LSTM SMOTE: An Effective Strategies for DDoS Detection in Imbalanced Network Environments

Rissal Efendi *, Teguh Wahyono, Indrastanti Ratna Widiasari *

Posted Date: 24 July 2024

doi: 10.20944/preprints202407.1825.v1

Keywords: Imbalanced Network; DDoS; SMOTE; LSTM

Article

LSTM SMOTE: An Effective Strategies for DDoS Detection in Imbalanced Network Environments

Rissal Efendi [1,*], Teguh Wahyono [1] and Indrastanti Ratna Widiasari [2,*]

[1] Department of Applied Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Indonesia; teguh.wahyono@uksw.edu
[2] Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Indonesia
* Correspondence: rissal.efendi@uksw.edu (R.E.); indrastanti@uksw.edu (I.R.W.)

Abstract: In detecting DDoS, deep learning faces challenges and difficulties such as high computational demands, long training times, and complex model interpretation. This research focuses on overcoming these challenges by proposing an effective strategy for detecting DDoS attacks in unbalanced network environments. This research uses SMOTE to increase the class distribution of the data set by allowing models using LSTM to learn time anomalies effectively when DDoS attacks occur. The experiments carried out have shown significant improvement in the performance of the LSTM model when integrated with SMOTE. These include validation loss results of 0.048 for LSTM SMOTE and 0.1943 for LSTM without SMOTE, accuracy of 99.50 and 97.50. Apart from that, there was an increase in the f1 score from 93.4% to 98.3%. In this research, it is proven that SMOTE can be used as an effective strategy to improve model performance in detecting DDoS attacks on heterogeneous networks, as well as increasing model robustness and reliability.

Keywords: imbalanced network; DDoS; SMOTE; LSTM

1. Introduction

Network security is crucial in maintaining integrity and availability of service. One of the main threats in network infrastructure is DDoS (Distributed Denial of Service) attacks, where attacks can agitate access to online services by sending the target network massive packets with unnecessary traffic. DoS (Denial of Service) is an attack that is capable of minimizing bandwidth and computational resources of a particular system in the network, thereby overloading the system with data traffic, thereby preventing the system from delivering routine services to authenticated users. A DoS is a considered cyber-attack that allows attackers to attempt to cause systems and servers down and inaccessible, thus obstructing consumers from accessing resources and servers [1]. DDoS can even damage a system further on a wider scale. Distributed Denial of Service (DDoS) attack is a kind of cyber attack executed by using a large number of geographically distributed computers or devices to simultaneously access a target computer resource, such as a website or network, with the aim of making it unavailable to authorized users [2]. DDoS continues to threaten and undermine network security in all fields of business regardless of their scale due to their increasing complexity, volume and frequency[3]

DDoS data can be used to identify DDoS attacks leveraging computational algorithms, including machine learning (ML) and Deep Learning (DL). The main objective of the study, however, focused on the earlier identification of DDoS attack impacts. Usually, the impacts are constrained by inadequate selection of predictor variable which employs to classify DDoS attacks and typical classifiers that produced subpar results to examine the correlation among the detector attributes in the DDoS data[4,5]. DL is an emerging field of computer science that employs an advanced set of features embedding methods to automated learning from past data and predict outcomes accurately[6]. It has been successfully employed in diverse deployment over the years, including financial market forecasting[7], student performance evaluation[8] , forecasting modeling[9], text

 doi:10.20944/preprints202407.1825.v1

classification [10], and some instances. Data analysts are motivated to develop effective strategies that helpful in aiding system administrators to detect DDoS effectively [11]. Therefore, for reliable DDoS attack information, it is vital to examine and use models on DDoS data.

Currently, DDoS detection generally depends on DL methods and algorithms to distinguish normal traffic and attacks. However, detection success is frequently constrained by several challenges, including Class Imbalance [12]. In many problems[13–16], the target group is the class of interest, for example the positive class. A widely recognized example of class imbalanced ML context is the packet diagnostic task of DDoS detection where the most of the packet are normal and detecting DDoS is of higher interest. For the instance, in some studies, researchers consider the predominant category group of DDoS attack as the negative class. These imbalanced data sets might be highly demanding, particularly within the context of with big data analytics[17,18], and unconventional ML approaches are frequently needed to achieve favorable outcomes. A comprehensive grasp of the class imbalance issue and the existing methods to handle it, as such the prevalence of skewed data exists in numerous real-world applications. Methods for handling class imbalance in ML can be categorized into three groups: algorithm-level methods, data-level techniques, and hybrid approaches. To mitigate the level of imbalance, data-level techniques is implemented by employing various data sampling methods.

In handling class imbalance, Algorithm-level methods is often implemented with a weight or cost schema, involve adjusting learner or its output to mitigate partiality towards the group of majorities. Hybrid systems strategically integrate both algorithm methods and the sampling [19]. DL methods have gain popularity as they have advanced ideas in computer vision, speech recognition and other fields. Their recent achievement can be credited to an enhanced data availability, enhancements in resources with algorithmic innovation that expedite training and enhance generalization to new data. In fact, many researchers concur that the DL topics with class imbalanced data is underexplored [20,21]. Therefore, there is a requirement for an oversampling method that is specifically designed for DL models, can work on raw data, preserving their inherent properties, and generate high-quality artificial data that can enhance minority classes and achieve balance in the training set[22].

The emergence of DL has transformed DDOS attack detection, LSTM, a specific recurrent neural network design initiated by Hochreiter and Schmidhuber [23], has attained significant traction for its capacity to account for time-based patterns in sequential data. In DDOS attack environment, where time-series data are pervasive, LSTM models have shown has demonstrated strong potential regarding the detection of abnormal behavior and potential cyber security threats. several researchers have investigated LSTM-based intrusion detection systems[24–26]. These studies emphasize the effectiveness of LSTM models in distinguishing and labeling diverse attacks accurately in computer network environments.

