National College of Ireland

A Deep Learning approach for Cyber Threat Detection using Intrusion Detection Data

MSc Research Project

Data Analytics

Mayuri Bhogate

Student ID: x22220453

School of Computing

National College of Ireland

Supervisor: Mr. Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Miss Mayuri Ganpat Bhogate……

Student ID: ……x22220453……..……

Programme: ……MSc Data Analytics …… **Year:** ……2023-24……..

Module: ……MSc Research Project..……..……

Supervisor: ……Mr. Jaswinder Singh……..……

Submission Due Date: ……12/08/2024……..……

Project Title: A Deep Learning approach for Cyber Threat Detection using Intrusion Detection Data.

Word Count: ……8465…… **Page Count**……22……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Mayuri Bhogate ……

Date: ……11/08/2024……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Learning approach for Cyber Threat Detection using Intrusion Detection Data

Mayuri Bhogate

22220453

Abstract

This paper is a systematic review of the deep learning models and their effectiveness in detecting emerging cyber threats, specifically focusing on Distributed Denial of Service (DDoS) attack using balanced data obtained from Kaggle. CNN, LSTM, BiLSTM, and a novel methodology proposed in the study namely Stacked BiLSTM with Self-Attention Mechanism are analyzed. This research is motivated by the fact that the current and emerging cyber threats are complex and usually escape both conventional detection systems. The primary purpose is to optimize the detection accuracy and improve adaptability of cyber threat detection system. It proved that the proposed model, Stacked BiLSTM with Self-Attention Mechanism has better results compared to other models, where classification accuracy is up to 98.39% while CNN achieved 58%, LSTM achieved 55%, and BiLSTM achieved 96%. Based on these findings, there is an indication that with the help of self-attention mechanisms, it is possible to pay more attention to key elements and enhance real-time detection in various environments. By offering a detailed and effective solution to cyber threat identification, this research brings an important value to the literature and has clear application in improving counter measures against complex cyber threats.

1 Introduction

Computer security threats are much more complex and common, constituting a real problem to cybersecurity all over the world. It is often the case that the more advanced and sophisticated the threat the traditional technologies are not capable of detecting these threats thereby highlighting the need for more advanced detection methods. The World Economic Forum (2024) stated that, the nature of cyber threats would become more complex hence attackers will use artificial intelligence in phishing, deepfake, and zero-day vulnerability exploitation. Proliferation of digital and IT systems as well as internet of things also increases the exposure factor which makes it more critical to detect emerging cyber threats (Acronis, 2024).

Some papers have focused on the use of ML and DL in cybersecurity issues and techniques. Mijwil, Salem, and Ismaeel (2023) give pertinent descriptions on the role of ML, DL in identifying threats and averting cyber incidences. It is in agreement with their systematic review that deep learning algorithms like CNN and RNN outperform other methodologies of ML in cyber risk identification and prevention. This proves the fact that there is always need for change and development concerning the information security technologies.

https://www.acronis.com/en-eu/blog/posts/cyber-security-trends/#XOVA46A6KE

In another study, Okoli et al. (2024) discuss about the use of ML in cybersecurity and how well ML is working in the identification of threats and protection mechanism. They have stressed the need to modify the current ML algorithms since the security threat is ever-evolving. Moreover, Shaukat et al. (2020) compared different categories of ML for classification of cybers threats and evaluated that DBNs are more accurate in terms of precision and recall as compared to other methods such as SVM and DT.

This study is prompted by the growing nature of cybercrimes that remain undetected by conventional anti-virus software's. This means that the threats can manifest themselves and cause optimal damages to be inflicted on valuable data and other possessions by attackers, who are also likely to change their tactics at regular intervals because of the vulnerability of the target networks. The use of Artificial Intelligence and Automation in cybersecurity is identified to be increasingly significant by Acronis (2024) and is credited with the responsibilities of threat identification and mitigation through the analysis of large amounts of data in real time to look for signs of intrusion. Thus, the goal of this work is to improve cyber threat detection systems based on the results of the development and assessment of more sophisticated deep learning models.

The literature review has also shown that deep learning has the potential of addressing cybersecurity threats, except for ranking them based on prior knowledge of the threat database alone. For instance, Ashraf et al. (2022) developed a smart framework using a combination of Random Forest (RF) and Multi-Layer Perceptron (MLP) models for the detection of cyber-physical and satellite system security threats in arguing for simple and dynamic cybersecurity systems. In this context, this work expands the literature by proposing a new Stacked BiLSTM with Self-Attention Mechanism, which is expected to overcome these issues with higher detection rates and flexibility.

Research Question and Objectives:

The primary research question proposed in this study is: "To what extent does integrating a Self-Attention Mechanism into a Stacked BiLSTM model improve the detection accuracy and adaptability of cyber threat detection systems compared to traditional deep learning models?"

1. Examine the current performance of traditional deep learning model in cyber threat detection: CNN, LSTM, BiLSTM.

2. Designing a unique Stacked BiLSTM with self-attention mechanism to increase accuracy and adaptability.

3. Implementing and training the models using customized balanced dataset obtained from Kaggle.

4. Evaluating the performance of the proposed model against the conventional models using measures like accuracy, confusion matrix, classification report, specificity, and sensitivity.

To this end, it is crucial to point out that this research has been based on a customized dataset that has been developed from the given DDoS flows obtained from different public IDS datasets, including CSE-CIC-IDS2018-AWS, CICIDS2017, and the CIC DoS dataset (2016). These datasets were generated employing various experimental DDOS traffic generation tools in various years which add more variation in it. The extracted DDoS related flows were the flows which were further amalgamated in this direction with the respective separate benign flows that were also extracted from the corresponding base data sets. The last balanced dataset is (12,794,627) rows with each row including a forward or reverse flow, and therefore containing (84) columns features. Some of the most important features are FlowID, Timestamp, Fwd Seg Size Min, Src IP, Dst IP, Flow IAT Min, Src port, Tot Fwd Pkts, Init Bwd Win Bytes.

Although this approach brings variance and robustness the results may differ because of the differences in the data collection and experimental procedures of the datasets. Finally, the study particularised on DDoS attacks and more research needs to be conducted to analyse the probability of the results on other types of cyber threats. This dataset can be accessed publicly at Kaggle using below link:

https://www.kaggle.com/datasets/devendra416/ddos-datasets

This report is organized as follows: This paper is organized into Section 2, Related Work, where the existing literature about employing deep learning in cyber security is surveyed. Section 3 draws on Research Methodology and section 4 on Design Specification. The Implementation is described in Section 5, which speaks about the general architecture and deployment of the models. Section 6, Evaluation, analyzes the model's accuracy and makes a comparison between the programs. Finally, the Conclusion is provided in Section 7 in which the recommendations for future research are given.

