# M-LSTM: Multiclass Long Short-Term Memory based Approach for Detection of DDoS Attacks

Ms. Vimal Gaur (Research Scholar),
Department of CSE, MM Engineering College, Maharishi Markandeshwar Deemed to be University, Mullana, Ambala, Haryana, India and Assistant Professor, Maharaja Surajmal Institute of Technology, Janakpuri, Delhi- 110 058) ORCID ID: 0000-0003-4097-1859
Dr. Rajneesh Kumar (Professor),
Department of CSE, MM Engineering College, Maharishi Markandeshwar Deemed to be University, Mullana, Ambala, Haryana, India ORCID ID: 0000-0002-8139-3533

**Article Info**
**Page Number: 1375-1394**
**Publication Issue:**
**Vol. 71 No. 3s2 (2022)**

**Article History**
**Article Received:** 28 April 2022
**Revised:** 15 May 2022
**Accepted: 20** June 2022
**Publication: 21** July 2022

**Abstract:** Distributed Denial of Service attack is a ubiquitous menace to computer networks. In this attack, several nodes attack the server by sending huge amount of traffic. Server in unable to identify the difference in requests from malicious users and benign users and hence processes all the requests. As a result of processing attack traffic, the whole network will come to halt after sometime. In this paper, an M-LSTM model has been proposed for early detection of DDoS attacks. We demonstrate the feasibility of this model by comparing results of binary, multiclass (grouped and ungrouped) classification long-short term model on CICDDoS2019 dataset. Experimental results show that Layer-2 LSTM Multiclass grouped classification yield maximum values of Precision, Recall and F-1 Score as 98.75%, 97.5% and 98% respectively.

## 1. Introduction

With the growth of internet, network attacks are also evolving at a great pace. DDoS attack is the most common network attack. These attacks also modifies, damages data so they are also called as active attacks. With the passage of time, attacker have come across new tools for performing these kinds of attacks. Nowadays, it has become easier for an attacker to compromise victim machine. Usually, DDoS attacks occur on IoT networks. This is due to lack of security mechanisms in IoT devices. IoT devices have limited resources and memory.

Several methods for resolution against DDoS attacks in IoT devices have been analyzed by researchers.

1. Firewalls and Traffic Filtering-: These are two network security techniques who follow a set of rules to protect network from attacks. These rules detect and block DDoS attacks by monitoring network traffic carefully. Different strategies have been proposed by researcher for prevention of these attacks (reactive, proactive) and for getting sufficient knowledge about network traffic (individual, cooperative). Different combinations (reactive + individual, reactive + cooperative, proactive + individual, proactive + cooperative) of these two techniques are selected to install filters on routers, which will block the anonymous traffic from entering the network. These filtering mechanisms can be used in SDN environment, cloud computing etc.

2. Traceback Mechanism-: A proper traceback procedure has to be initiated as soon as DDoS attack has been detected via a detection algorithm. It basically helps in identifying real origin of attacker. These traceback procedures may require special hardware or software support from ISP, while others may depend on IP addresses of routers. Researchers have proposed various traceback schemes viz. Entropy variations, Pushback, Hop by Hop Tracing, Packet Marking, Packet Logging, and ICMP messaging.
3. IDS and IPS-: Various IDS and IPS are available these days for providing security to IoT devices. IDS and IPS are considered to be most important systems for detecting and preventing against DDoS attacks. They operate upon a certain set of predefined rules and policies for identifying normal traffic and malicious traffic. They basically monitor network traffic continuously by using a set of network analyzers. IDS may operate at a host level (Host-based IDS), Network level (Network-based IDS) depending upon whether DDoS detection is done in online mode or offline mode. Further IDS can work both for machine learning and deep learning algorithms.
4. Using Entropy Variation-: Entropy is a measure of the uncertainty in flow of packets over a network. A less value of rate of entropy indicates complete benign traffic. Hence a larger value of rate of entropy means malicious traffic. Anomaly detection using entropy require a continuous monitoring of flow of data across network. This technique is considered to be very effective technique for detecting traffic patterns and hence normalized entropy can be measured effectively. A threshold value of entropy is set initially for measuring random variables. Hence, the value of calculated normalized entropy can be checked against threshold entropy value- if it is greater than threshold then we can conclude that flow of data has been received from intended user.
5. Use Software Defined Networking-: Software Defined Networking paradigm for IoT networks have been acquired for mitigating DDoS attacks in the year 2016. Since then researchers are adopting this paradigm for addressing DDoS attacks on IoT network. The main objective of adopting this paradigm is to separate data plane and control plane. Network management becomes easier as network elements (controllers, IoT Gateways) can operate in different environments (collaborative and non-collaborative).

