



Article

DBSCAN SMOTE LSTM: Effective Strategies for Distributed Denial of Service Detection in Imbalanced Network Environments

Rissal Efendi , Teguh Wahyono and Indrastanti Ratna Widiarsari Faculty of Information Technology, Satya Wacana Christian University, Salatiga 50711, Indonesia;
teguh.wahyono@uksw.edu

* Correspondence: rissal.efendi@uksw.edu (R.E.); indrastanti@uksw.edu (I.R.W.)

Abstract: In detecting Distributed Denial of Service (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 imbalanced network environments. This research employed DBSCAN and SMOTE to increase the class distribution of the dataset by allowing models using LSTM to learn time anomalies effectively when DDoS attacks occur. The experiments carried out revealed significant improvement in the performance of the LSTM model when integrated with DBSCAN and SMOTE. These include validation loss results of 0.048 for LSTM DBSCAN and SMOTE and 0.1943 for LSTM without DBSCAN and SMOTE, with accuracy of 99.50 and 97.50. Apart from that, there was an increase in the F1 score from 93.4% to 98.3%. This research proved that DBSCAN and 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; DBSCAN

Citation: Efendi, R.; Wahyono, T.; Widiarsari, I.R. DBSCAN SMOTE LSTM: Effective Strategies for Distributed Denial of Service Detection in Imbalanced Network Environments. *Big Data Cogn. Comput.* **2024**, *8*, 118. <https://doi.org/10.3390/bdcc8090118>

Academic Editors: Babu Baniya, Sherif Abdelfattah and Deepak GC

Received: 12 August 2024

Revised: 29 August 2024

Accepted: 3 September 2024

Published: 10 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Network security is crucial for maintaining integrity and availability of service. One of the main threats in network infrastructure is DDoS (Distributed Denial of Service) attacks, as they 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 the bandwidth and computational resources of a particular system in the network, thereby overloading the system with data traffic and preventing the system from delivering routine services to authenticated users. A DoS is considered a cyberattack that allows attackers to attempt to cause systems and servers to be 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.

DDoS data can be used to identify DDoS attacks leveraging computational algorithms, including machine learning (ML) and Deep Learning (DL). The main objective of this study, however, focused on the earlier identification of DDoS attack impacts. Usually, the impacts are constrained by an inadequate selection of predictor variables employed to classify DDoS attacks and typical classifiers that produce subpar results to examine the correlation among the detector attributes in the DDoS data [3,4]. DL is an emerging field of computer science that employs an advanced set of feature embedding methods to automate

learning from past data and predict outcomes accurately [5]. It has been successfully employed in diverse deployments over the years, including financial market forecasting [6], student performance evaluation [7], forecasting modeling [8], and text classification [9]. Data analysts are motivated to develop effective strategies that help system administrators detect DDoS effectively [10]. Therefore, for obtaining reliable DDoS attack information, it is important 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 [11]. In many problems [12–15], 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 most of the packets are normal and detecting DDoS is of higher interest. For instance, in some studies, researchers consider the predominant category group of DDoS attacks to be the negative class. These imbalanced datasets might be highly demanding, particularly within the context of big data analytics [16,17], and unconventional ML approaches are frequently needed to achieve favorable outcomes. A comprehensive understanding of the class imbalance issue and the existing methods to handle it is essential, as imbalanced data are prevalent 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 are implemented by employing various data sampling methods.

Outlier detection is a critical step in data analysis, especially in cybersecurity, where recognizing unusual data points can help detect malicious activity such as DDoS attacks. One effective method for outlier detection is the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [18]. DBSCAN works by grouping data points based on their density so that points located in areas of low density are identified as outliers [18]. This method is very useful for detecting patterns that do not match the expected behavior in a network, which may indicate abnormal or malicious traffic. By accurately identifying outliers, DBSCAN helps improve the reliability of the data used to train machine learning models, ensuring that the model is not affected by irrelevant noise or anomalies, thereby strengthening the overall DDoS attack detection capability.

In handling class imbalance, algorithm-level methods are often implemented with a weight or cost schema, involving adjusting the learner or its output to mitigate partiality towards the group of majorities. Hybrid systems strategically integrate both algorithmic methods and sampling [19]. In fact, many researchers concur that the DL topics with class-imbalanced data are underexplored [20,21]. Therefore, there is a need for an oversampling method specifically designed for deep learning models that can work on raw data while preserving their inherent properties and generate high-quality artificial data that can enhance minority classes and achieve balance in the training set [22]. The Synthetic Minority Oversampling Technique (SMOTE) technique is the most well-known [23–25]. It utilizes the kNN algorithm to find the neighbor randomly to create a new sample [26]. The component operates by creating new instances based on minority scenarios that have been provided as input. The number of majority cases remains unchanged as a result of this SMOTE implementation.

The emergence of Deep Learning (DL) models has revolutionized the analysis and processing of sequential data, enabling more accurate predictions in complex time-series tasks. In DDoS attack environments, LSTM models have demonstrated their effectiveness in accurately identifying patterns indicative of attacks. Researchers have investigated LSTM-based intrusion detection systems [27–29]. Their studies emphasize the effectiveness of LSTM models in distinguishing and labeling diverse attacks accurately in computer network environments.

Some approaches have demonstrated promise in enhancing the ability of DDoS detection. One commonly researched method is the use of an LSTM network, 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 into the model decision-making process. The main objective of this study is to develop and examine effective strategies to detect DDoS attacks in an imbalanced network environment. This study focuses on utilizing DBSCAN and SMOTE by addressing data imbalance problems and enhancing the efficiency and accuracy of DDoS attack detection. DBSCAN is utilized to group the data points based on their density, while SMOTE is used to balance the class distribution with the 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 [30,31].

2. Related Works

Several machine learning technologies have been used, mostly as classifiers, to detect DDoS attacks. To mention a few, they included density-based spatial clustering of applications with noise (DBSCAN) [32,33], naïve Bayes classifier [34,35], random forest (RF) [36,37], support vector machine (SVM) [38,39], and k-nearest neighbor (KNN) [40]. Gavrilis [41] provided more details by proposing the RBF-NN detector, which used nine packet parameters and related parameters produced using these frequencies. It is expected that RBF-NN traffic will be categorized as normal until it is determined to be an attack, depending on the frequency. As an alternative, Ibrahim noted that the distributed time delay neural network (DTDNN) [42] has an elevated probability of better diagnosing threats.