In some approaches that use machine learning, such as algorithms utilizing deep learning, have demonstrated promise in enhancing the ability of DDoS detection. One commonly researched method is the use of LSTM networks, which is a type of neural network architecture that is capable of handling temporal data well. However, using LSTMs in the context of DDoS detection is not always practical or efficient for all network environments. LSTMs often require significant computing resources and can require long training times. Additionally, analyzing outcomes from LSTM models can also be complex, especially in contexts that require a profound insight of the model decisions-making process. The main objective of this study is to develop and examine the effective strategies to detect DDoS attack in an imbalance network environment. This study focusses on utilizing SMOTE (Synthetic Minority Over-sampling Technique) by addressing data imbalance problem and enhancing the efficiency and accuracy of DDoS attack detection. SMOTE is used to balance the class distribution with dataset and to allow LSTM models to learn more effectively and deliver more accurate results in detecting attacks. By leveraging LSTM-based deep learning, the model shows exceptional capability in discovering subtle and time-sensitive anomalies, facilitating the early detection of advanced and persistent cyberattacks [27,28].

 doi:10.20944/preprints202407.1825.v1

2. Materials and Methods

This section describes the proposed model and algorithm employed in this study such as how data was collected, data was processed, handling imbalanced data by using SMOTE, splitting dataset into data training and testing, developing the model using LSTM, model evaluation and comparing LSTM and LSTM with SMOTE results. Figure 1 represents the research methods conducted to achieve this goal:

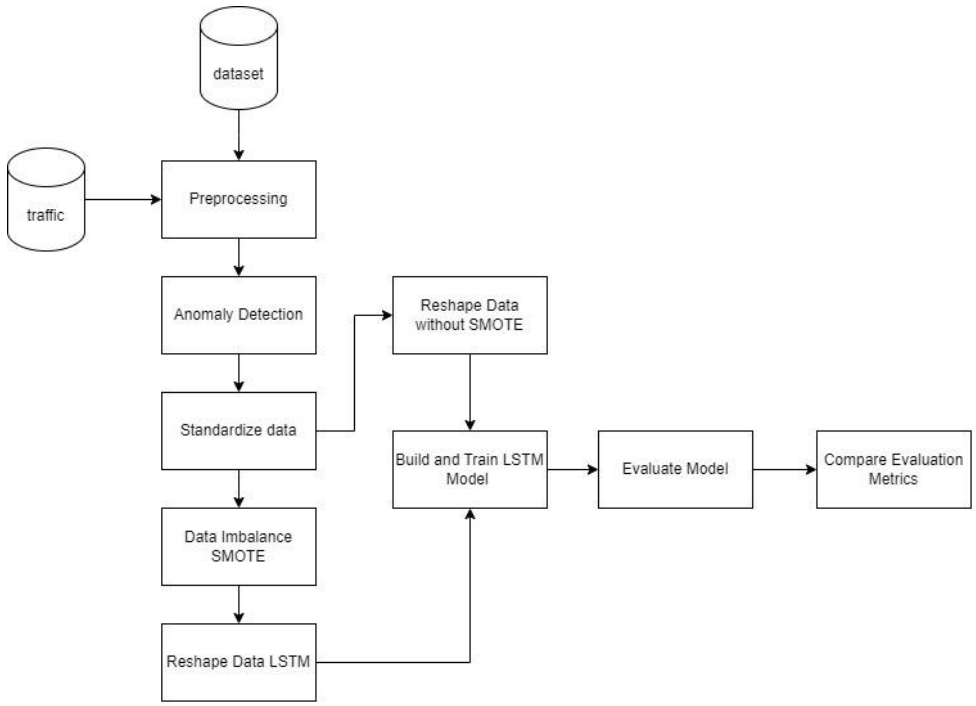

Figure 1. Flowchart of research.

2.1. Data Collection

In this research, the first step that we conducted was collecting relevant data. The data collected included attributes such as time, source, destination, protocol, length, clusters, and anomalies. This data includes information about normal and suspicious network traffic. This data collection is important to provide a basis for model training and evaluation. Data was taken from sources that have network traffic records, which reflected the real conditions of the observed network. When data was collected, Host with IP address of 172.16.0.6 – 172.16.0.8 became the DDoS attacker while the host with the IP address of 192.168.50.12 was the target of attack. This host with IP address received massive and abnormal traffic from IP address 172.16.0.5 – 172.16.0.8. The data used was 1,048,575 records collected for DDoS attack detection analysis. This data includes time, source, destination, protocol, and length of traffic. The data then was processed using SMOTE and analyzed using LSTM algorithm. The simplified network topology is represented in Figure 2.

 doi:10.20944/preprints202407.1825.v1

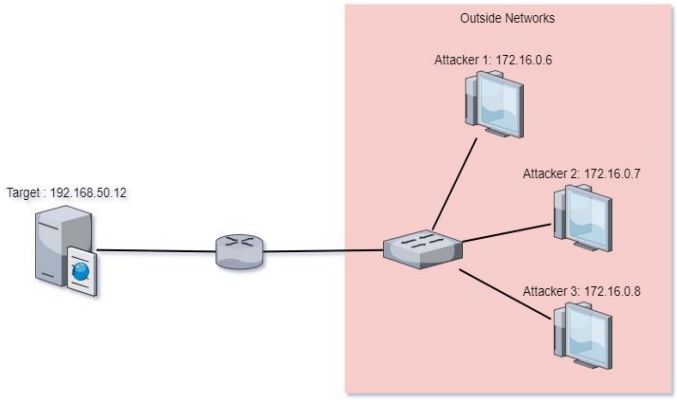

Figure 2. Simplified Network Topology.