2 Related Work

2.1 Traditional Deep Learning Models in Cyber Threat Detection

Yang et al. (2022) conduct the systematic literature study on the anomaly-based network intrusion detection techniques and datasets. It features the analysis and evaluation of application domains, data preprocessing, detection methods and ways, as well as metrics of assessments. Reviewing deep learning approach to IDS, the paper acknowledges the applicability of CNNs and RNNs but also emphasizes the key difficulties such as low ratio of instances in the dataset, and the absence of clear guidelines for data preprocessing. As for the future work, the researchers are advised to concentrate more on the datasets that are much stronger and inviting more enhancements in the feature engineering to yield much better results that will make the detectors much more accurate and flexible. Ashiku and Dagli (2021) present Deep learning-based Network Intrusion Detection System (NIDS) that uses the Convolutional Neural Network. What also worth mentioning is that the chosen model is based on the UNSW-NB15 dataset that covers modern network behaviors and includes different synthetically generated attacks. The CNN architecture proposed has then been regularized and hyperparameterized to achieve increased performance. The model was able to achieve detection accuracy of 95.6% on a user-defined dataset, which improves the existing Deep Learning based NIDS. The future work will be to focus on the approaches such as transfer learning and bootstrapping to cope with zero-day attack and deal with the problems of imbalance class.

2.2 Hybrid approaches for Threat Detection

Hybrid Model by Mehmood et al. (2022) of Network Intrusion Detection using Support Vector Machines (SVM) and Adaptive Neuro-Fuzzy Inference System (ANFIS) is discussed. The methodology includes the data preconditioning in which the NSL-KDD dataset is transformed and subjected to min-max normalization; and second, using the Random Forest Recursive Feature Elimination for feature selection. The efficiency of the hybrid model shows that it has a detection rate of 99.3% and a means square error of 0. 084964, in turn, enhances the detection performance for the different types of attack. Future work includes the improvement of deep learning classifiers' accuracy and focusing on the computational efficiency that appears in the real-time application of the proposed solution.In Shaukat, Luo, and Varadharajan (2022), the authors present a new approach that aims at improving the anti-robustness of deep learning based Malware detection systems. The authors test ten types of adversarial attacks on the detectors based on artificial neural networks, as well as the authors' new approach involving two robust attack types (rFGSM and Grosse). Thus, the experiments prove that the proposed combination attack has a much lower evasion rate, which is on average 12% for VirusShare and

18% for VXHeaven when compared to other models. The future work present to continue the study of the hybrid defense mechanisms to enhance the performance of the model against various types of adversarial strategies. Ashraf et al. (2022) formulated an intrusion detection system called RFMLP, which is an ensemble of Random Forest and Multilayer Perceptron algorithms for cyber-physical and satellite systems security. The paper compares the model's results with three data sets including KDD-CUP 99, NSL-KDD, and STIN and highlights that RFMLP show better performance than conventional models consisting of SVM, logistic regression with high accuracy in identifying DoS or U2R attacks. Strengthening ideas for the future work include improving the feature reduction step using more sophisticated methods and applying the data augmentation method such as SMOTE to increase the detection rate. In this paper, Bouchama and Kamal (2021) described the improving of detection of cyber threats by modeling the network traffic successfully using machine learning. It uses supervised learning, unsupervised learning, and a combination of both neural networks, support vector machines and isolation forests by searching for anomalies in the network traffic. The evaluation entails impressive enhancements in the accuracy of detection as well as a decrease in the number of False Positives when compared to conventional rule-based approaches. The authors also underline that model adaptation should be constant in order to address changing threats and recommend further research to incorporate explainability measures into the model and resolve issues with dynamic environments.

2.3 Challenges and Future Discussions in Cyber Security

Ahsan et al., (2022) elaborate on the different uses of machine learning in cybersecurity with the view of improving threat identification. They assess the prior conventional models such as SVM, decision trees, and deep learning techniques, that depict the kind of duty they play to detect and prevent the cyber-attack. From the review, it is seen that though these methods are efficient, they have some issues including data imbalance and dynamic threats. As for the future work, the authors recommend searching for the hybrid solutions that would use different techniques to increase the aspect of adaptability and the rates of detection in the context of the more dynamic scenarios. Future studies need to focus on using more current data and exploring more sophisticated feature extraction and selection techniques.Aslan et al. (2023) provide a detailed discussion on security threats, risks, types of threats, attacks, and countermeasures with focus on new threats arising with IoT and Cloud Computing. The authors have raised concern concerning the effectiveness of conventional security systems and noted the importance of incorporating Deep learning, Machine learning, and Blockchain among other contemporary innovations that enhance risks freeness detection. Future works should aim at enhancing the resistance of these technologies against evasion methods and in creating enhanced models including forms of a hybrid to mitigate persistent threats
Dash et al. (2022) provide a detailed description on the use of Artificial Intelligence in the context of cybersecurity, and the advantages as well as issues associated with the field. The authors share more details on how the AI techniques are useful specifically in regards to machine learning in intrusion detection in which the detection, prediction, and response durations in the face of cybercrimes are boosted. The study describes the AI's upside in enhancing security measures mentioned earlier; at the same time, the study discusses risks like the dual-use nature, where hackers use AI for more elaborate threats. The authors recommend the following research directions to manage these risks and enhance federated learning for preserving AI confidentiality in cybersecurity.The detailed literature review on machine learning and deep learning in cybersecurity is presented by Mijwil, Salem, and Ismaeel (2023). This paper investigates the uses of the identified techniques towards fighting cyber threats, specifically in the context of intrusion detection systems and the detection of malware. Still, the authors also underscore the fact that despite an applicability of the traditional models, the newer ones such as deep neural networks (DNNs) and the recurrent neural networks (RNNs) are more

accurate in detecting complex cyber threats. Future research should thus build upon the current work by attempting to increase these models' ability to modify with the appearance of new or evolving threats by implementing target schemes of hybrid versions and analyzing larger sets of data. Okoli et al. (2024) offer a description of the deployment of machine learning (ML) in cybersecurity concentrating on threat identification and protection strategies. The authors examine different strands of ML practices including the most common supervise and unsupervised learning and underline the ability to solve cyber security threats. This paper demonstrates that use of ML, and more specifically, deep learning, increases the level of detection and optimizes it to outperform previous techniques. However, the challenge like adversarial attacks and data bias are there. The discussion on future work indicates the improvement of the properties of current and new models by update and ethic incorporation as the countermeasures for the adversarial attack on the defence system.

2.4 Emerging Techniques and advance models

In Yaseen (2023) the focus is on the threat detection and response systems in cybersecurity with the emphasis made on the utilization of AI models especially the use of deep learning as a detection tool. This process engages the use of AI models including, CNNs, LSTMs and GANs to monitor and counter threats as they happen. The outcomes also reveal an increase in the detection accuracy and response time compared with regular techniques. Thus, author was designated that in future, it is necessary to refine them to identify new types of cyber threats, develop the combined use of the models, and improve the explainability of AI-based systems for creating the more qualitative and effective cybersecurity systems.A survey on the application of NNs in IDS is presented by Drewek-Ossowicka, Pietrołaj, and Rumiński in 2021. The method used is the review of literature with an emphasis on enhancing IDS using NN architectures such as MLP, CNN, RNN, and or a combination of these architectures. The authors divide different NN approaches under the categories depending on the efficiency of cyber threats detection. The survey also reveals that the deep learning models are most effective in dealing with sophisticated and dynamic tactics of an attack. The future work in this regard is required to focus on the adversarial attack problems and more efficient training data set to overcome IDS drawbacks.Aldarwbi, Lashkari, and Ghorbani (2022) given a research proposal for a new NIDS called "The Sound of Intrusion." The methodology works on the concept of converting the network traffic flows to sound and then using deep learning including CNN, LSTM, DBN for detection of intrusions. Experiments were carried on the NSL-KDD and CICIDS2017 datasets using the proposed system, and the detection accuracies obtained were 84. 82% and 99. 41%, along with low false alarm percentages. Analysing the data, it is concluded that the consideration of network traffic in the aspect of audio detection can enhance its accuracy level. In future work, more adequate models of signal formation and more complex deep learning classifiers will be examined to refine the method of anomaly detection.