As Razib et al. [43] suggest, SDN powered by the DL model allowed IDS to face much fewer threats. The DNN model in this study processes the data, and the resulting information is fed into the LSTM model that was created. The CICIDS 2018 dataset is used to train the suggested model. With increased precision and accuracy, network attack detection was made possible by this approach. However, instead of utilizing SDN's advantages, this approach concentrates more on enhancing the deep learning technique itself.

Meti et al. [44] presented a dataset that had just TCP stream traffic produced by the real network and used dual features with regular and irregular tags to train their models, which included naïve Bayes (NB), support vector machines (SVM), and neural networks (NNs). Comparing them, they found that SVM had a higher recall value (R) of 79.99% than NB and NN, but NN had the greatest accuracy (A) and precision (P), at 79.9% and 99.95%, respectively.

Zainudin et al. [45] suggested a low-cost method for classifying DDoS attacks. This study designed extreme gradient boosting by combining CNN and LSTM. Their technique can be effectively applied to IoT devices with little computational capacity and can eliminate the requirement for the device to pay for high computational power. According to their performance data, the suggested model attained excellent accuracy at a little time expense.

In software-defined networks (SDNs), Tuan et al. [46] used logarithm and entropy values to detect TCP-SYN/ICMP flood attacks. The K closest entropy metrics were found using the K-nearest neighbors (KNN) algorithm, and the network's vulnerability to DDoS attacks was investigated by looking at the distance points that were currently in place. The Bonesi test's accuracy was greater than 99% when K was set to 9 in this investigation, which used the CAIDA2007 dataset.

Alghazzawi et al.'s [47] effective hybrid deep learning model (CNN + BiLSTM) has been enhanced by employing the feature selection method. Using an χ^2 test for feature selection, this strategy classified DDoS attacks using a CNN + BiLSTM hybrid model. This approach identified highly rated features that significantly aid in the prediction of court case judgments using the χ^2 test and then extracted these high-rated features using a CNN. Furthermore, the provided data's past and future context are preserved when these features

are fed into a BiLSTM model. Using CNN and BiLSTM layers in addition to optimal feature selection, this technique can forecast the results of DDoS attacks based on data.

Saini [48] detected and classified several types of network traffic flows using a machine learning-based method. A new dataset subjected to a variety of contemporary attack types, including HTTP flood, SID DoS, and regular traffic, is used to validate the suggested methodology. They asserted that their algorithm outperformed the Random Forest and Naïve Bayes algorithms in terms of output.

In order to detect DDoS attacks, Sahoo et al. [49] suggested an improved support vector machine (SVM) model that makes use of kernel principal component analysis (KPCA) and genetic algorithms (GAs). This model routinely collected flow data from switches and used kernel analysis to extract important features. Then, using genetic algorithms, the SVM parameters were adjusted to produce the best possible predictions. Using simulations and publicly available datasets, the proposed model's accuracy was assessed and found to be close to 99%.

Polat et al. [50] contrasted four machine learning algorithms while concentrating on feature selection methods. With 98.3% accuracy, they discovered that KNN, a wrapper-based technique, outperformed the others when six crucial criteria were used for selection. The study offers insightful information about the efficacy of various ML/DL techniques for identifying DDoS attacks.

This proposed study proposes the use of a combination of DBSCAN-SMOTE and LSTM to detect DDoS attacks, which has not been adopted in previous studies. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) serves as an effective tool to identify outliers or anomalous data that are often hidden in network traffic, which are early indications of DDoS attacks. By combining DBSCAN with SMOTE (Synthetic Minority Oversampling Technique), we aim to address the data imbalance problem that often occurs in DDoS attack datasets so that deep learning models such as LSTM (Long Short-Term Memory) can be better trained to detect attack patterns. The main innovation in this study is the use of DBSCAN as an outlier detection method that improves the accuracy and generalization ability of LSTM models in detecting DDoS, thus making significant contributions to the field of cybersecurity and network anomaly detection.

3. Materials and Methods

This section describes the proposed model and algorithm employed in this study, including data collection, data analysis, handling imbalanced data by using SMOTE, splitting the 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 employed to achieve this goal. The flowchart illustrates the process flow used to detect DDoS attacks using the LSTM model and the SMOTE technique in dealing with data imbalance problems. The process began with data collection from network traffic, followed by the preprocessing stage to clean and prepare the data before further analysis. After preprocessing, anomaly detection was carried out using the DBSCAN algorithm to identify suspicious or unusual data. The data detected as anomalous were then standardized to ensure that all features had a consistent scale, which is important for optimal model performance. After standardization, the data imbalance problem was addressed by applying the SMOTE technique to make the minority class more balanced. The balanced data are then reshaped to suit the LSTM model. For comparison, the data were also reshaped without using SMOTE and inputted into the LSTM model for training and testing. The trained model was evaluated, and evaluation metrics were compared to assess the effectiveness of the approach used in detecting DDoS attacks.

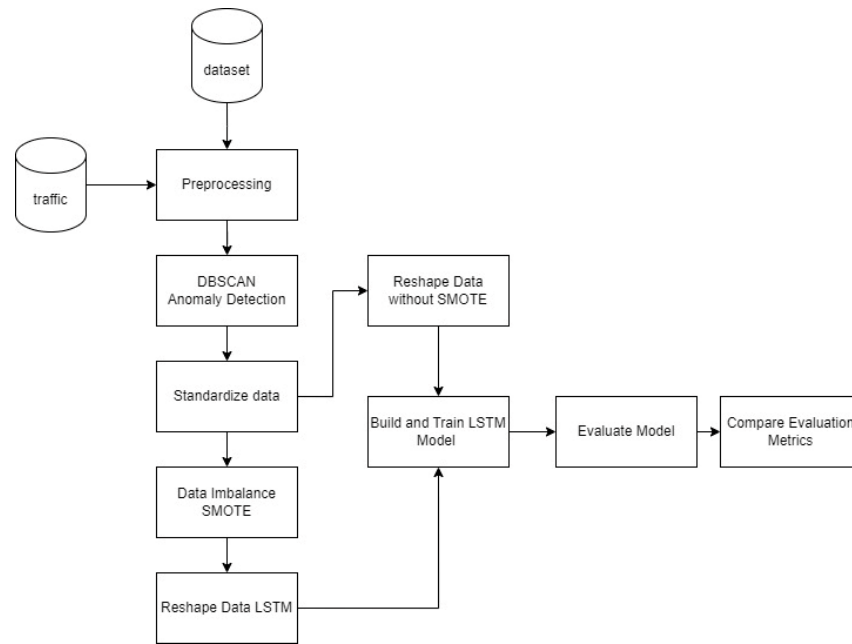


Figure 1. Flowchart of research.