2.2. Data Pre-Processing

After data was collected, we performed data pre-processing by cleaning and preparing data before being used in model training. The steps in data pre-processing encompassed removing or imputing missing values and correcting errors in the data, reshaping data into a appropriate format for further analysis, converting the time format into a model compatible form and transforming categorical attributes into a numeric form that can be processed by the model.

2.3. Standardization of Features

Feature standardization is the next step after data pre-processing. Standardization is used to ensure that all features are at the same scale. This is important because features with different scales can negatively affect model performance. In this study, we used StandardScaler to standardize the data to have a distribution with a mean of 0 and a standard deviation of 1. Feature standardization helps the model to learn patterns from the data more effectively.

2.4. Handling Data Imbalance

Classification of data becomes challenging due to the vast due to extensive scale and imbalance characteristic of data. Class imbalance problem grows into major concern in data mining. Imbalance problem arises in which one of the two classes having significantly more prevalent sample compared to other classes. Most algorithms tend to prioritize classifying the majority sample while neglecting or misclassifying minority sample. The minority samples are uncommon yet significant. An oversampling method, SMOTE generates synthetic observations from existing samples of the minority class. It does not only replicate the existing data; it also generates new data points that are closely resemble the minority class using data augmentation to enhance minority classes. These new synthetic training instances are randomly generated by selecting one or more K-nearest neighbors for each of the minority classes. After finishing oversampling, the issue of an imbalanced dataset is resolved and different classification models was ready tobe tested. In this research, SMOTE was employed to increase the number of samples from minority classes so that the model can learn better from the data. In network traffic datasets, the amount of data indicating DDoS attacks was usually much less than normal traffic data. In our dataset, each sample classified as a DDoS attack (minority class) was selected for oversampling. In case a sample exhibits the attributes: Time=0.001132, Source=192.168.50.1, Destination=172.16.0.5, Protocol=HTTP, Length=1139, Cluster=-1, Anomaly=True. For every minority sample, we searched k, the nearest neighbors in attribute space using the k-Nearest Neighbors (k-NN) algorithm. For example, for a sample with the above attributes, we found several nearest neighbors that are also DDoS attack samples in the dataset. After the nearest neighbors was found, we randomly chose one of them. For example, the selected nearest neighbor had the attributes: Time=0.000774, Source=192.168.50.1, Destination=172.16.0.5,

 doi:10.20944/preprints202407.1825.v1

Protocol=TCP, Length=66, Cluster=-1, Anomaly=True. A synthetic sample is then generated using the formula:

$$x_{new} = x_i + (x_{neighbour} - x_i) * \delta \,, \quad (1)$$

where,

x_i: the original sample (for example, Time=0.001132, Source=192.168.50.1, Destination=172.16.0.5, Protocol=HTTP, Length=1139, Cluster=-1, Anomaly=True),
$x_{neighbor}$: the selected nearest neighbor (for example, Time=0.000774, Source=192.168.50.1, Destination=172.16.0.5, Protocol=TCP, Length=66, Cluster=-1, Anomaly=True) and
δ: a random number between 0 and 1.

For example: if δ= 0.5, synthetic sample x_{new} is calculated as follows: $Time_{new}$=0.001132+(0.000774−0.001132) ×0.5=0.000953.

Other attributes such as Source, Destination, Protocol, Length, Cluster, and Anomaly were also calculated in the same way. The result is a new synthetic sample that might look like: Time=0.000953, Source=192.168.50.1, Destination=172.16.0.5, Protocol=Mix, Length=602.5, Cluster=-1, Anomaly=True.

2.5. Training and Testing Data

After the data was normalized, then the data was split into a training and a test dataset. This data split was done to perform that the model could be properly evaluated on data that has never been seen before. Data was divided by a certain ratio, a training split of 80% and a testing split of 20%. The data was split randomly to ensure that the data in the training set and test set were representative.

2.6. LSTM Model Development

In this phase, we utilized LSTM to construct the model. LSTM is a kind of artificial neural network which is suitable for time series data processing and complex patterns detection in the data. In the case of DDoS attack detection, network data is often sequential and has temporal dependencies. LSTM is very effective in handling this type of data because of its ability to remember information over long and short periods of time. Here is an explanation of how each LSTM formula is used in this context:

1. Forget gate

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f, \quad (2)$$

Forget gate determines which new information from the previous cell state (C_{t-1}) that must be forgotten. In the case of DDoS, this can mean forgetting irrelevant information from previous network traffic that is unrelated to the attack. When LSTM receives network data on time *t* forget gate will use the previous hidden state (h_{t-1}) and current input (x_t) to decide how much information from the past to remember or forget.

2. Input gate

$$i_t = \sigma(W_f . [h_{t-1}, x_t] + b_f, \quad (3)$$

$$\tilde{C}_t = tanh(W_c . [h_{t-1}, x_t] + b_{fc}, \quad (4)$$

The input gate decides what new information will be stored in the current cell state C_t In the case of DDoS, this means adding important information about new network packets that may indicate attack patterns. LSTM will retrieve the previous hidden state (h_{t-1}) and current input (x_t) to determine what new information needs to be added to the current cell state (C_t).