3 Research Methodology

The present study employed a systematic approach for formulating and testing deep learning models against DDoS attacks as show in the Fig.1

3.1 Data Collection

The dataset that was used was a selective collection of DDoS flows from different open-source IDS datasets, the CSE-CIC-IDS2018-AWS dataset, CICIDS2017, and the CIC DoS dataset 2016. This combined dataset is basically downloaded from Kaggle that consists of equal

proportion of DDoS attack traffic and benign traffic. Because of the large number of records, only a sample of 200, 000 records were considered for training and testing.

3.2 Data Preprocessing

The primary steps in data preprocessing include Data cleaning to remove missing values, duplicates, and irrelevant or noisy features.

1. **Data Cleaning:**

   Features with a missing percentage greater than 50% were discouraged and excluded; while records having a percentage of missing values not exceeding 5% were either imputed or omitted depending on their importance. To address the issue of faulty data, there were specific columns with the infinite values that were effectively handled.

Fig 1. Overview of Research Methodology

2. **Feature encoding and Scaling**

   Categorical variables were encoded using the `LabelEncoder` to convert it to the numerical data which is suitable for training. Numerical features were scaled using feature scaling with `MinMaxScaler` such that, all the features were on equal stand as far as training the model was concerned.

3. **Feature Selection:**

   Irrelevant features that would not be useful in the performance of the predictive modelling were dropped in order to relieve the dimensionality. Evaluation of top fifteen feature importance was done employing the 'ExtraTreesClassifier' which aided in distinguishing the most crucial features to the target variable.

4. **Data Splitting**

   The data set was then separated accordingly into training, validation, and test sets at a ratio of 60:20:20 respectively by applying stratified sampling to ensure the distribution

of the classes in each set was representative. This step allowed the models to have an equal distribution of every class when training and testing the model.

3.3 Model Training

The study adopted conventional deep learning paradigms: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM). Moreover, a Stacked BiLSTM with Self-Attention Mechanism was introduced and tested in the research.

1. **Convolutional Neural Network (CNN):**

   The CNN model architecture consisted of Conv1D, Flatten, Dense, and Dropout layers. ReLU was used for all hidden layers except for the output layer, where SoftMax was utilized.

2. **Long Short-Term Memory (LSTM):**

   The components of LSTM model included LSTM and Dense and the Activation function used was the SoftMax function for the output layer.

3. **Bidirectional LSTM (BiLSTM):**

   The Final architecture of the BiLSTM model consists of Bidirectional LSTM along with Dense layers and Sigmoid activation.

4. **Stacked BiLSTM with Self-Attention:**

   The novel model proposed, Stacked BiLSTM with Self-Attention, contained multiple Bidirectional LSTM layers, an attention layer, Dense and Dropout layers and ReLU activation for the hidden layers and Sigmoid activation for the output layer. The proposed model was trained with the Adam optimizer and categorical crossentropy loss.

Model training was done using the training set while hyper-parameter optimization was done based on the results of validation set. Methods like drop out and early stopping, decreasing the learning rate were used to avoid overfitting and improve the training speed. All the models were trained for 10 epochs in the case of training while in case of testing all the models were tested with a batch size of f 64.

3.4 Model Evaluation

The models' performance is assessed according to accuracy, confusion matrix, classification report, specificity, sensitivity to allow comparison of the results and to determine the robustness of the reliability of the results.

4 Design Specification

The Design Specification lays out the structure, approaches, and frameworks used in the actualisation of the proposed deep learning models with an emphasis on the proposed Stacked BiLSTM with Self-Attention Mechanism.To classify the DDoS attacks, several deep learning models used in the research are: CNN, LSTM, BiLSTM, and the newly developed S-BiLSTM-

SAM. Specifically, each of the models was designed with specific target strengths in the areas of feature extraction, temporal dependency learning, and sequence attention.

4.1 CNN

The Convolutional Neural Network (CNN) model used in this study aims at identifying the DDoS attacks and the features of the sequence data. As illustrated in Fig. 2 The model starts with the input layer for preprocessed sequence data and the input shape is defined by the training dataset. The feature extraction step is conducted by a Conv1D layer which applies 8 filters and a stride of 3 to learn local patterns in data. The reasons for applying Conv1D are related to their ability to analyze sequential data, which allows detecting patterns related to cyber threats in the context of time-series. ReLu activation function is used here to introduce non-linearity as the network can learn more features. The output from this Conv1D layer is in the form of feature map and then a Flatten layer is incorporated to convert the feature map into a one-dimensional vector. This transformation is done in order to feed the data into the fully connected dense layers successfully.

Layer Type	Layer Name	Output Shape	Number of Parameters	Activation Function
Input Layer	_	(None, X_train.shape[1],1)	_	_
Conv1D Layer	conv1d	(None,72,8)	32	ReLU
Flatten Layer	flatten	(None,576)	0	_
Dense Layer	dense_1	(None,6)	3462	ReLU
Dense Layer	dense_2	(None,6)	42	ReLU
Dropout Layer	dropout	(None,6)	0	_
Output Layer	dense_3	(None,2)	14	Softmax

Fig.2 CNN Model Architecture

The flattened vector is fed through two Dense layers with 6 units and ReLU activation. These dense layers are used to learn the abstract features of the data and ReLU helps the model to learn non-linearity. The reason behind the choice of adopting two dense layers is to enhance the model depth thereby helping it capture details that might be difficult to be seen within a single dense layer.

To address the issue of overfitting a Dropout layer with dropout rate set to 0.4. The last layer of the model is then a Dense layer with two neurons, the number of classes we have, which outputs predictions. SoftMax is applied to the layer to estimate class probabilities for efficient classifying of each input sequence. The model is trained using categorical crossentropy as the loss function and SGD optimizer with the learning rate of 0. 00001. The entire architecture is written in TensorFlow and Keras, though Python is used because of its huge number of ML libraries.

4.2 LSTM

LSTM networks are effective when the data contains a series of values ordered in time and the context of the data points is important, making the use of LSTM networks perfect for this cybersecurity application. As seen in Fig. 3, the model is started with an LSTM layer in which 12 units are used. The return_sequences parameter is set to False meaning that the LSTM layer

will only return the last state and not a sequence of outputs. This is suitable for classification problems where the decision can be made given the last observation in a sequence. The input shape is defined corresponding to the dimensions of the training data so that the model processes the sequence data properly.