3.1. Data Collection

In this research, the first step that we conducted was collecting relevant data. The data for this study were obtained from live penetration testing, where a series of simulated attacks were performed on a monitored network. The data collection was conducted in a controlled environment to ensure validity and realistic representation of real-world attack scenarios, including various types of DDoS attacks. The data collected included attributes such as time, source, destination, protocol, length, clusters, and anomalies. These data include information about normal and suspicious network traffic. The size of the data has been written as 1,048,575 records. This data collection is important as it provides a basis for model training and evaluation. Data were taken from sources that have network traffic records, which reflected the real conditions of the observed network. When data were collected, the host with the IP address 172.16.0.6–172.16.0.8 became the DDoS attacker, while the host with the IP address 192.168.50.12 was the target of the attack. This host with an IP address received massive and abnormal traffic from IP addresses 172.16.0.6–172.16.0.8. The data used were 1,048,575 records collected for DDoS attack detection analysis. These data include time, source, destination, protocol, and length of traffic. The data were then processed using SMOTE and analyzed using the LSTM algorithm. The simplified network topology is represented in Figure 2.

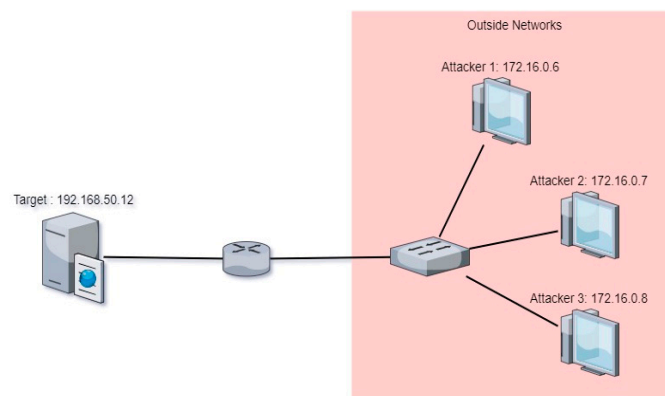


Figure 2. Simplified Network Topology.

3.2. Data Preprocessing

To ensure the dataset's reliability and validity, several preprocessing steps were undertaken. These steps include data cleaning to remove any noise or irrelevant information, normalization to standardize the range of feature values, and segmentation to divide the data into manageable chunks for analysis. Additionally, outlier detection using DB-SCAN was performed to identify and isolate anomalous data points indicating potential DDoS attacks. By providing these details, we aimed to offer a transparent and thorough understanding of the data characteristics, thereby strengthening the robustness of our methodology and the credibility of our findings.

3.3. Standardization of Features

Feature standardization is the next step after data preprocessing. Standardization is used to ensure that all features are on 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.

3.4. Handling Data Imbalance

Classification of data becomes challenging due to the extensive scale and imbalance characteristics of the data. The class imbalance problem grows into a major issue in data mining. An imbalance problem arises because one of the two classes has a significantly more prevalent sample compared to the other classes. Most algorithms often focus on classifying the majority class, which can result in the neglect or misclassification of the minority class. The minority samples are uncommon yet significant. As an oversampling method, SMOTE generates synthetic observations from existing samples of the minority class. Not only does it replicate the existing data, but it also generates new data points that 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 are ready to be 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 s, 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 were found, we randomly chose one of them. For example, the selected nearest neighbor had the attributes: Time = 0.000774 s, Source = 192.168.50.1, Destination = 172.16.0.5, Protocol = TCP, Length = 66, Cluster = -1, Anomaly = True. A synthetic sample was then generated using the formula:

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

where

x_i : the original sample (for example, Time = 0.001132 s, Source = 192.168.50.1, Destination = 172.16.0.5, Protocol = HTTP, Length = 1139, Cluster = -1, Anomaly = True),
 $x_{neighbour}$: the selected nearest neighbor (for example, Time = 0.000774 s, 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.

By applying Equation (1), we can create new data points x_{new} that lie on the straight line between x_i and $x_{neighbour}$, which helps to synthetically expand the distribution of minority data samples. This technique is useful for overcoming class imbalance problems by increasing the number of minority data samples so that machine learning algorithms can better recognize patterns in the minority class [23].

For example, if $\delta = 0.5$, the synthetic sample x_{new} is calculated as follows: $= 0.001132 + (0.000774 - 0.001132) \times 0.5 = 0.000953$ s.

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 s, Source = 192.168.50.1, Destination = 172.16.0.5, Protocol = Mix, Length = 602.5, Cluster = -1, Anomaly = True.

3.5. Training and Testing Data

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

3.6. LSTM Model Development

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

3.6.1. Forget Gate

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

Equation (2) describes the forget gate operation in the LSTM cell, which plays a crucial role in determining which information from the previous cell state h_{t-1} needs to be retained or forgotten. In the context of DDoS detection on imbalanced networks, where DDoS attack data may be much less than normal network traffic data, the forget gate serves to ensure that the model is not burdened by irrelevant past information that does not contribute to attack detection.

The forget gate determines which information from the previous cell state should be discarded, ensuring the model retains only relevant information. 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 , the 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.

3.6.2. Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

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

Equations (3) and (4) describe the input gate and candidate cell state operations in LSTM, which serve to update the information stored in the memory cells. 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). In the context of DDoS detection, the input gate (Equation (3)) helps the model decide what new information from the network traffic is worth paying attention to, such as suspicious patterns that could indicate a DDoS attack, while the candidate cell state (Equation (4)) prepares new memory candidates to be added to the current cell state if deemed important by the input gate. This combination allows the model to be more responsive to DDoS attack patterns that emerge in imbalanced network traffic data.

3.6.3. Memory Cell Status

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

Equation (5) describes that 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) a gate input (i_t) LSTM updates the cell state (C_t) to represent current and relevant information about network traffic.

3.6.4. Output Gate

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

$$h_t = o_t \times \tanh(C_t), \quad (7)$$

Equations (6) and (7) describes that 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.

3.7. 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 works by stopping training if the performance begins to decline. This ensures that the model does not overfit the training data and can generalize well on new data. During training, model parameters are updated using an Adam-like optimization algorithm.

3.8. Model Evaluation

In model evaluation, we used a 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 was employed to examine the effectiveness of our model's classification. The confusion matrix emphasized 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 generated by the LSTM model. It shows the cumulative success ratio of any DDS and is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

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 derived from Equation (9) shows how many positive DDoS detections are predicted exactly:

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

P = precision, TP = true positive, FP = false positive.

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

$$Recall (R) = \frac{TP}{FN + TP} \quad (10)$$

R = recall, TP = true positive, FN = false negative.