These issues motivate the consideration of the entire CICDDoS2019 (70% data for training and 30% data for testing) dataset for experimentation purposes.

We propose an M-LSTM model for early detection of DDoS attacks. The contributions of M-LSTM model on binary, multiclass grouped and multiclass ungrouped data have been investigated. A Multilayered G-LSTM model handle multiclass DDoS attacks. These multiclass attacks are further classified into grouped and ungrouped data. Finally, multiclass grouped layer-2 LSTM model yields promising Precision, Recall, F-1 score as 98.75%, 97.5% and 98% respectively. The key contribution of this paper includes detection of DDoS attacks with efficient performance parameters.

The structure of this paper is as follows: Section 2 presents related work. This section describes the methodologies used by researchers on the CICDDoS2019 dataset. Section 3 describes the materials and methods to be followed in this paper. Results and discussions have been discussed in Section 4. Finally, section 5 highlights the conclusion and future work.

## 2. Literature Survey

This section explicitly describes the technicalities proposed by esteemed researchers for detecting Distributed Denial of Service attacks on IoT devices. Performance parameters of several machine learning algorithms have been presented along with scope of the work on CICDDoS2019 dataset in table 1.

Table 1 Methodologies Proposed on CICDDoS2019 Dataset

| S.No. | Author | Machine Learning Algorithms | Performance Parameters | Scope of Work |
|---|---|---|---|---|
| 1. | Alghoson et al. [9] | RF, LGB, CatBoost, CNN (Binary Classification) | Random Forest model offer best detection accuracy as 99.9974% for 20 features. Two feature selection methods - correlation matrix using Pearson Correlation (filter method), The Decision Tree model (embedded method) have been adopted. | Multiclass Classification must be employed. |
| 2. | Kushwah et al. [10] | ELM Model with Blackhole Optimization Algorithm | Accuracy = 99.80 | Multiclass Classification must be employed. |
| 3. | Chartuni et al. [11] | Neural Networks | Precision = 94.21%<br>Recall = 94.03%<br>F-1 Score = 94.12% | More Deep Learning algorithms can be explored. |
| 4. | Can et al. [12] | Automatic Feature selection and MLP | Precision = 91.16%<br>Recall = 79.41%<br>F-1 Score = 79.39% | Advanced Feature Selection Algorithms must be used. |
| 5. | Gaur et al. [13] | Random Forest, Decision Tree, XGBoost, SNN, DNN (Binary Classification) | 98.34% accuracy for ANOVA with XGBoost. We have applied three feature selection algorithms. | Further accuracy can be improved with more feature selection algorithms. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6. | Odumuyiwa et al. [14] | Unsupervised Machine learning Algorithms (Binary Classification) | Autoencoder | 89.45 % | 86.17% | This approach didn't use an entire dataset, the first result is for SYN_Flood, and the second is for UDP_Lag. |
| | | | Restricted Boltzman machine | 56.51 % | 50.89% | |
| | | | K-means Clustering Algorithm | 75.38 % | 71.39% | |
| | | | Expectation-Maximization Clustering Algorithm | 70.96 % | 67.59% | |
| 7. | Abbas et al. [15] | PCA is used for Pre-processing. MIX dataset (PORTMAP, LDAP) is used by Random Forest. (Binary Classification) | Random Forest gives 99.976% accuracy. | | | Data has been used in partial mode. |
| 8. | Alamri et al. [16] | LR, RF, XGBoost (Binary and Multiclass Classification) | Accuracy with Binary class<br>LR= 80%<br>RF=98.5%<br>XGBoost=99.7%<br>Accuracy with Multiclass<br>LR= 35%<br>RF=83%<br>XGBoost=91.3% | | | This approach results in less accuracy value for Multiclass. The maximum value achieved is 91.3% for XGBoost. |
| 9. | Parfenov et al. [17] | Gradient Boosting, AdaBoost, CatBoost. Extra Tree Feature Selection has also been applied. | The precision with full features<br>Gradient Boosting=97.1%, AdaBoost=61.4%, CatBoost=97.1% With Extra Tree Feature Selection<br>Gradient Boosting=97%, AdaBoost=62.3%, CatBoost=96.7%<br>These results are for 25 features. | | | Gradient boost Achieves maximum precision value, but on the application of extra tree feature selection, this |