Layer Type	Layer Name	Output Shape	Number of Parameters	Activation Function
Input Layer	_	(None, X_train.shape[1],1)		
LSTM Layer	lstm	(None,12)	672	
Output Layer	dense	(None,2)	26	Softmax

Fig.3 LSTM Model Architecture

Next is the Dense layer that acts as the Output Layer after the LSTM layer have been incorporated. This layer is made of 2 neurons as this layer is used to predicting the class (for example benign or DDoS attack). The Softmax activation function is used here in converting the output of the Dense layer to probability distribution over the classes. This probabilistic output is very useful for classification type problems because it gives the model a measure of confidence that it has classified an instance correctly.

The model used categorical crossentropy loss for its compilation. Stochastic Gradient Descent (SGD) is used with a learning rate of 0. 0001 and is selected as the optimizer. This makes SGD more effective in this regard due to its simplicity and efficiency in dealing with large data sets.

4.3 BiLSTM

The model used in this research known as Bidirectional Long Short Term (BiLSTM) is aimed at enhancing the recognition of DDoS attacks due to its ability to consider dependencies in both forward and backward way in sequence data. Due to this forward and backward information processing, BiLSTM networks are well suited to applications where context from future and previous inputs is critical for creating precise forecasts.

Layer Type	Layer Name	Output Shape	Number of Parameters	Activation Function
Input Layer	_	(None, X_train.shape[1],1)	_	_
BiLSTM Layer	bidirectional	(None,10)	480	_
Output Layer	dense	(None,2)	22	Sigmoid

Fig.4 BiLSTM Model Architecture

As shown in Fig.4, Bidirectional LSTM is the first layer of the model and it includes bidirectional LSTM layer with 5 units. The bidirectional wrapper enables the model to analyze the input sequence data both forward and backward, which makes the model more perceptive to the total context. The return_sequences parameter is set to False this implies that the layer will return only the final hidden state instead of all the hidden states. This configuration is suitable for classification problems where the output for the classification alone is sufficient to make a decision.

The next layer of the architecture is the BiLSTM layer, and after this, we have the Dense layer which is the Output layer. This layer comprises of 2 neurons and these are the classes which in this example are benign or a DDoS attack. This is where the activation function comes in; in this case, the sigmoid function is applicable since they bring out the probability of the input vector belonging to a binary class.

The model is compiled using categorical crossentropy loss function since it is commonly used in multi-class classification but is easily applicable to binary classification even when it comes to one hot encoded labels. To appropriately fine-tune this model, the appointed optimizer is Stochastic Gradient Descent with the learning rate set at 0. 01. The increase in the learning rate helps to increase the speed of the convergence particularly in the models with lesser units in their configurations, though this calls for constant supervision in an effort to ensure that the models do not over-shoot the optimal solution. The model is implemented using TensorFlow and Keras to take advantage of the flexibility of the Python language, which hosts numerous deep learning frameworks.

4.4 Stacked BiLSTM with Self-Attention mechanisms

The proposed method Stacked BiLSTM with Self-Attention Mechanism is versatile and would enhance the cyber threat detection accuracy and flexibility. This architecture builds upon BiLSTM networks and incorporates Self-Attention to make the model's predictions based on significant input sequence parts and ignore the less relevant ones. This combination is especially effective in the tasks that require the awareness of the context of past and future inputs like in the pattern analysis of cybersecurity data.

Layer Type	Layer Name	Output Shape	Number of Parameters	Activation Function
Input Layer	_	(None, X_train.shape[1],1)	_	_
BiLSTM Layer 1	bidirectional_1	(None,74,24)	2688	_
BiLSTM Layer 2	bidirectional_2	(None,74,24)	2688	_
Self-Attention Layer	attention	(None,24)	48	_
Dense Layer	dense_1	(None,10)	250	ReLU
Dense Layer	dense_2	(None,6)	66	ReLU
Flatten Layer	flatten	(None,6)	0	_
Dropout Layer	dropout	(None,6)	0	_
Output Layer	dense_3	(None,2)	14	Sigmoid

Fig.5 Stacked BiLSTM with Self attention mechanism architecture

As shown in Fig.5, the architecture consists of two Bidirectional LSTM layers stacked with one on top of the other these are defined with 12 units. Bidirectional LSTM performs forward and backward passes on the input data, thus making the model exercise its capability in relation to the entire sequence. The stacking of these layers creates more layers and the model is able to learn temporal dependencies of a higher level. The return_sequences parameter is set to True for both layers; therefore, each LSTM layer will output all the hidden states for the sequence in addition to the final state. This is crucial for feeding the whole of the sequence to the next Self-Attention component.

It is, therefore, significant that the main novel of this architecture is called the Self-Attention layer. This can be attributed to a special mechanism of attention that is applied in the generations of this model to compute sequence specific context vector and thus selectively attend the important parts of the sequence. Especially, the attention layer computes weight vector (att_weight) for each time step in the sequence. These weights are derived from the input by first passing it through a non-linearity (tanh), multiplying it with learned parameters and then scaling it by a softmax function. Using the computed attention scores, subsequent matrices and vectors are then created in such a way that it calculates the segmentation of the input

sequence through a weighted sum, which in turn means that the model can pay more attention to this time step as opposed to the other time steps depending on its importance in arriving at the final output. This mechanism will be especially important when working with a large sequence in which only a fragment of the code contains important information. The next step after the Self-Attention mechanism that has focused on handling the crucial elements of the sequence is feeding it through a subsequent chain of densified layers. The input Dense layer has 10 neurons and the activation function is ReLU, the second Dense layer has 6 neurons and the activation function is also ReLU. These layers are also for the fine tuning of the features which are extracted by BiLSTM and Self-Attention layers. The ReLU activation function is then applied to bring non-linearity which means that the model can capture the non-linear interactions to the features.After the final dense layer, the output passes through another layer which convert the output from a two-dimensional array to one-dimensional array. This Flatten layer helps to shape the data for the final classification process by making the used tensor multi-axially shaped for the output layer. Nevertheless, one must put some measures in place to avoid overfitting, a step carried out by adding a Dropout layer with a rate of 0. 55 is added after the Flatten layer. In dropout, randomly some neurons are dropped out during training such that the model is not over-reliant on such neurons and so overfits to some features & hence has a better capability of overfitting to unseen data.The last hidden layer is a Dense layer that has 2 output neurons as we have restricted the output space to the two classes that range from benign to DDoS attack. The last layer uses the Sigmoid activation function to give a probability for each class which can enable the binary classification.The used loss function is categorical crossentropy, which is suitable for multi-class classification tasks, and the model is compiled with it. The Adam optimizer is used for training, which includes the adaptive learning rate methods and momentum and hence improves the general convergence to the optimization solution. As the measure of the model's performance in training and testing, accuracy is incorporated into the model. The stacked BiLSTM layers give the architecture the possibility to learn the long-distance temporal dependencies processing the input sequence in forward-backward and in multiple layers. This depth is important to achieve because it will enable the analyst to identify complex patterns that may be indicative of cyber threats. The integration of Self-Attention also introduces opportunities to the model in extracting the attending parts within the series to effectively and robustly transform even for extensive and noisy series where the starting point hence the crucial segments are located.