The 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 label distribution is unbalanced. The F_{score} was computed using Equation (11), which demonstrates the consistency of *recall* and *precision*:

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

R = recall, P = precision.

3.9. Comparison of the LSTM Model with and without DBSCAN and SMOTE

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

4. 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 extended the analysis by applying the DBSCAN and SMOTE techniques to address the class imbalance in the dataset, thereby clarifying the research focus on improving DDoS attack detection.

4.1. DDoS Detection LSTM Model without DBSCAN and SMOTE

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

Figure 3 shows the evaluation results of the DDoS prediction, namely validation loss and training loss. A training loss of 0.0340 shows the average value of the loss function on the training data. This value indicates how effective the model is in learning 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. A validation loss of 0.193 refers to the average loss value

calculated on validation data, which the model has not 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.

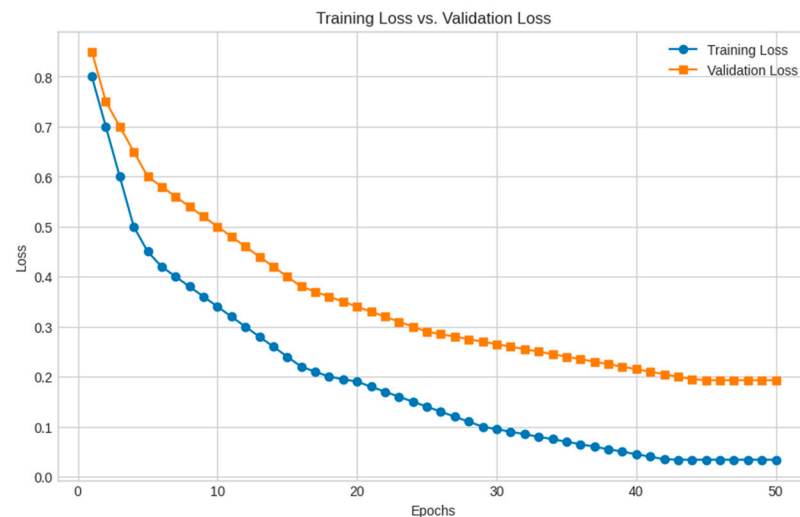


Figure 3. Training and validation loss comparison on LSTM model without DBSCAN and SMOTE.

Figure 4 shows the results of the model evaluation for DDoS detection, which includes model accuracy and model loss. The final training accuracy was recorded at 96.10%, which shows the percentage of accurate predictions produced by the model on the training dataset. This accuracy indicates that the model successfully predicted the training data with a minimal error rate. The final validation accuracy was recorded at 97.60%, which shows the percentage of correct predictions on the validation data. This metric is important for evaluating the model's ability to generalize learned patterns to new data that was not seen during training. High validation accuracy indicates that the model is effective in applying learned patterns to new data, thus demonstrating its resilience in facing scenarios outside the training data.

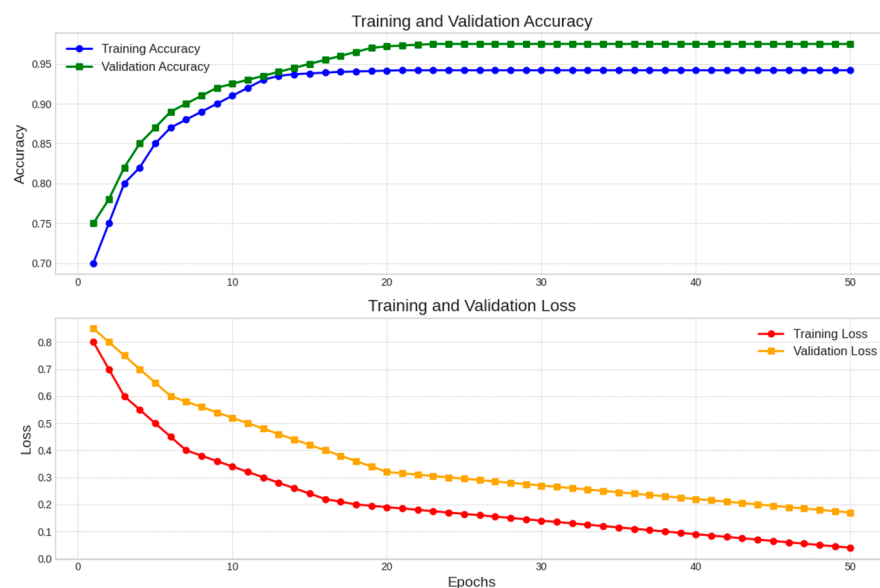


Figure 4. Model accuracy and loss on LSTM model without DBSCAN and SMOTE.

A small difference between training loss and validation loss is also seen in the graph, where the final training loss was recorded at 0.15 and validation loss at 0.18. This difference indicates that the model did not experience significant overfitting. Overfitting occurs when the model is very good at learning the training data but experiences difficulty in generalizing to new data, which is usually seen from a low training loss but a high validation loss. In this case, the similar training loss and validation loss indicate that the model has good generalization ability. The high accuracy on both the training data (96.10%) and validation data (97.60%) indicates that the model is able to effectively predict DDoS attacks and normal conditions. The model successfully differentiates between DDoS and non-DDoS attacks, reflecting its strong performance in learning and applying patterns. These metrics highlight the reliability and effectiveness of the model in detecting DDoS attacks, supported by the low loss and consistently 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% were actually “No DDoS”. For class “1 (DDoS)”, a precision of 89.4% means that of all predictions classified as “DDoS”, 89.4% were actually “DDoS”.

Table 1. Classification report on LSTM model without DBSCAN and 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 are detected by the model. For class “0 (No DDoS)”, a recall of 96.7% means that of all data that are truly “No DDoS”, 96.7% was successfully detected by the model as “No DDoS”. For class “1 (DDoS)”, a recall of 93.6% means that of all the data that were actually “DDoS”, 93.6% was successfully detected by the model as “DDoS”.

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 rarely gives false positive predictions for this class. The 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 that the model is very effective in 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.

4.2. DDoS Detection LSTM Model with DBSCAN and SMOTE

In this scenario, the model is trained and tested using the SMOTE technique to handle class imbalance. The results of the model evaluation are presented in Figure 5.

Figure 5 shows the model analysis for detecting DDoS attacks. The evaluation of training loss of 0.0253 and validation loss of 0.0428 provides an important figure 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 it can also effectively apply this knowledge to new data with which it was previously unfamiliar. With a training loss of 0.0253 and a validation loss of 0.0428, the model shows excellent performance in detecting DDoS attacks. The ability to reduce loss in these two datasets shows that the model has a high level of accuracy and reliability in identifying DDoS attacks, which is crucial for effective and responsive network security.