<table>
<tr><td></td><td></td><td>(Binary and Multiclass Classification )</td><td colspan="3"></td><td>precision deteriorates.</td></tr>
<tr><td rowspan="4">10.</td><td rowspan="4">Shurman et al. [18]</td><td rowspan="4">LSTM Model (3 Variants)</td><td></td><td>Train Accuracy</td><td>Test Accurac y</td><td rowspan="4">These results are with full features.</td></tr>
<tr><td>Model I</td><td>92.05%</td><td>91.54%</td></tr>
<tr><td>Model II</td><td>97.27%</td><td>96.74%</td></tr>
<tr><td>Model III</td><td>99.85%</td><td>99.19%</td></tr>
<tr><td>11.</td><td>Rahman et al. [19]</td><td>Three machine Learning Algorithms: LR, DT, SVM (Binary Classification )</td><td colspan="3">SVM achieves the highest accuracy as 97.1%</td><td>A complete feature set has been used.</td></tr>
<tr><td>12.</td><td>Chesney et al. [20]</td><td>Logistic Regression (Binary Classification )</td><td colspan="3">Logistic regression gives an accuracy of 99.70%</td><td>The complete dataset has not been chosen for implementation . (Logistic Regression has been applied on LDAP file).</td></tr>
<tr><td>13.</td><td>Ferrag et al. [21]</td><td>CNN (Binary and Multiclass Classification )</td><td colspan="3">Binary Class- CNN = 99%<br>Multi-Class- CNN = 90%</td><td>This approach gives less accuracy for Multiclass.</td></tr>
<tr><td>14.</td><td>Sanchez et al. [22]</td><td>RF is used (Binary Classification )</td><td colspan="3">RF gives an accuracy of 99%</td><td>This dataset is used for binary classification. Hyperparamete r Tuning is done using GridSearch.</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| 15. | Elsayed et al. [23] | RNN with Autoencoder (Binary Classification ) | The proposed model DDoSNet turns out to be best with 99% accuracy | This result is for binary classification. |
| 16. | Assis et al. [24] | Gated Recurrent Units (GRU) deep learning method, CNN, LSTM, DNN, SVM, LR, KNN, and GD (Binary Classification ) | GRU achieves accuracy closer to 100% | GRU is not used as a multi-label classifier. |
| 17. | Manikumar et al. [25] | Extra Tree-Based Classifier, Three Machine learning Algorithms (KNN, DT,RF) (Binary Classification ) | KNN=87.34%, DT=93.83%, RF=95.19%. Random Forest gives maximum accuracy. | We have achieved 1.55% more accuracy for RF with an Extra tree classifier. |
| 18. | Li et al. [26] | Introduced a new variable for calculating Temporal False Omission Rate (TFOR) | Average Temporal False Omission Rate = 0.3447% and True positive rate is 100% and FPR is 3%. | Results are obtained with full features. |
| 19. | Jia et al. [27] | LSTM, CNN Model | LSTM Accuracy= 98.9%<br>CNN Accuracy=99.9% | A complete feature set has been used for these results. |
| 20. | Sharafaldin et al. [28] | Machine Learning Algorithms | ID3 gives 78% Precision Value. | Values have been obtained |

| | | | | |
|---|---|---|---|---|
| | | (ID3, RF, NB, LR) (Binary Classification) | | with full features. |
| 21. | Vuongl et al. [29] | Random Forest Regressor has been applied for selecting 24 features (Binary Classification) | The proposed method gives 99.3% precision with a grouping of labels. | Recall, Precision and F1 Score have been calculated for individual attack types. We have calculated these values after combining all the attack types. |

**3. Materials and Methods**

We analyzed a cloud-based environment called Google Colab (an online jupyter notebook environment) on CICDDoS2019 dataset. Although a few researchers have aimed at achieving good accuracy for binary classifiers, multiclass classification was not paid much attention. In this paper, we will focus on comparison of binary and multiclass classification. This is done by analyzing Precision, F-1 score and Recall as performance parameters. The main reason for not paying much attention to accuracy is that it measures near the target value and does not work well for multiclass target variable. Also with more than two classes we don't know whether all classes are being predicted equally well. This paper focusses on precision, as results of repeated measurements are achieved successfully, so it provides useful assessment. Further, F1-Score is a good measure when there is an uneven class distribution and when number of correct hits is to be achieved, recall is preferred.