Specifically, the integration of BiLSTM coupled with Self-Attention is useful as the utilization of critical features in sequential data makes cyber threat detecting much easier. This is followed by regularization through Dropout, and the selection of the Adam optimizer that is relatively resistant to semi-random noise and thus makes the model more generalized.

5 Implementation

The implementation phase of this study aimed to design, educate, and assess the efficacy of the introduced deep learning models for identifying DDoS attacks. The final stage of implementation implies that the primary output created was a clean and normalized dataset ready for feature training. In the process of data preprocessing, the raw data was first cleaned removing any duplicate records and handling the missing values properly, the categorical

features which were nominal in nature were encoded and the numerical had been scaled, finally the entire dataset was properly split into the training, validation and test data set. It can also be said that such a transformation of the data provided all models with high quality and unified input.

Four deep learning models were developed and trained during this phase: A CNN model for extracting spatial features of the input data; An LSTM model to handle temporal dependent sequences in the data; BiLSTM for Bi-directional sequential data; and the proposed Stacking BiLSTMs with Self-Attention to improve the accuracy and to detect patterns from the relevant sections of the input sequence. Both models were trained and fine-tuned on the transformed dataset with the hyperparameters specifically changed with regards to the validation accuracy to avoid over fitting.

To quantify the performance of all these models, comprehensive evaluation metrics were created. Such evaluation metrics entailed accuracy, confusion matrix, classification report, specificity, and sensitivity. The given approach allowed providing comprehensive insights into every specific model to compare its advantages and disadvantages in the assessment of DDoS attacks accurately.

The implementation process started with the data preprocessing, which involved removing the unwanted columns and rows in the raw data set, converting categorical data into their numeric form, and normalizing the numerical data respectively. As a next step data was divided into training, validation and test sets. After this, each deep learning model was described and then developed in TensorFlow using Keras. CNN based model aims to extract spatial features of the input data, while LSTM for temporal features and BiLSTM for both temporal forward and backward features. The use of the novel Stacked BiLSTM with Self-Attention Mechanism was put into consideration to improve the detection of aspects of the input sequence.

The actual models were then built on the transformed dataset centered with the right hyperparameters selected, this was followed by adopting certain payable measures including early stopping and learning rate decay, to check on overfitting and expedite the modelling process. Finally, for each of the models, the performance was measured on the test set to obtain corresponding performance values. The final products were the models that were trained, performance metrics, and graphical representation of the models' performance in the identification of DDoS attacks.

Altogether, the outcomes of the implementation phase reveal that several deep learning models were successfully designed, trained, and tested for the purpose of DDoS attack detection. The application of modern apparatuses and approaches allowed obtaining credible outcomes and confirming the potential of the presented Stacked BiLSTM with Self-Attention Mechanism to improve the detection performance and flexibility.

6 Evaluation

The evaluation section of this study offers the categorization of the DDoS attack detection performance based on the results of the models that were tested and implemented in this research. The results are discussed in the context of the research question. Each sub-section addresses how the given experiment has been done, presenting their outcome.

6.1 Experiment 1: CNN Model

6.1.1 Accuracy Score:

```
1245/1245 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step
CNN Accuracy:  0.584416888398012
```

Fig.6 CNN model accuracy

The CNN model reached an accuracy of 58. 44% on the test set. This score reveals that the model approximately yielded a correct classification rate of about 58. 44 % of the instances of network traffic. However, this low accuracy reveals that the model has a shortcoming in accurately categorizing the traffic as benign and DDoS traffic.

6.1.2 Training and Validation Accuracy and Loss:

Training vs. Validation Accuracy: The training accuracy gains were obtained from approximately 55 percent to 60 percent for the epochs while the validation accuracy hovers between 56 to 58 percent. The difference between the training and validation accuracies indicate that the model is probably overfitting as it performs well on the training set while poorly on the validation set.

Training vs. Validation Loss: The training and the validation loss reduce progressively over the epochs as the model continues to learn. However, the validation loss is always lower than the training loss, while the accuracy trends may suggest that the model is underfitting the data. Based on this, the small difference between the validation loss and the training loss indicates that the actual problem the model is solving remains ambiguous and does not encompass the necessary features for proper DDoS detection.

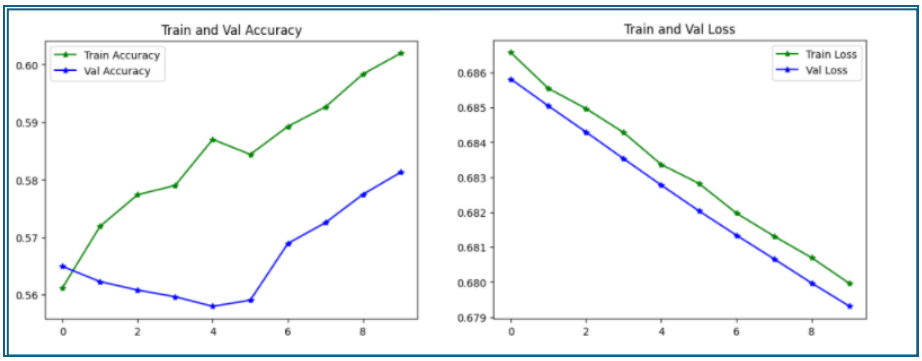

Fig.7 CNN model Training and validation Accuracy and Loss

6.1.3 Confusion Matrix:

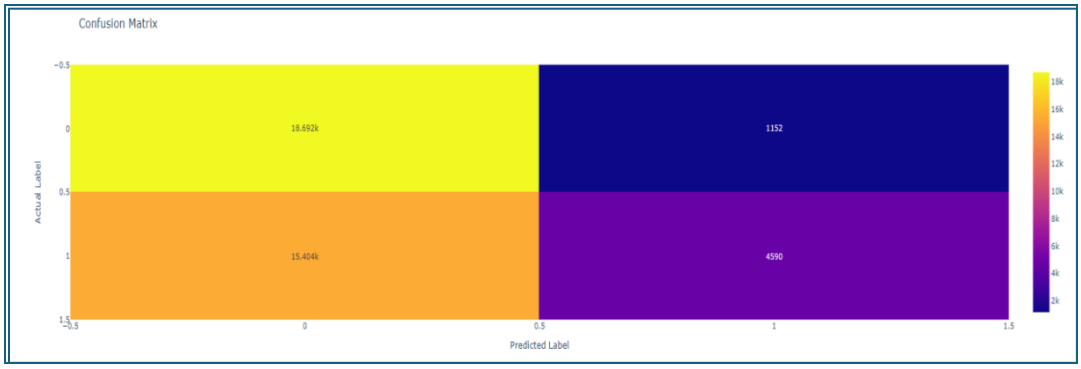

Fig.8 CNN Model Confusion Matrix

The matrix of confusion shows that among 18,692 actual cases of benign traffic, 1,152 were identified as DDOS attacks. On the other hand, the classifier flagged 4,590 out of 15,404 DDoS instances as the benign traffic. This high rate of misclassification especially when they are on the side of the negatives of DDoS detection is highly alarming as it depicts the inability of the model to label the malicious traffic correctly.

6.1.4 Classification Report:

These metrics suggest that, although the model is better at detecting the presence of DDoS traffic the recall for DDoS attack is low; meaning more over Malicious activities that originate from a DDoS attack will go unnoticed by the proposed model. Another important area where the presented approach is inadequate is the low F1-score for DDoS traffic.