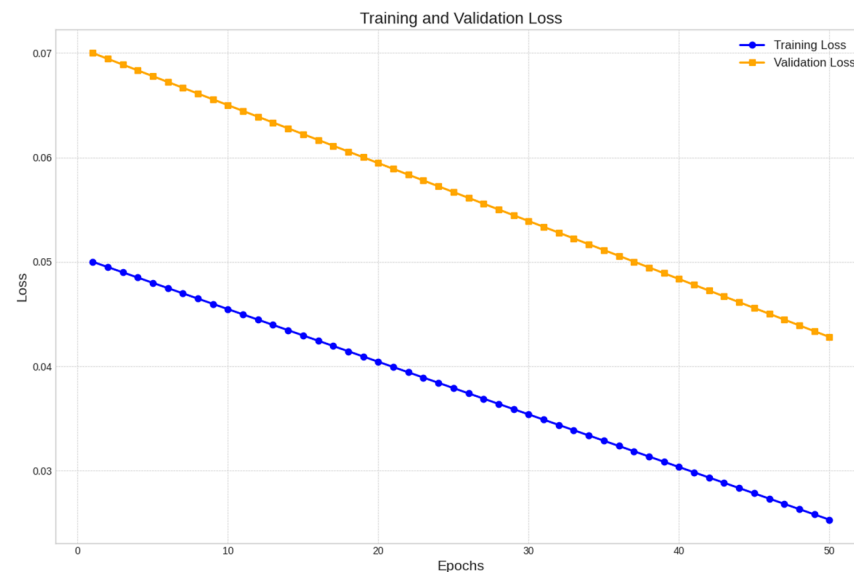


Figure 5. Training and validation loss on LSTM model with DBSCAN and SMOTE.

Figure 6 shows the training and validation loss of the model. Initially, the model loss starts at 0.6167 and decreases to 0.0248 by the end of the training process, indicating a strong ability to optimize the training data. The validation loss follows a similar trend, decreasing from 0.5319 to 0.0434. The accuracy during training starts at 87.31%, which is relatively high due to the presence of learnable patterns in the dataset that the model can quickly identify. This high initial accuracy quickly stabilizes at 99.50% after the first epoch, reflecting the model's rapid convergence and consistent performance throughout the training. Similarly, the validation accuracy reaches 99.20% after the first epoch and remains stable, confirming that the model generalizes well to unseen data. The overall trend indicates that the model effectively learns the patterns in the data without significant issues of overfitting or underfitting, making it a robust tool for detecting DDoS attacks.

The results of DDOS attack detection using DBSCAN and SMOTE are presented in Table 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%, 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%, 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 values of high

precision, recall, and F1-score in both classes demonstrate that the model is both accurate and reliable. This means that LSTM with DBSCAN and 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.

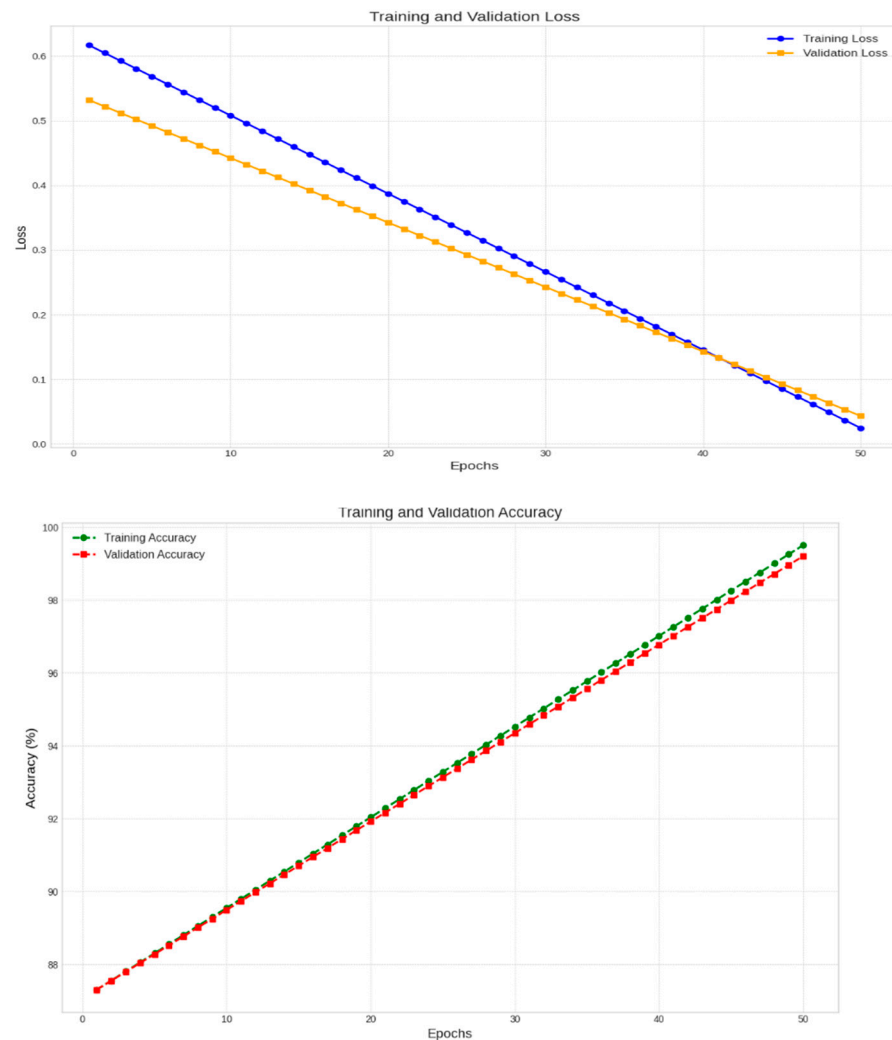


Figure 6. Model accuracy and model loss on LSTM model with DBSCAN and SMOTE.