It is not possible to directly calculate Precision, Recall and F-1 Score for multiclass classification problem, hence they have to be converted into micro or macro scoring methods. In this paper, we have calculated Marco averaging as this scoring method takes the arithmetic mean of all the pre calculated methods. Recall is chosen over the other methods as we are trying to reduce the number of false positives here to better optimize our model.

Multi-layer LSTM model (figure 1) have been used for binary classification and multiclass classification of data[30, 32]. Further multiclass classification have been divided into two types viz. Grouped and Ungrouped as below-:

Case I-: Binary Classification

It refers to classification, where we can identify whether an attack has occurred or not.

Case II-: Multiclass Grouped Classification

It also refers identifying one class among a range of classes but here grouping of classes have been made. Since the classes are imbalanced they have been grouped into four groups. Imbalanced class labels have been grouped into four labels as follows [29, 31]-:

ISSN: 2094-0343
2326-9865

Label 1: UDP, UDP-Lag, SYN (Reflection based attacks)
Label 2: NetBIOS, LDAP (Exploitation based attacks)
Label 3: BENIGN
Label 4: MSSQL
Case III-: Multiclass Ungrouped Classification
It refers to classification, where we can identify one class among a range of classes. Here each class is treated independently [33].

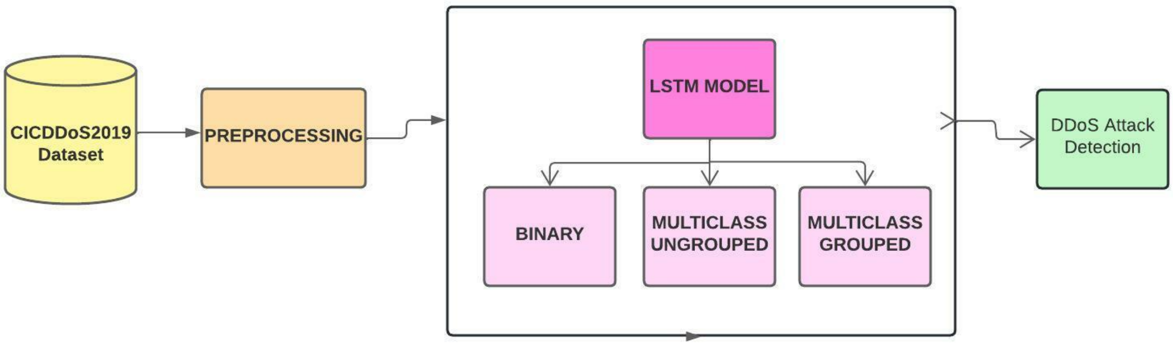


Figure 1. M-LSTM: Multiclass LSTM Model for Detection of DDoS Attacks

The algorithms used have been described below for each layer of LSTM.

| **Algorithm 1: LSTM Layer 1** |
|---|
| Input: x_train.shape[2] , batch_size = 32/64<br>Initialization: Define Sequential model : model = Sequential()<br>1:model.add(LSTM(100/50,input_dim=(seq_array.shape[1],seq_array.shape[2]), return_sequences=False))<br>2:model.add(Dense(units=64/32,activation=[‘relu’,’tanh’,’Linear’]))<br>3:model.add(Dense(units=32/16,activation=[‘relu’,’tanh’,’Linear’]))<br>4:model.add(Dense(units=16/8,activation=[‘relu’,’tanh’,’Linear’]))<br>5: model.add(Dense(units=[4,5], activation=‘softmax’))<br>6:model.compile(loss=‘sparse_categorical_crossentropy’, optimizer =‘adam’, metrics=[“accuracy”])<br>7: history= model.fit(seq_array, label_array)<br>8: epochs=100, validation_data=(val_seq_array, val_label_array)<br>9: Callbacks=[EarlyStopping(min_delta=0, patience=3,verbose=0,mode=’auto’), ReduceLRonPlateau(monitor=’loss’,min_lr=0.000001)]<br>10: training_loss = history.history[‘loss’]<br>11: test_loss = history.history[‘val_loss’]<br>12: val = model.predict(val_seq_array)<br>13: val_class = np.argmax(val,axis=1)<br>14: cm = confusion_matrix(val_label_array, val_class)<br>15: plt.show() |