6.1.5 Sensitivity and Specificity:

The sensitivity and specificity imply quite a large degree of bias in the model being studied. This model wants to be extremely sensitive to benign traffic, but it has very low sensitivity to DDoS traffic which is a critical problem when developing models for cybersecurity. The specificity values echo this imbalance, demonstrating how the model prioritizes benign traffic over accurately identifying DDoS attacks.

Classification Report :

	precision	recall	f1-score	support
0	0.55	0.94	0.69	19844
1	0.80	0.23	0.36	19994
accuracy			0.58	39838
macro avg	0.67	0.59	0.52	39838
weighted avg	0.67	0.58	0.52	39838

	class	sensitivity	specificity
0	0	0.229569	0.941947
1	1	0.941947	0.229569

Fig.9 CNN Model: Classification Report And Sensitivity and Specificity Report

6.2 Experiment 2: LSTM Model

6.2.1 Accuracy Score:

The proposed LSTM model was able to reach an accuracy of 54.79% accuracy. As seen in the evaluations of the classifications and accuracy percentages, the LSTM model was less accurate

than the CNN model, indicating that the LSTM model failed to correctly classify much of the network traffic especially in: distinguishing between the benign and DDoS traffic.

```
1245/1245 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step
LSTM Accuracy:  0.5479692755660425
```

Fig 10. LSTM Model accuracy

6.2.2 Training and Validation Accuracy and Loss

Training vs. Validation Accuracy: The training accuracy was a very volatile set of results that was characterized by having a bad dip towards the second half of the epochs, set against relatively flat though slightly declining validation accuracy. This gives a hint of fluctuations that the training process of the model can exhibit over the training epochs, especially since the LSTM model tends to be more complicated than the basic classifiers, or due to the improper hyperparameter tuning. The small distinction between the training and validation loss is an indication that the model may have overfitted and will not perform optimally on new data.

Training vs. Validation Loss: In the training process, it is observed that both the training and validation loss always decrease through epochs, and this is really expected because of the learning characteristics of the system. Nevertheless, the small difference between these two loss values may signal that the model is learning inefficiently, which, in turn, explains low accuracy. It is evident that in relation to the original model, the presented changes did not lead to substantial assumptions, which also indicates a potentially unsuitable architecture of the neural network model for the given task.

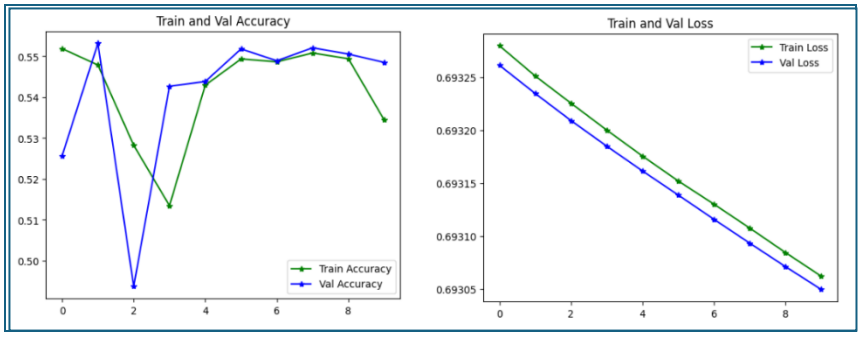

Fig11. LSTM Model: Training and Validation accuracy and loss

6.2.3 Confusion Matrix

The confusion matrix offer insight on how the model is doing relating to the classes used in the experiment. According to the matrix, the proposed model was able to accurately identify 5215 samples of benign traffic and 16515 samples of DDoS traffic. But at the same time, it classified 14,629 benign data points as DDoS and 3,379 DDoS data points as benign. The high value of misclassification specifically the false positive and false negative implies that there are major difficulties in achieving correct classification by the model.

Fig12. LSTM Model: Confusion Matrix

6.2.4 Classification Report:

The above metrics depict that even though the model has better recall for the DDoS traffic, it is not efficient in terms of precision and recall regarding benign traffic. Analyzing the results for benign traffic, one can conclude that the F1-score is rather low; this means that the model possesses good precision but low recall coefficient, which, in turn, leads to a large amount of both false positives and false negatives.

Classification Report :

	precision	recall	f1-score	support
0	0.61	0.26	0.37	19844
1	0.53	0.83	0.65	19994
accuracy			0.55	39838
macro avg	0.57	0.55	0.51	39838
weighted avg	0.57	0.55	0.51	39838

	class	sensitivity	specificity
0	0	0.830999	0.262800
1	1	0.262800	0.830999

Fig13. LSTM Model: Classification Report and Sensitivity and Specificity Report

6.2.5 Sensitivity and Specificity:

The sensitivity and specificity demonstrate that the LSTM model concentrates more on classifying traffic as DDoS, at the same time minimizing the classification of correct traffic. This imbalance may lead to a high ratio of false positives, which means that most of the normal traffic is considered malicious, which is unsuitable for real applications.

6.3 Experiment 3: BiLSTM Model

6.3.1 Accuracy Score:

In this case, the BiLSTM model recorded a high accuracy of 96.16 % percent as a result of the conducted evaluations. This high accuracy shows that the current model was very effective in the classification of the network traffic thereby enhancing the accuracy as compared to the previous models (CNN and LSTM).

```
1245/1245 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step
BILSTM Accuracy:  0.9615693558913601
```

Fig14. BiLSTM Accuracy

6.3.2 Training and Validation Accuracy and Loss:

Training vs. Validation Accuracy: From the provided graphs for both training and validation data set, we observe a constant and progressive increment of the accuracy rates with the increase of epochs to roughly 95% to 96%. The difference between training and validation accuracy is quite small, which can be interpreted as the model not overfitted on training data.

Training vs. Validation Loss: The training and validation loss decreases progressively, and the two plots are relatively parallel to each other. This means that the model which is being used is replicable and the minimization of the loss is also being done in the same way across both the training and the validation set.

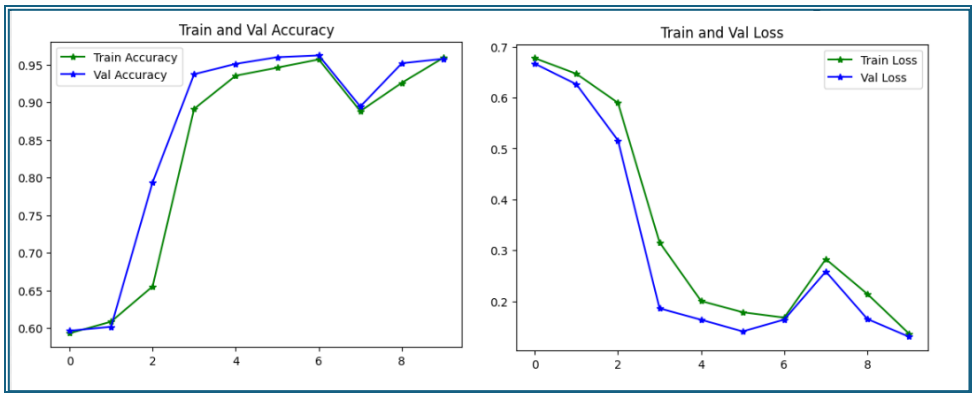

Fig15. BiLSTM Model Training and Validation accuracy and loss

6.3.3 Confusion Matrix:

Among the total of 18,686 benign samples, 373 instances were wrongly identified as DDoS, while 18,313 were correctly categorized.Of the 19,994 DDoS incidents, only 1,158 were classified as non-malicious while 18,836 were correctly categorized.This misclasification rate is low especially when considering previous models making evident the capability of the BiLSTM model in distinguishing between benign and mallicous traffic.