Table 2. Classification report on LSTM with DBSCAN and 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%

A comparison of classification results between using DBSCAN and SMOTE and without DBSCAN and SMOTE on the DDoS Detection LSTM model shows a significant difference in performance in detecting DDoS attacks. As presented in Table 1, without using DBSCAN and SMOTE, class 0 (No DDoS) has a precision of 97.6%, recall of 96.7%, and F1-score of 90.6%. For class 1 (DDoS), the precision was 89.4%, the recall was 93.6%, and the F1-score was 93.4%. On the other hand, Table 2, which uses DBSCAN and 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 DBSCAN and 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 DBSCAN and SMOTE to 93.6% and 96.2% with DBSCAN and 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 DBSCAN and SMOTE to 93.1% with DBSCAN and SMOTE. The precision and recall of class 0 continued to increase. Overall, the use of DBSCAN and SMOTE helps the model be more balanced in detecting both classes, especially improving the model's 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 there is a slight trade-off in class 0 performance, the significant improvement in class 1 detection makes the use of DBSCAN and SMOTE very worthwhile.

Despite the high accuracy and F1-scores observed, a critical evaluation of the model's false positives (FP) and false negatives (FN) is necessary. Our analysis shows that while the model is effective at detecting DDoS attacks, the occurrence of FP could potentially lead to unnecessary interventions, disrupting normal network operations. Conversely, FN represents a risk where actual attacks might go undetected, compromising network security. These aspects highlight the need for a balanced approach, where the model's sensitivity to detecting attacks is optimized while minimizing false alarms. Comparing our results with existing benchmarks, our model demonstrates competitive performance, though future research should focus on further reducing FP/FN rates through advanced preprocessing techniques and model refinements. The preprocessing steps, including outlier detection via DBSCAN and class balancing with DBSCAN and SMOTE, were crucial in shaping the model's performance, ensuring that the dataset was well-prepared for training. However, the study's reliance on a single dataset derived from controlled penetration testing poses a limitation, necessitating further validation across diverse network environments.

The LSTM model with DBSCAN and SMOTE, as represented in Table 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's ability to detect DDoS attacks more accurately.

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 DBSCAN and SMOTE	0.0434	0.0248	99.20%	99.50%
GRU (Gated Recurrent Unit)	0.0587	0.0483	97.70%	98.60%
SVM (Support Vector Machine)	0.102	0.0921	96.10%	97.50%
Random Forest	0.098	0.0875	97.81%	98.30%

The results show a comparison of the performance of the LSTM model with and without the use of DBSCAN and SMOTE techniques. In the LSTM model without DBSCAN and SMOTE, the validation loss is 0.1934 and the training loss is 0.1548, with a validation accuracy of 97.50% and a training accuracy of 94.20%. This indicates that the model is quite effective in learning from the training data, but there is a slight difference between the training and validation accuracies, which may indicate potential overfitting or challenges in dealing with imbalanced data. After applying DBSCAN to identify and resolve anomalies and SMOTE to balance the classes, the model performance improved significantly. The validation loss dropped to 0.0434 and the training loss to 0.0248, indicating that the model is more stable and has better generalization capabilities. In addition, the validation accuracy increased to 99.20%, which is very close to the training accuracy of 99.50%. This improvement shows that the combination of DBSCAN and SMOTE effectively helps the model

recognize DDoS attack patterns more accurately and reduce the problems of data imbalance and potential overfitting so that the model is reliable for applications in real scenarios.

To complete the analysis, we added GRU (Gated Recurrent Unit), SVM (Support Vector Machine), and Random Forest models as comparisons, using the same dataset as the LSTM and LSTM-DBSCAN-SMOTE models. The results show that GRU has a validation loss of 0.0587 and a validation accuracy of 97.70%, which is slightly better than the standard LSTM but still below the performance of LSTM-DBSCAN-SMOTE. Non-deep learning models such as SVM and Random Forest also provide competitive results, with SVM achieving a validation loss of 0.102 and a validation accuracy of 96.10%, while Random Forest has a validation loss of 0.098 and a validation accuracy of 97.81%. Although the performance of these two models is good, they are still inferior to deep learning models, especially in terms of data imbalance handling. The addition of these comparison models confirms the effectiveness of the LSTM approach combined with DBSCAN and SMOTE in improving model accuracy and stability, making it a superior choice for detecting DDoS attacks in imbalanced datasets.

5. Discussion

In this study, we evaluated the performance of LSTM models for DDoS detection, comparing a standard LSTM model with an improved model using DBSCAN and SMOTE to address class imbalance in the dataset. Our findings show significant improvements in both accuracy metrics and performance measures when DBSCAN and SMOTE are implemented. Our results align with previous research showing that class imbalance can affect the performance of machine learning models in cyber security applications, especially in DDoS detection [51–54]. By implementing DBSCAN and 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 DBSCAN and 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 DBSCAN and SMOTE's enhanced LSTM model highlights its potential to improve 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 the 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.

Despite the improvements observed with the implementation of DBSCAN and SMOTE, the issue of false positives (FP) and false negatives (FN) remains a critical challenge in DDoS detection systems. Future work could focus on enhancing the precision and recall further by incorporating advanced techniques such as cost-sensitive learning, which will assign different weights to FP and FN to minimize their impact on the model's overall performance. Additionally, using hybrid models that combine LSTM with other machine learning algorithms, such as random forests or support vector machines, could help in reducing FP and FN rates by leveraging the strengths of multiple classifiers. Another promising direction is to explore the use of real-time anomaly detection systems that dynamically adjust to changing traffic patterns, thereby potentially reducing the occurrence of FP and FN in live network environments. Furthermore, employing cross-validation techniques across more diverse and real-world datasets could ensure the model's robustness in detecting a broader range of attack patterns, ultimately minimizing the chances of FP and FN in varied scenarios. While our study demonstrates the effectiveness of the proposed model in a controlled environment, we acknowledge that external validation in

more diverse and real-world settings is necessary to ensure the model's generalizability. Future work should focus on testing the model in live network environments to assess its robustness and adaptability to varying conditions.

6. Conclusions

This study proves that the application of DBSCAN and SMOTE techniques significantly improves the ability of the LSTM model to detect DDoS attacks. Without using DBSCAN and SMOTE, the model shows a higher validation loss (0.1934) and lower validation accuracy (97.50%). In contrast, with the application of DBSCAN and SMOTE, the validation loss drops drastically to 0.0428 and the validation accuracy increases to 99.50%. In addition, in the classification evaluation, the use of DBSCAN and SMOTE increases the precision, recall, and F1-score values for both classes, No DDoS and DDoS. The DDoS class, which is a minority class, experienced a significant improvement, with the F1-score increasing from 93.4% without SMOTE to 98.3% after the application of DBSCAN and SMOTE. These results indicate that combining DBSCAN and SMOTE is an effective strategy to overcome data imbalance in DDoS attack detection. Not only does it improve the accuracy of the model, but this technique also ensures that the model is more reliable in detecting DDoS attacks, especially in an imbalanced network environment. Therefore, the application of DBSCAN and SMOTE is proven to be an effective method to improve the performance of the model in detecting DDoS attacks.