3. Memory Cell Status

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

 doi:10.20944/preprints202407.1825.v1

The state of memory cells is updated by combining old information that is still relevant and new information that is important. In the context of DDoS, the state of memory cells (C_t) stores information about the network traffic that the model has seen up to the current point in time. By combining the output of the forget gate (f_t) an gate input (i_t) LSTM updates the cell state (C_t) to represent current and relevant information about network traffic

4. Output gate

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o \tag{6}$$

$$h_t = o_t * tanh(C_t), \tag{7}$$

The output gate decides which part of the memory cell state will be output as output (h_t) In the case of DDoS, this means deciding which information to use to determine whether the current network packet is part of an attack. The gate output will retrieve the current cell state (C_t) and determine which parts are relevant to output as output (h_t) which will be used in subsequent steps to predict whether a DDoS attack occurred.

2.7. Model Training with Early Stopping

Once the LSTM model is built, the next step is to train the model using the training data. Model training is carried out using the early stopping technique to avoid overfitting. Early stopping works by observing the model's performance on the validation set. In addition, it also work by stopping training if the performance begins to decline. This ensures that the model does not overfit on the training data and can generalize well on new data. During training, model parameters are updated using an Adam-like optimization algorithm.

2.8. Model Evaluation

In model evaluation, we used confusion matrix to define the model requirement. Some components of the confusion matrix are False positive (FP), false negative (FN), true positive (TP), and true negative (TN). The confusion matrix is provided to exhibit the effectiveness of our model's classification. The confusion matrix emphasizes whether predictions were valid. In addition, we evaluated our proposed model employing widely used metrics in DDoS. The mathematical formulas of precision, recall, and f-score are represented as follows:

Accuracy reflects the model's precise predictive performance. Accuracy is a metric that calculates the overall percentage of detected and abnormal results which is generated by the LSTM model. It shows the cumulative success ratio of any DDS and is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative

Precision or the false negative rate (FNR), commonly known as precision, is the proportion of misclassified attacks to the total number of attack occurrences. The precision was derived from Equation (9) shows how many positive DDoS detection are predicted exactly:

$$Precision\ (P) = \frac{TP}{FP + TP} \tag{9}$$

P=Precision, TP=True Positive, FP=False Positive

Recall or the detection rate (DR), commonly referred to as the true positive rate (TPR) which is the success ratio of identifying adverse occurrences relative to the overall number of adverse vectors. Equation (10), which assesses recall, exposes how many true positives are successfully detected:

$$Recall\ (r) = \frac{TP}{FN + TP} \tag{10}$$

r =recall,TP=True Positive,FN=False Negative

 doi:10.20944/preprints202407.1825.v1

F-score or F1 score is important because it offers the next information about the network performance. It considers both false positives and negatives. The F1 score is beneficial especially in cases where the class labels distribution is unbalanced. The F-score was computed using Equation (11), demonstrates the consistency of recall and precision:

$$F_{score} = 2x\frac{P \times R}{P + R} \quad (11)$$

$$F_{score} = 2x\frac{P \times R}{P + R}$$

R= Recall, P=Precision

2.9. Comparison LSTM Model with and without SMOTE

This research also compares two approaches: using SMOTE to handle data imbalance and not using SMOTE. In the first approach, SMOTE is used to increase the minority class sample so that the data becomes more balanced. In the second approach, data is used without oversampling. The results of both approaches are compared to see how handling data imbalance affects the performance in detecting DDoS attacks.

3. Results

In this research, we used the LSTM (Long Short-Term Memory) model to detect DDoS attacks based on network data consisting of various features such as time, source, destination, protocol, packet length, and cluster. We implemented the LSTM model and evaluated the model performance with various metrics. Additionally, we extend the analysis by applying the SMOTE technique to address class imbalance in the dataset, thereby clarifying the research focus on improving DDoS attack detection.

3.1. DDoS Detection LSTM Model without SMOTE

This research also compares two approaches: using SMOTE to handle data imbalance and not using SMOTE. In the first approach, SMOTE is used to increase the minority class sample so that the data becomes more balanced. In the second approach, data is used without oversampling. The results of both approaches are compared to see how handling data imbalance affects the performance

Figure 3 shows the evaluation results of the DDoS prediction namely validation loss and training loss. Training Loss of 0.0340 shows the average value of the loss function on the training data. This value specifies how effective the model learned the training data. The smaller the training loss value, the better the model is at optimizing the training data. In this case, the training loss of 0.155 shows that the model is very good at learning patterns in the training data. Validation Loss of 0.193 refers to the average loss value calculated on validation data, which the model hasn't been trained on. It serves to evaluate how well the model generalizes from its training data to new, unseen data. A lower validation loss suggests that the model can effectively apply learned patterns to new data, indicating good generalization. This metric is crucial in assessing the model's performance beyond training, helping to verify its reliability in real-world applications where unseen data may differ from the training set.

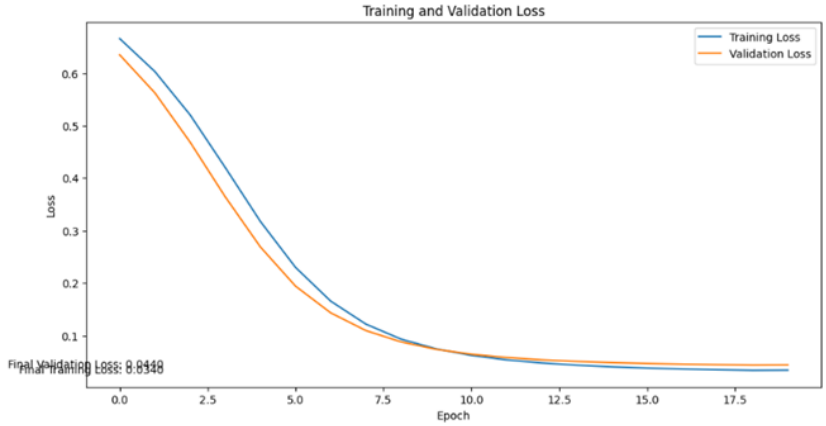

Figure 3. Training and Validation Loss Comparisson on LSTM Model without SMOTE.

 doi:10.20944/preprints202407.1825.v1

Figure 4 shows the evaluation results of the DDoS detction namely model accuracy and model loss. Final Training Accuracy (0.9420 or 94.20%) represents the percentage of accurate predictions generated by the model on the training dataset. An accuracy of 99.50% signifies the model's near-perfect performance in predicting training data, indicating minimal errors during training. Final Validation Accuracy (0.9750 or 97.50%) indicates the percentage of correct predictions on validation data. This metric evaluates the model ability to generalizes to unseen data. A high validation accuracy suggests that the model effectively applies learned patterns to new data, demonstrating its robustness beyond the training set and validating its performance in scenarios.