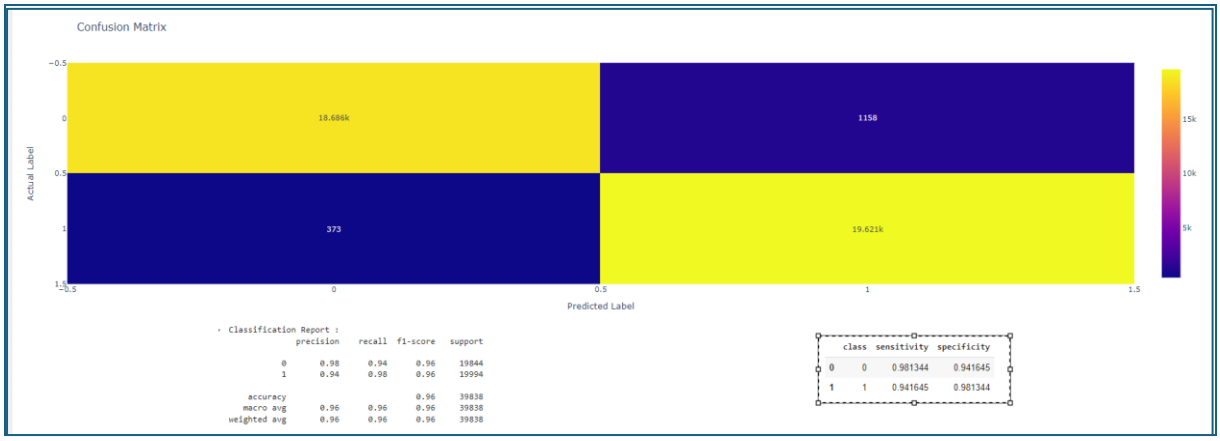

Fig16. BiLSTM Model Confusion Matrix, Classification report and Sensitivity and Specificity Report

6.3.4 Classification Report:

These metrics prove reasonable, indicating that the model performs well, having a high level of precision, recall, and F1-scores for each class. This balance is especially important for a

DDoS detection system, as it means that the possibility of both false positives and false negatives is kept to a minimum.

6.3.5 Sensitivity and Specificity:

If we compare the specificity and sensitivity measures given in the table above it is worth noticing that the proposed BiLSTM model has high accuracy in the classification of benign as well as DDoS traffic. The comparably high sensitivity and specificity of both classes also help the model to better perform in the real-world situations where it is often important that both classes are identified to the greatest level of accuracy.

6.4 Experiment 4: Stacked BiLSTM with Self Attention Mechanism

6.4.1 Accuracy

Specifically, the Stacked BiLSTM with Self-Attention model proved to have outstanding results reporting an accuracy of 98.39%. This high accuracy clearly shows that the current model of the classifier is extremely efficient in terminating the classification of benign and DDoS traffic which the previous models failed to achieve.

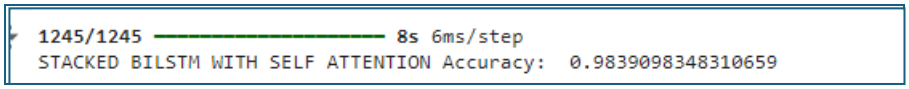

Fig17. Accuracy for Stacked BiLSTM with Self Attention Mechanism

6.4.2 Training and Validation Accuracy and Loss:

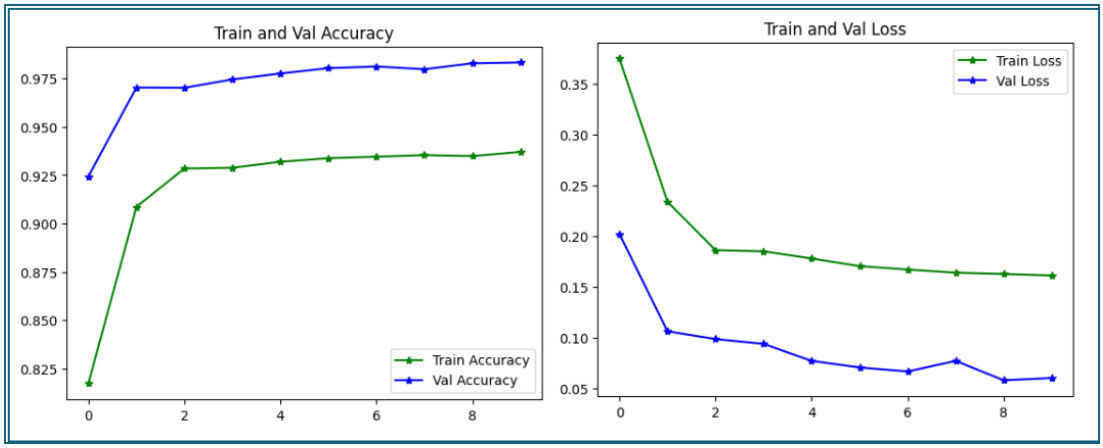

Fig18. Training and Validation Accuracy and Loss for Stacked BiLSTM with Self Attention Mechanism

Training vs. Validation Accuracy: The accuracy of the training sets and the validation sets rise drastically in the first epochs and then remain fluctuating around 97% – 98%. The low difference that exists between the training and validation accuracy shows that the model retains high accuracy on new unseen data and does not overfit.

Training vs. Validation Loss: The training and validation loss both show a decreasing progression, or in other words, the training and validation loss are rather similar to each other. This implies that we are optimizing for loss and ensuring a low gap between the training and the validation results to support the model's reliability.

6.4.3 Confusion Matrix:

Among the benign instances there were 19,239 and out of them 605 were classified as DDoS whereas 18,634 were accurately classified. From the total number of 19,994 DDoS cases, only 36 were labeled as benign but in reality they are not DDoS, while 19,958 were accurately categorized as DDoS.It can be clearly observed that the false positive rate as well as the false negative rate is extremely minimal, which actually emphasizes on the true positive and true negative rates achieved by the Stacked BiLSTM with Self-Attention model that is designed to classify the benign and malicious traffic.

