - loss: 0.0771 - accuracy: 0.9711 - val_loss: 0.0628 - val_ac
curacy: 0.9744
9211/9211 [==============================] - 114s 12ms/step - loss: 0.0628 - accuracy: 0.9744
Best Hyperparameters: {'num_filters_cnn': 64, 'rnn_units': 64, 'dropout_rate': 0.3}
Best Validation Accuracy: 0.9853230714797974
```

```
In [22]: from tensorflow import keras
         # Create a new model with the best hyperparameters
         best_model = create_cnn_rnn_model(input_shape, best_num_filters_cnn, best_rnn_units, best_dropout_rate)
```

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle5.com/review-history/112853