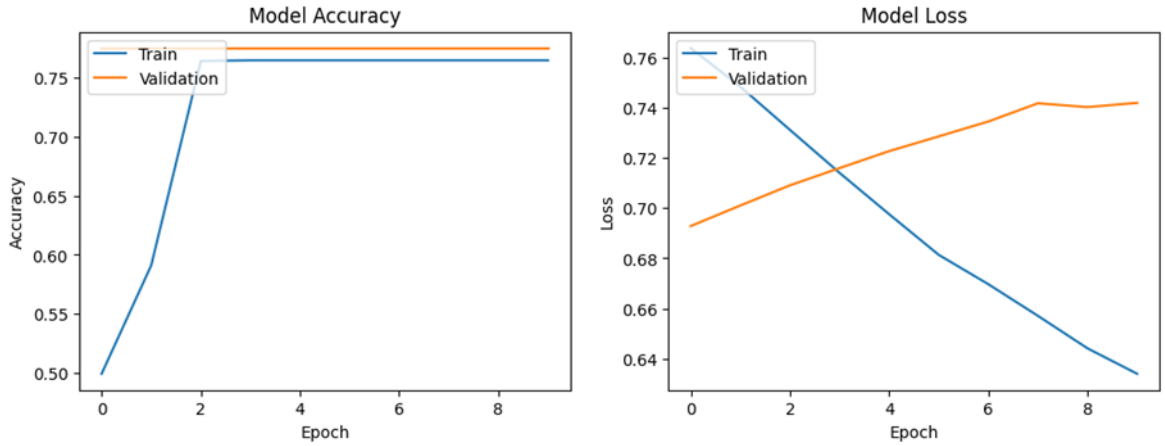

Figure 4. Model Accuracy and Loss on LSTM Model without SMOTE.

The accuracy of 99.20% shows that the model is also very accurate in predicting data that was not seen during training, indicating the strong generalization ability of the model. The small difference between training loss and validation loss indicates the model is not overfitting. Overfitting happens when the model excels at learning training data but struggles to generalize to new data, typically seen with low training loss but high validation loss. In this case, similar training loss and validation loss indicate that the model has good generalization. High accuracy on both training data (99.50%) and validation data (99.20%) indicates the model effectively predicts DDoS attacks and normal conditions. It distinguishes well between DDoS and non-DDoS attacks, demonstrating strong performance in learning and applying patterns. These metrics highlight the model's reliability and effectiveness in detecting DDoS attacks, supported by low loss and consistent high accuracy across datasets.

Table 1 presents critical evaluation metrics for two classes in a classification model: "0 (No DDoS)" and "1 (DDoS)". It includes Precision, Recall, and F1-score, which are used to verify the model's performance in predicting each class accurately. Precision measures the proportion of positive predictions that are truly positive. For class "0 (No DDoS)", a Precision of 97.6% means that of all predictions classified as "No DDoS", 97.6% of them were actually "No DDoS". For class "1 (DDoS)", a Precision of 89.4% means that of all predictions classified as "DDoS", 89.4% of them were actually "DDoS".

Table 1. Classification Report on LSTM Model without SMOTE.

Class	Accuracy	Precision	Recall	f1-score
0 (No DDOS)	96.33%	97.6%	96.7%	90.6%
1 (DDOS)	95.42%	89.4%	93.6%	93.4%

Recall determines the proportion of positive data that is actually detected by the model. For class "0 (No DDoS)", Recall of 96.7% means that of all data that is truly "No DDoS", 96.7% was successfully detected by the model as "No DDoS". For class "1 (DDoS)", Recall of 93.6% means that of all the data that was actually "DDoS", 93.6% was successfully detected by the model as "DDoS".

 doi:10.20944/preprints202407.1825.v1

F1-score is a harmonization of Precision and Recall, providing a single value that describes the balance between the two metrics. F1-score is important when the class distribution is unbalanced. For class "0 (No DDoS)", the F1-score of 90.6% shows a good balance between Precision and Recall. For class "1 (DDoS)", the F1-score of 93.4% also shows a better balance between Precision and Recall.

Overall, the model shows excellent performance in detecting both “No DDoS” and “DDoS”. The class "0 (No DDoS)" has a very high Precision, indicating that the model very rarely gives false positive predictions for this class. Recall is also high, although slightly lower than Precision, meaning some "No DDoS" instances may be incorrectly detected as "DDoS". For class "1 (DDoS)", the slightly lower Precision indicates some false positives, but the high Recall indicates the model is very effective at detecting DDoS attacks when they occur. The good balance between Precision and Recall in both classes is reflected in the F1-score which is also high, indicating that this model is reliable in classifying both classes effectively.