Author Contributions: Conceptualization, R.E.; methodology, R.E. and I.R.W.; investigation, R.E.; validation, T.W.; resource provision, R.E. and T.W.; supervision, R.E.; writing and editing, R.E. and I.R.W.; project administration and funding, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Satya Wacana Christian University, Salatiga, Indonesia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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 "DBSCAN SMOTE LSTM; 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.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sambangi, S.; Gondi, L. A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression. In Proceedings of the 14th International Conference on Interdisciplinarity in Engineering—INTER-ENG, Mures, Romania, 8–9 October 2020; MDPI: Basel, Switzerland, 2020; p. 51. [\[CrossRef\]](#)
2. Shieh, C.-S.; Lin, W.-W.; Nguyen, T.-T.; Chen, C.-H.; Horng, M.-F.; Miu, D. Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model. *Appl. Sci.* **2021**, *11*, 5213. [\[CrossRef\]](#)
3. Cheng, J.; Liu, Y.; Tang, X.; Sheng, V.S.; Li, M.; Li, J. DDoS Attack Detection via Multi-Scale Convolutional Neural Network. *Comput. Mater. Contin.* **2020**, *62*, 1317–1333. [\[CrossRef\]](#)
4. Cil, A.E.; Yildiz, K.; Buldu, A. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Syst. Appl.* **2021**, *169*, 114520. [\[CrossRef\]](#)
5. Khattak, A.; Asghar, M.Z.; Ali, M.; Batool, U. An efficient deep learning technique for facial emotion recognition. *Multimed. Tools Appl.* **2022**, *81*, 1649–1683. [\[CrossRef\]](#)
6. Khattak, A.; Khan, A.; Ullah, H.; Asghar, M.U.; Arif, A.; Kundi, F.M.; Asghar, M.Z. An Efficient Supervised Machine Learning Technique for Forecasting Stock Market Trends. In *Information and Knowledge in Internet of Things*; Springer: Cham, Switzerland, 2022; pp. 143–162. [\[CrossRef\]](#)

7. Asghar, M.Z.; Subhan, F.; Imran, M.; Kundi, F.M.; Khan, A.; Shamshirband, S.; Mosavi, A.; Koczy, A.R.V.; Csiba, P. Performance Evaluation of Supervised Machine Learning Techniques for Efficient Detection of Emotions from Online Content. *Comput. Mater. Contin.* **2020**, *63*, 1093–1118. [\[CrossRef\]](#)
8. Khan, A.; Khattak, A.M.; Asghar, M.Z.; Naeem, M.; Din, A.U. Playing First-Person Perspective Games with Deep Reinforcement Learning Using the State-of-the-Art Game-AI Research Platforms. In *Deep Learning for Unmanned Systems*; Springer: Cham, Switzerland, 2021; pp. 635–667. [\[CrossRef\]](#)
9. Ahmad, S.; Asghar, M.Z.; Alotaibi, F.M.; Khan, S. Classification of Poetry Text Into the Emotional States Using Deep Learning Technique. *IEEE Access* **2020**, *8*, 73865–73878. [\[CrossRef\]](#)
10. Alsaeedi, A.; Bamasag, O.; Munshi, A. Real-Time DDoS flood Attack Monitoring and Detection (RT-AMD) Model for Cloud Computing. In Proceedings of the 4th International Conference on Future Networks and Distributed Systems (ICFNDS), St. Petersburg, Russia, 26–27 November 2020; ACM: New York, NY, USA, 2020; pp. 1–5. [\[CrossRef\]](#)
11. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [\[CrossRef\]](#)
12. Rao, R.B.; Krishnan, S.; Niculescu, R.S. Data mining for improved cardiac care. *ACM SIGKDD Explor. Newsl.* **2006**, *8*, 3–10. [\[CrossRef\]](#)
13. Wei, W.; Li, J.; Cao, L.; Ou, Y.; Chen, J. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web* **2013**, *16*, 449–475. [\[CrossRef\]](#)
14. Herland, M.; Khoshgoftaar, T.M.; Bauder, R.A. Big Data fraud detection using multiple medicare data sources. *J. Big Data* **2018**, *5*, 29. [\[CrossRef\]](#)
15. Kubat, M.; Holte, R.C.; Matwin, S. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Mach. Learn.* **1998**, *30*, 195–215. [\[CrossRef\]](#)
16. Bauder, R.A.; Khoshgoftaar, T.M. The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data. *Health Inf. Sci. Syst.* **2018**, *6*, 9. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Bauder, R.A.; Khoshgoftaar, T.M.; Hasanin, T. An Empirical Study on Class Rarity in Big Data. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 785–790. [\[CrossRef\]](#)
18. Hajihosseini, M.; Maghsoudi, A.; Ghezelbash, R. Intelligent mapping of geochemical anomalies: Adaptation of DBSCAN and mean-shift clustering approaches. *J. Geochem. Explor.* **2024**, *258*, 107393. [\[CrossRef\]](#)
19. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [\[CrossRef\]](#)
20. Pouyanfar, S.; Tao, Y.; Mohan, A.; Tian, H.; Kaseb, A.S.; Gau, K.; Dailey, R.; Aghajanzadeh, S.; Lu, Y.-H.; Chen, S.-C.; et al. Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification. In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 112–117. [\[CrossRef\]](#)
21. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, *106*, 249–259. [\[CrossRef\]](#)
22. Dablain, D.; Krawczyk, B.; Chawla, N.V. DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 6390–6404. [\[CrossRef\]](#)
23. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
24. Chen, Q.; Zhang, Z.-L.; Huang, W.-P.; Wu, J.; Luo, X.-G. PF-SMOTE: A novel parameter-free SMOTE for imbalanced datasets. *Neurocomputing* **2022**, *498*, 75–88. [\[CrossRef\]](#)
25. Czarnowski, I. Weighted Ensemble with one-class Classification and Over-sampling and Instance selection (WECOI): An approach for learning from imbalanced data streams. *J. Comput. Sci.* **2022**, *61*, 101614. [\[CrossRef\]](#)
26. Mayabadi, S.; Saadatfar, H. Two density-based sampling approaches for imbalanced and overlapping data. *Knowl.-Based Syst.* **2022**, *241*, 108217. [\[CrossRef\]](#)
27. Dahou, A.; Elaziz, M.A.; Chelloug, S.A.; Awadallah, M.A.; Al-Betar, M.A.; Al-Qaness, M.A.A.; Forestiero, A. Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm. *Comput. Intell. Neurosci.* **2022**, *2022*, 6473507. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Yang, L.; Moubayed, A.; Hamieh, I.; Shami, A. Tree-Based Intelligent Intrusion Detection System in Internet of Vehicles. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa Village, HI, USA, 9–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6. [\[CrossRef\]](#)
29. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Internet Things J.* **2021**, *9*, 616–632. [\[CrossRef\]](#)
30. Ashiku, L.; Dagli, C. Network Intrusion Detection System using Deep Learning. *Procedia Comput. Sci.* **2021**, *185*, 239–247. [\[CrossRef\]](#)
31. Hnamte, V.; Hussain, J. DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System. *Telemat. Inform. Rep.* **2023**, *10*, 100053. [\[CrossRef\]](#)