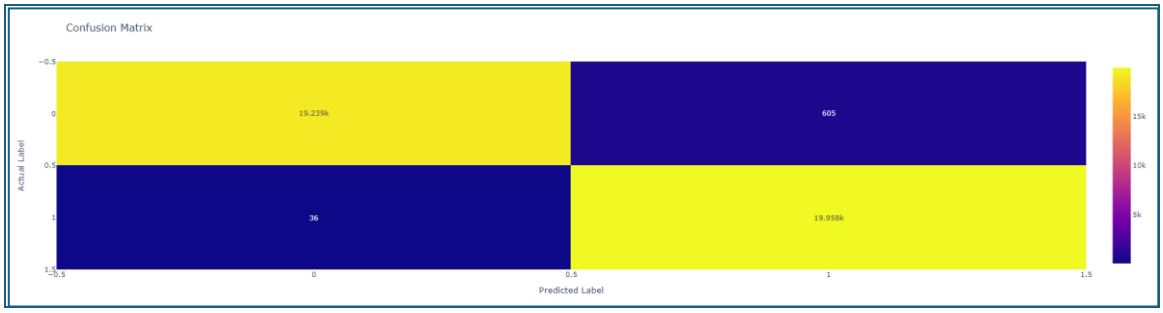

Fig19. Confusion Matrix for Stacked BiLSTM with Self Attention Mechanism

6.4.4 Classification Report:

The presented metrics prove that the model is virtually accurate in terms of precision, recall, and F1-scores for both classes, which guarantees the tool's high efficiency in the context of DDoS detection. It is important for both types of traffic to be correctly identified which is why the above metrics should be balanced.

Classification Report :

	precision	recall	f1-score	support
0	1.00	0.97	0.98	19844
1	0.97	1.00	0.98	19994
accuracy			0.98	39838
macro avg	0.98	0.98	0.98	39838
weighted avg	0.98	0.98	0.98	39838

	class	sensitivity	specificity
0	0	0.998199	0.969512
1	1	0.969512	0.998199

Fig20. Classification Report and Sensitivity and Specificity Report

6.5.5 Sensitivity and Specificity:

The sensitivity and specificity metrics denote the model's ability to detect both benign and DDoS traffic, and shows that the Stacked BiLSTM with Self-Attention model is highly sensitive and specific. The fact that the proposed values are very close indicates that the model is stable regardless of the type of traffic, which can provide a basis for employing it in real-world cybersecurity problems.

6.5 Discussion

The comparison of the four models—CNN, LSTM, BiLSTM, and the Stacked BiLSTM with Self-Attention—show that the models gradually improve with increases in complexity. This CNN model which was acceptable for spatial features did not fully capture temporal dependencies giving an accuracy of 58. 44% as well as relatively high false positive and false negative rates. The LSTM model which is capable of handling sequential data appeared slightly better with the training accuracy of 54. 79% of the time but this was coupled by huge

fluctuations and instability of the performance measures.
The BiLSTM model brought a great enhancement in terms of accuracy with a 96.16%. Data can be processed in different directions – forward and backward, due to which it managed to consider more complicated temporal relations resulting in the decrease in possible misclassifications. But the improvement that was apparent from the Stacked BiLSTM model was that of the Self-Attention mechanism. It was found to be 98.39% accurate, this model proved useful in paying more attention to some parts of the input sequence and came close to identifying the sequence's significant with an almost perfect accuracy , recall, and an F1-score. As a result DDoS detection proves the necessity to employ more complex mechanisms such as bidirectional Long Short Term memory and Self-attention layers as seen in the Stacked BiLSTM baseline. Maximizing all these measures is seen in the final model, which is therefore the most suitable for real-life applications.

7 Conclusion and Future Work

This research focused on improving the ability of detecting DDoS attacks using deep learning models especially with the method of employing Self-Attention Mechanism in to the Stacked BiLSTM model. The work contrasted the CNN, LSTM, BiLSTM, and the proposed architecture named Stacked BiLSTM with Self-Attention. It was noticed that although, the CNN and LSTM models offered a fairly good prediction accuracy of 58. 44% and 54. 77% , the BiLSTM model offered a much higher accuracy of 96.16%. However, the absolutely best result was achieved while using Stacked BiLSTM with Self-Attention method with the accuracy of 98.39%. This proves its better capacity of identifying DDoS attacks as it learns better focusing on parts of the input sequence that are the most important.

The future work should be on applying the proposed Stacked BiLSTM with Self-Attention for other types of threat, integrating the model in real-time environment, and providing better interpretability of the model. Moreover, the extension of the model would be valuable through the inclusion of contemporary and advancing datasets that would make the model relevant concerning current or emerging threats. Examining and evaluating the variations and identifying ways to apply the model to a large set of imbalanced data would also improve the system's performance and its applicability in various Cybersecurity sectors. This was in a bid to establish a stronger and far more flexible cyber threat detection mechanism good to adapt to present and future changes.

References

Ahsan, M., Nygard, K.E., Gomes, R., Chowdhury, M.M., Rifat, N. and Connolly, J.F., 2022. Cybersecurity threats and their mitigation approaches using Machine Learning—A Review. Journal of Cybersecurity and Privacy, 2(3), pp.527-555.

Ashraf, I., Narra, M., Umer, M., Majeed, R., Sadiq, S., Javaid, F. and Rasool, N., 2022. A deep learning-based smart framework for cyber-physical and satellite system security threats detection. Electronics, 11(4), p.667.

Aslan, Ö., Aktuğ, S.S., Ozkan-Okay, M., Yilmaz, A.A. and Akin, E., 2023. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), p.1333.

Bouchama, F. and Kamal, M., 2021. Enhancing cyber threat detection through machine learning-based behavioral modeling of network traffic patterns. International Journal of Business Intelligence and Big Data Analytics, 4(9), pp.1-9.

Dash, B., Ansari, M.F., Sharma, P. and Ali, A., 2022. Threats and opportunities with AI-based cyber security intrusion detection: a review. International Journal of Software Engineering & Applications (IJSEA), 13(5).

Mijwil, M., Salem, I.E. and Ismaeel, M.M., 2023. The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review. Iraqi Journal For Computer Science and Mathematics, 4(1), pp.87-101.

Okoli, U.I., Obi, O.C., Adewusi, A.O. and Abrahams, T.O., 2024. Machine learning in cybersecurity: A review of threat detection and defense mechanisms. World Journal of Advanced Research and Reviews, 21(1), pp.2286-2295.

Shaukat, K., Luo, S. and Varadharajan, V., 2022. A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks. Engineering Applications of Artificial Intelligence, 116, p.105461.

Aldarwbi, M.Y., Lashkari, A.H. and Ghorbani, A.A., 2022. The sound of intrusion: A novel network intrusion detection system. Computers and Electrical Engineering, 104, p.108455.

Yaseen, A., 2023. AI-driven threat detection and response: A paradigm shift in cybersecurity. International Journal of Information and Cybersecurity, 7(12), pp.25-43.

Ashiku, L. and Dagli, C., 2021. Network intrusion detection system using deep learning. Procedia Computer Science, 185, pp.239-247.

rewek-Ossowicka, A., Pietrołaj, M. and Rumiński, J., 2021. A survey of neural networks usage for intrusion detection systems. Journal of Ambient Intelligence and Humanized Computing, 12(1), pp.497-514.

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y. and Han, H., 2022. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security, 116, p.102675.

Mehmood, M., Javed, T., Nebhen, J., Abbas, S., Abid, R., Bojja, G.R. and Rizwan, M., 2022. A hybrid approach for network intrusion detection. CMC-Comput. Mater. Contin, 70(1), pp.91-107.