3.2. DDoS Detection LSTM Model with SMOTE

In this scenario, the model is trained and tested using the SMOTE technique to handle class imbalance. The following are the results of the model evaluation:

Figure 5 shows model analysis for detecting DDoS attacks, the evaluation of Training Loss of 0.0253 and Validation Loss of 0.0428 provides an important picture of the performance and reliability of the model. A low Training Loss value such as 0.0253 indicates that the model is very efficient in learning complex patterns that may exist in the training data related to DDoS attacks. This means that the model accurately reduces the prediction error on the training data, which directly reflects the adaptability and learning to features that differentiate between DDoS attacks and normal network traffic. Validation Loss values that are slightly higher than Training Loss, such as 0.0428, indicate that the model has a good ability to avoid overfitting. This means the model is not only able to remember and predict well the data used in training, but can also effectively apply this knowledge to new data with which it was previously unfamiliar. Training Loss of 0.0253 and Validation Loss of 0.0428, the model shows excellent performance in detecting DDoS attacks. The ability to reduce loss on these two datasets shows that the model has a high level of accuracy and reliability in identifying DDoS attack threats, which is crucial for effective and responsive network security.

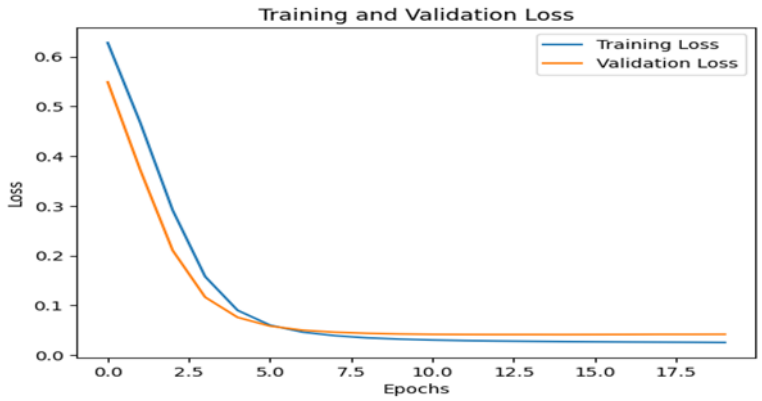

Figure 5. Training and Validation Loss on LSTM Model with SMOTE.

Figure 6 shows the model was trained and validated with data showing excellent performance. During training, the model loss decreased significantly from 0.6167 at the beginning to 0.0248 at the end of training, indicating that the model was improved at optimizing the training data. Validation loss also decreased consistently from 0.5319 initially to 0.0434, indicating that the model maintained its performance on never-before-seen data. The accuracy of the model during training is very high, starting from 87.31% and immediately reaching stability at 99.50% after the first epoch, which remains consistent until the end of training. The same applies to validation accuracy, which reaches 99.20% after the first epoch and remains stable throughout training. The steady decrease in loss and high accuracy indicate that the model learns significantly without significant issue on overfitting or

 doi:10.20944/preprints202407.1825.v1

underfitting. With excellent performance on validation data, this model can be relied on to detect DDoS attack patterns with high accuracy and low loss, making it an effective tool in detecting DDoS attacks on the network.

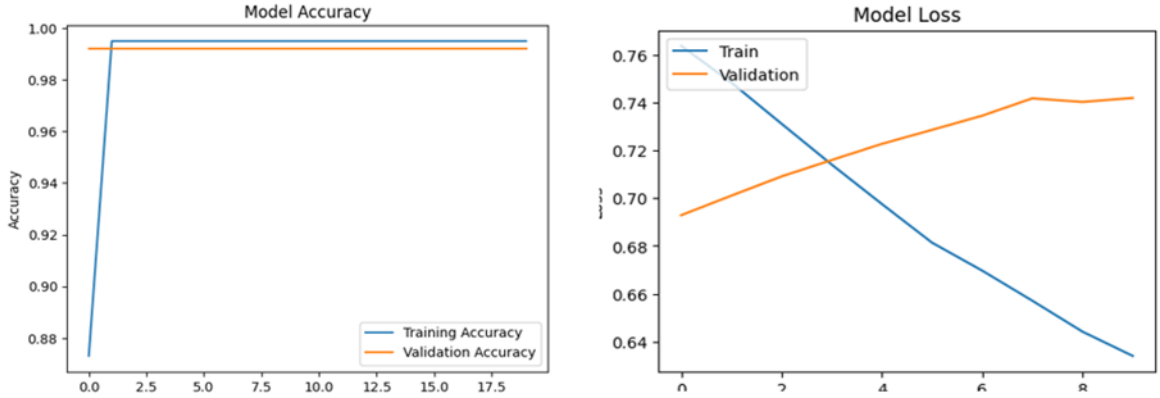

Figure 6. Model Accuracy and Model Loss on LSTM Model with SMOTE.

The results of DDOS attack detection by using SMOTEE was shown in Figure 2. Two classes were formed in this model: class 0 (No DDOS) and class 1 (DDOS). For class 0 (no DDOS), it achieves a precision of 98.5%. This means that out of all the instances predicted as "No DDoS" by the model, 98.5% were correctly identified. The recall for DDOS detection reaches 97.3%, which indicating that the model successfully identified 97.3% of all actual "No DDoS" instances. The f1-score, which is the mean of precision and recall, is 93.1%. This score reflects a good balance between precision and recall for class 0. For class 1 (DDoS), the model achieves a precision of 93.6%, indicating that 93.6% of the instances predicted as "DDoS" were correct. The recall for this class is 96.2%, meaning the model correctly recognized 96.2% of all actual "DDoS" instances. The f1-score for class 1 is 98.3%, which showing excellent performance in detecting DDoS attacks with a strong balance between precision and recall. The LSTM model which is supported by the SMOTE technique provides excellent performance in detecting DDoS attacks. The Value of high precision, recall, and f1-score in both classes demonstrate that the model is both accurate and reliable. this means that LSTM with SMOTE is able to address class imbalance. Overall, the model's performance metrics indicate a strong ability to detect and distinguish between DDoS and non-DDoS traffic effectively.