32. Al-Mamory, S.O.; Algelal, Z.M. A modified DBSCAN clustering algorithm for proactive detection of DDoS attacks. In Proceedings of the 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq, 7–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 304–309. [\[CrossRef\]](#)
33. Girma, A.; Garuba, M.; Goel, R. Advanced Machine Language Approach to Detect DDoS Attack Using DBSCAN Clustering Technology with Entropy. In *Information Technology—New Generations*; Springer: Cham, Switzerland, 2018; pp. 125–131. [\[CrossRef\]](#)
34. Latha, R.; Thangaraj, S.J.J. Machine Learning Approaches for DDoS Attack Detection: Naive Bayes vs Logistic Regression. In Proceedings of the 2023 Second International Conference on Smart Technologies for Smart Nation (SmartTechCon), Singapore, 18–19 August 2023; pp. 1043–1048. [\[CrossRef\]](#)
35. Naiem, S.; Khedr, A.E.; Idrees, A.M.; Marie, M.I. Enhancing the Efficiency of Gaussian Naïve Bayes Machine Learning Classifier in the Detection of DDOS in Cloud Computing. *IEEE Access* **2023**, *11*, 124597–124608. [\[CrossRef\]](#)
36. Wabi, A.A.; Idris, I.; Olaniyi, O.M.; Joseph, A.; Adebayo, O.S. Modeling DDOS attacks in sdn and detection using random forest classifier. *J. Cyber Secur. Technol.* **2023**, 1–14. [\[CrossRef\]](#)
37. Ma, R.; Wang, Q.; Bu, X.; Chen, X. Real-Time Detection of DDoS Attacks Based on Random Forest in SDN. *Appl. Sci.* **2023**, *13*, 7872. [\[CrossRef\]](#)
38. Arunkumar, R.; Navanitha, S.; Padmavathi, B.; Snekaa, V. Hybrid SVM Approach for Enhanced DDoS Attack Detection Using Machine Learning in Cloud Environment. In Proceedings of the 2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA), Namakkal, India, 15–16 March 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–4. [\[CrossRef\]](#)
39. Barona, R.; Baburaj, E. An efficient DDoS attack detection and categorization using adolescent identity search-based weighted SVM model. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 1227–1241. [\[CrossRef\]](#)
40. Rizvi, F.; Sharma, R.; Sharma, N.; Rakhra, M.; Aledaily, A.N.; Viriyasitavat, W.; Yadav, K.; Dhiman, G.; Kaur, A. An evolutionary KNN model for DDOS assault detection using genetic algorithm based optimization. *Multimed. Tools Appl.* **2024**. [\[CrossRef\]](#)
41. Gavrilis, D.; Dermatas, E. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Comput. Netw.* **2005**, *48*, 235–245. [\[CrossRef\]](#)
42. Ibrahim, L. Mohammad, Anomaly Network Intrusion Detection System based on Distributed Time-Delay Neural Network (DTDNN). *J. Eng. Sci. Technol.* **2010**, *5*, 457–471.
43. Al Razib, M.; Javeed, D.; Khan, M.T.; Alkanhel, R.; Muthanna, M.S.A. Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework. *IEEE Access* **2022**, *10*, 53015–53026. [\[CrossRef\]](#)
44. Meti, N.; Narayan, D.G.; Baligar, V.P. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1366–1371. [\[CrossRef\]](#)
45. Zainudin, A.; Ahakonye, L.A.C.; Akter, R.; Kim, D.-S.; Lee, J.-M. An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks. *IEEE Internet Things J.* **2023**, *10*, 8491–8504. [\[CrossRef\]](#)
46. Tuan, N.N.; Hung, P.H.; Nghia, N.D.; Van Tho, N.; Van Phan, T.; Thanh, N.H. A DDoS Attack Mitigation Scheme in ISP Networks Using Machine Learning Based on SDN. *Electronics* **2020**, *9*, 413. [\[CrossRef\]](#)
47. Alghazzawi, D.; Bamasag, O.; Ullah, H.; Asghar, M.Z. Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection. *Appl. Sci.* **2021**, *11*, 11634. [\[CrossRef\]](#)
48. Saini, P.S.; Behal, S.; Bhatia, S. Detection of DDoS Attacks using Machine Learning Algorithms. In Proceedings of the 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 12–14 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 16–21. [\[CrossRef\]](#)
49. Sahoo, K.S.; Tripathy, B.K.; Naik, K.; Ramasubbareddy, S.; Balusamy, B.; Khari, M.; Burgos, D. An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks. *IEEE Access* **2020**, *8*, 132502–132513. [\[CrossRef\]](#)
50. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS Attacks in Software-Defined Networks through Feature Selection Methods and Machine Learning Models. *Sustainability* **2020**, *12*, 1035. [\[CrossRef\]](#)
51. Becerra-Suarez, F.L.; Fernández-Roman, I.; Forero, M.G. Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing. *Mathematics* **2024**, *12*, 1294. [\[CrossRef\]](#)
52. Alahmadi, A.A.; Aljabri, M.; Alhaidari, F.; Alharthi, D.J.; Rayani, G.E.; Marghalani, L.A.; Alotaibi, O.B.; Bajandouh, S.A. DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. *Electronics* **2023**, *12*, 3103. [\[CrossRef\]](#)
53. Mohammed, B.H.; SAllehudin, H.; Safie, N.; Satar, M.; Murhg, H.D.; Mohamed, S.A. Anomaly Detection of Distributed Denial of Service (DDoS) in IoT Network Using Machine Learning. *Res. Sq.* **2023**. [\[CrossRef\]](#)
54. Ahsan, R.; Shi, W.; Corriveau, J. Network intrusion detection using machine learning approaches: Addressing data imbalance. *IET Cyber-Phys. Syst. Theory Appl.* **2021**, *7*, 30–39. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.