| **Algorithm 2: LSTM Layer 2** |
|---|
| Input: x_train.shape[2] , batch_size = 32/64<br>Initialization: Define Sequential model : model = Sequential()<br>1:model.add(LSTM(100/50,input_dim=(seq_array.shape[1],seq_array.shape[2]), return_sequences=True))<br>2: model.add(LSTM(100/50, return_sequences=False))<br>3:model.add(Dense(units=64/32,activation=[‘relu’,’tanh’,’Linear’]))<br>4:model.add(Dense(units=32/16,activation=[‘relu’,’tanh’,’Linear’]))<br>5:model.add(Dense(units=16/8,activation=[‘relu’,’tanh’,’Linear’]))<br>6: model.add(Dense(units=[4,5], activation=‘softmax’))<br>7:model.compile(loss=‘sparse_categorical_crossentropy’, optimizer=‘adam’, metrics=[“accuracy”])<br>8: history= model.fit(seq_array, label_array)<br>9: epochs=100, validation_data=(val_seq_array, val_label_array)<br>10: Callbacks=[EarlyStopping(min_delta=0, patience=3,verbose=0,mode=’auto’), ReduceLRonPlateau(monitor=’loss’,min_lr=0.000001)]<br>11: training_loss = history.history[‘loss’]<br>12: test_loss = history.history[‘val_loss’]<br>13: val = model.predict(val_seq_array)<br>14: val_class = np.argmax(val,axis=1)<br>15: cm = confusion_matrix(val_label_array, val_class)<br>16: plt.show() |
| |

| **Algorithm 3: LSTM Layer 3** | |
|---|---|
| Input: x_train.shape[2] , batch_size = 32/64<br>Initialization: Define Sequential model : model = Sequential()<br>1:model.add(LSTM(100/50,input_dim=(seq_array.shape[1],seq_array.shape[2]), return_sequences=True))<br>2: model.add(LSTM(100/50, return_sequences=True))<br>3: model.add(LSTM(100/50, return_sequences=False))<br>3:model.add(Dense(units=64/32,activation=[‘relu’,’tanh’,’Linear’]))<br>4:model.add(Dense(units=32/16,activation=[‘relu’,’tanh’,’Linear’]))<br>5:model.add(Dense(units=16/8,activation=[‘relu’,’tanh’,’Linear’]))<br>6: model.add(Dense(units=[4,5], activation=‘softmax’))<br>7:model.compile(loss=‘sparse_categorical_crossentropy’, optimizer=‘adam’, metrics=[“accuracy”])<br>8: history= model.fit(seq_array, label_array)<br>9: epochs=100, validation_data=(val_seq_array, val_label_array) | |

| | |
|---|---|
| 10:Callbacks=[EarlyStopping(min_delta=0,patience=3,verbose=0,mode='auto'), ReduceLRonPlateau(monitor='loss',min_lr=0.000001)]<br>11: training_loss = history.history['loss']<br>12: test_loss = history.history['val_loss']<br>13: val = model.predict(val_seq_array)<br>14: val_class = np.argmax(val,axis=1)<br>15: cm = confusion_matrix(val_label_array, val_class)<br>16: plt.show() | |

**4. Results and Discussions**

We have performed a series of iteration with Layer-1 LSTM, Layer-2 LSTM and finally with Layer-3 LSTM for Binary, Multiclass grouped and Multiclass ungrouped respectively. The input, output and analysis for each layer have been described below in table 2 with following parameters-:

Activation Function = ReLU Rectified Linear Unit
Learning rate= 0.0000001
Epochs= 20 with a callback function.
Adam Optimizer
ReduceLROnPlateau
Patience = 3
Verbose = 0
Mode = 'auto'