The LSTM model with SMOTE as represented on figure 3 has significant improvements in performance with lower validation loss and higher accuracy when compared to the LSTM model without SMOTE. This shows that the use of SMOTE helps in improving the model ability to detect DDoS attacks more accurately.

Comparison of classification results between using SMOTE and without SMOTE on DDoS Detection LSTM model shows a significant difference in performance in detecting DDoS attacks. In table 1 without using SMOTE, class 0 (No DDoS) has a precision of 97.6%, recall of 96.7%, and f1-score of 90.6%. For class 1 (o), precision was 89.4%, recall 93.6%, and f1-score 93.4%On the other hand, table 2 which uses SMOTE shows that class 0 has a precision of 98.5%, recall of 97.3%, and f1-score of 93.1%. For class 1, precision reached 93.6%, recall 96.2%, and f1-score 98.3%. These results indicate that the use of SMOTE improves the model's performance in detecting the minority class, namely the DDoS class (class 1). Class 1 precision and recall improved from 89.4% and 93.6% without SMOTE to 93.6% and 96.2% with SMOTE, respectively. Class 1 F1-score also increased significantly from 93.4% to 98.3%. However, this increase was accompanied by a slight decrease in class 0 (No DDoS) f1-score, from 90.6% without SMOTE to 93.1% with SMOTE. However, class 0 precision and recall continued to increase. Overall, the use of SMOTE helps the model be more balanced in detecting both classes, especially improving the model ability to more accurately detect DDoS (class 1) attacks. This is important in the context of network security, where proper detection of DDoS attacks is crucial. While

 doi:10.20944/preprints202407.1825.v1

there is a slight trade-off in class 0 performance, the significant improvement in class 1 detection makes using SMOTE very worthwhile

Table 2. Classification Report on LSTM with SMOTE.

Class	Accuracy	Precision	Recall	f1-score
0 (No DDOS)	96.71%	98.5%	97.3%	93.1%
1 (DDOS)	96.12%	93.6%	96.2%	98.3%

Table 3. Validation and Training Results Comparison.

Model	Validation Loss	Training Loss	Validation Accuracy	Training Accuracy
LSTM	0.1934	0.1548	97.50%	94.20%
LSTM with SMOTE	0.0428	0.0253	99.50%	99.20%

4. Discussion

In this study, we evaluated the performance of LSTM models for DDoS detection, comparing a standard LSTM model with an improved model using SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance in the dataset. Our findings show significant improvements in both accuracy metrics and performance measures when SMOTE is implemented. Our results are in line with previous research showing that class imbalance can affect the performance of machine learning models in cyber security applications, especially in DDoS detection[29–32]. By implementing SMOTE, we succeeded in overcoming this problem, as evidenced by a decrease in validation loss from 0.1934 to 0.0428 and an increase in validation accuracy from 97.50% to 99.50%. This improvement confirms the effectiveness of SMOTE in increasing the model's robustness against DDoS attacks. The implications of our findings go beyond the immediate scope of this study. Achieving higher recall and precision in detecting DDoS attacks (98.3% f1-score) with SMOTE's enhanced LSTM model highlights its potential in improving network security measures. This approach not only strengthens defense mechanisms against evolving cyber threats, but also emphasizes the important role of data preprocessing techniques in optimizing model performance. Future research directions should explore additional data augmentation methods to further sharpen the generalization ability of the model. Further research could investigate the application of ensemble learning techniques or integration of real-time network traffic data to improve the adaptability of our approach in dynamic network environments. Additionally, expanding this research to cover a wider range of attack scenarios and more diverse datasets would provide deeper insights into the scalability and robustness of our proposed methodology.

5. Conclusions

Based on the experimental results, SMOTE significantly improves the model capabilty in detecting DDoS attacks. Without implementing SMOTE, the LSTM model reaches higher validation loss and lower accuracy compared to models using SMOTE. Specifically, the model without SMOTE obtains a validation loss of 0.1934 and validation accuracy of 97.50%, while with SMOTE, validation loss drops to 0.0428 and validation accuracy increases to 99.50%. In the classification table, using SMOTE increases precision, recall, and f1-score for both classes (No DDoS and DDoS). The DDoS class, which is a minority class, shows significant improvement, with f1-score increasing from 93.4% without SMOTE to 98.3% with SMOTE.

In general, this research proves that by involving SMOTE in the model, it makes an effective technique to overcome data imbalance in DDoS attack detection. Using SMOTE not only improves model accuracy, but also ensures that the model is more reliable in detecting DDoS attacks. Thus, SMOTE proves to be an effective strategy to improve model performance in unbalanced network environments

Author Contributions:

Funding:

Acknowledgments: We would like to express our gratitude to the Directorate of Infrastructure and Digitalization at Satya Wacana Christian University for making it possible for us to obtain all the crucial information required for our network penetration testing activities, which in turn made this study possible. We were provided with the necessary tools and support to finish our research on "Effective Strategies for DDoS Detection in Imbalanced Network Environments." Their dedication to furthering cybersecurity research is much appreciated. Without their kind donation and cooperation, this study would not have been possible without their kind donation and collaboration.

Conflicts of Interest:

References

1. S. Sambangi and L. Gondi, "A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression," in *The 14th International Conference on Interdisciplinarity in Engineering—INTER-ENG 2020*, Basel Switzerland: MDPI, Dec. 2020, p. 51. doi: 10.3390/proceedings2020063051.
2. C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, "Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model," *Applied Sciences*, vol. 11, no. 11, p. 5213, Jun. 2021, doi: 10.3390/app11115213.
3. J. Cheng, Y. Liu, X. Tang, V. S. Sheng, M. Li, and J. Li, "DDoS Attack Detection via Multi-scale Convolutional Neural Network," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 1317–1333, 2020, doi: 10.32604/cmc.2020.06177.
4. J. Cheng, Y. Liu, X. Tang, V. S. Sheng, M. Li, and J. Li, "DDoS Attack Detection via Multi-scale Convolutional Neural Network," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 1317–1333, 2020, doi: 10.32604/cmc.2020.06177.
5. A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Syst Appl*, vol. 169, p. 114520, May 2021, doi: 10.1016/j.eswa.2020.114520.
6. A. Khattak, M. Z. Asghar, M. Ali, and U. Batool, "An efficient deep learning technique for facial emotion recognition," *Multimed Tools Appl*, vol. 81, no. 2, pp. 1649–1683, Jan. 2022, doi: 10.1007/s11042-021-11298-w.
7. A. Khattak *et al.*, "An Efficient Supervised Machine Learning Technique for Forecasting Stock Market Trends," 2022, pp. 143–162. doi: 10.1007/978-3-030-75123-4_7.
8. M. Zubair Asghar *et al.*, "Performance Evaluation Of Supervised Machine Learning Techniques For Efficient Detection Of Emotions From Online Content," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1093–1118, 2020, doi: 10.32604/cmc.2020.07709.
9. A. Khan, A. M. Khattak, M. Z. Asghar, M. Naeem, and A. U. Din, "Playing First-Person Perspective Games with Deep Reinforcement Learning Using the State-of-the-Art Game-AI Research Platforms," 2021, pp. 635–667. doi: 10.1007/978-3-030-77939-9_18.
10. S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and S. Khan, "Classification of Poetry Text Into the Emotional States Using Deep Learning Technique," *IEEE Access*, vol. 8, pp. 73865–73878, 2020, doi: 10.1109/ACCESS.2020.2987842.
11. A. Alsaeedi, O. Bamasag, and A. Munshi, "Real-Time DDoS flood Attack Monitoring and Detection (RT-AMD) Model for Cloud Computing," in *The 4th International Conference on Future Networks and Distributed Systems (ICFNDS)*, New York, NY, USA: ACM, Nov. 2020, pp. 1–5. doi: 10.1145/3440749.3442606.
12. J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, p. 27, Dec. 2019, doi: 10.1186/s40537-019-0192-5.
13. R. B. Rao, S. Krishnan, and R. S. Niculescu, "Data mining for improved cardiac care," *ACM SIGKDD Explorations Newsletter*, vol. 8, no. 1, pp. 3–10, Jun. 2006, doi: 10.1145/1147234.1147236.
14. W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, Jul. 2013, doi: 10.1007/s11280-012-0178-0.
15. M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big Data fraud detection using multiple medicare data sources," *J Big Data*, vol. 5, no. 1, p. 29, Dec. 2018, doi: 10.1186/s40537-018-0138-3.
16. M. Kubat, R. C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," *Mach Learn*, vol. 30, no. 2/3, pp. 195–215, 1998, doi: 10.1023/A:1007452223027.

17. R. A. Bauder and T. M. Khoshgoftaar, "The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data," *Health Inf Sci Syst*, vol. 6, no. 1, p. 9, Dec. 2018, doi: 10.1007/s13755-018-0051-3.
18. R. A. Bauder, T. M. Khoshgoftaar, and T. Hasanin, "An Empirical Study on Class Rarity in Big Data," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Dec. 2018, pp. 785–790. doi: 10.1109/ICMLA.2018.00125.
19. B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Nov. 2016, doi: 10.1007/s13748-016-0094-0.
20. S. Pouyanfar *et al.*, "Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, Apr. 2018, pp. 112–117. doi: 10.1109/MIPR.2018.00027.
21. M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/j.neunet.2018.07.011.
22. D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023, doi: 10.1109/TNNLS.2021.3136503.
23. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
24. A. Dahou *et al.*, "Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm," *Comput Intell Neurosci*, vol. 2022, pp. 1–15, Jun. 2022, doi: 10.1155/2022/6473507.
25. L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-Based Intelligent Intrusion Detection System in Internet of Vehicles," in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9013892.
26. L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet Things J*, vol. 9, no. 1, pp. 616–632, Jan. 2022, doi: 10.1109/JIOT.2021.3084796.
27. L. Ashiku and C. Dagli, "Network Intrusion Detection System using Deep Learning," *Procedia Comput Sci*, vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.
28. V. Hnamte and J. Hussain, "DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System," *Telematics and Informatics Reports*, vol. 10, p. 100053, Jun. 2023, doi: 10.1016/j.teler.2023.100053.
29. F. L. Becerra-Suarez, I. Fernández-Roman, and M. G. Forero, "Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing," *Mathematics*, vol. 12, no. 9, p. 1294, Apr. 2024, doi: 10.3390/math12091294.
30. A. A. Alahmadi *et al.*, "DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions," *Electronics (Basel)*, vol. 12, no. 14, p. 3103, Jul. 2023, doi: 10.3390/electronics12143103.
31. A. A. Alahmadi *et al.*, "DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions," *Electronics (Basel)*, vol. 12, no. 14, p. 3103, Jul. 2023, doi: 10.3390/electronics12143103.
32. R. Ahsan, W. Shi, and J. Corriveau, "Network intrusion detection using machine learning approaches: Addressing data imbalance," *IET Cyber-Physical Systems: Theory & Applications*, vol. 7, no. 1, pp. 30–39, Mar. 2022, doi: 10.1049/cps2.12013.