Table 2 Performance Parameters of Binary and Multiclass Data using LSTM Model

| | Binary Classification | Multiclass Grouped Classification | Multiclass UnGrouped Classification |
|---|---|---|---|
| **Input** | Dense Units = 32,16,8<br>LSTM units = 50<br>Batch size = 128 | Dense Units = 32,16,8<br>LSTM units = 50<br>Batch size = 64 | Dense units = 32,16,8<br>LSTM units = 50<br>Batch size = 64 |
| **Output** | Precision = 0.980<br>Recall = 0.955<br>F-1 Score = 0.970 | Precision = 0.9875<br>Recall = 0.9750<br>F-1 Score = 0.9800 | Precision = 0.9785<br>Recall = 0.9442<br>F-1 Score = 0.9585 |
| **Analysis** | The maximum value is obtained with layer-1 LSTM | The maximum results have been obtained using layer-2 LSTM | The maximum results of ungrouped classification are obtained with layer-1 LSTM. |

ISSN: 2094-0343
2326-9865

Layer-2 LSTM Multiclass grouped classification yield maximum values of Precision, Recall and F-1 Score as 98.75%, 97.5% and 98% respectively.

Thereafter, comparison between different activation functions for binary, multiclass grouped and multiclass ungrouped classification have been described respectively in tables 3-5.

Table 3 Performance Parameters of binary classification with different Activation Functions

| Binary/Multiclass | Activation Function | LSTM Units | Dense Units | Batch Size (64/128) | Precision | | | Recall | | | F1 Score | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | Average Precision | 0 | 1 | Average Recall | 0 | 1 | Average F1 Score |
| Binary | Relu | 50 | 32,16,8 | 128 | 0.97 | 0.99 | 0.98 | 0.91 | 1 | 0.955 | 0.94 | 1 | 0.97 |
| Binary | Linear | 50 | 32,16,9 | 128 | 0.97 | 0.99 | 0.98 | 0.89 | 1 | 0.945 | 0.93 | 1 | 0.965 |
| Binary | Sigmoid | 50 | 32,16,10 | 128 | 0.92 | 0.99 | 0.955 | 0.9 | 0.99 | 0.945 | 0.91 | 0.99 | 0.95 |
| Binary | Tanh | 50 | 32,16,11 | 128 | 0.96 | 0.99 | 0.975 | 0.88 | 1 | 0.94 | 0.92 | 0.99 | 0.955 |
| Binary | LeakyRelu | 50 | 32,16,12 | 128 | 0.92 | 0.99 | 0.955 | 0.9 | 0.99 | 0.945 | 0.91 | 0.99 | 0.95 |

Table 4 Performance Parameters of Grouped Multiclass classification with different Activation Functions

| Binary/Multiclass | Activation Function | LSTM Units | Dense Units | Batch Size(64/128) | Precision | | | | | Recall | | | | | F1 Score | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | Average | 0 | 1 | 2 | 3 | Average | 0 | 1 | 2 | 3 | Average |

| | | | | | | | | | Precision | | | | | Recall | | | | | F1 Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Multi class Grouped | Relu | 50 | 32, 16, 8 | 64 | 0.99 | 0.99 | 0.98 | 0.99 | 0.9875 | 1 | 0.92 | 0.99 | 0.99 | 0.975 | 0.99 | 0.95 | 0.99 | 0.99 | 0.98 |
| Multi class Grouped | Linear | 50 | 32, 16, 8 | 64 | 0.99 | 0.94 | 0.99 | 0.99 | 0.9775 | 1 | 0.91 | 0.99 | 0.99 | 0.9725 | 1 | 0.92 | 0.99 | 0.99 | 0.975 |
| Multi class Grouped | Sigmoid | 50 | 32, 16, 9 | 64 | 0.99 | 0.96 | 0.99 | 0.98 | 0.98 | 1 | 0.91 | 0.99 | 0.99 | 0.9725 | 0.99 | 0.93 | 0.99 | 0.99 | 0.975 |
| Multi class Grouped | Tanh | 50 | 32, 16, 10 | 64 | 0.99 | 0.95 | 0.99 | 0.99 | 0.98 | 1 | 0.91 | 0.99 | 0.99 | 0.9725 | 0.99 | 0.93 | 0.99 | 0.99 | 0.975 |
| Multi class Grouped | LeakyRelu | 50 | 32, 16, 11 | 64 | 0.99 | 0.95 | 0.98 | 0.99 | 0.9775 | 1 | 0.9 | 0.99 | 0.99 | 0.97 | 0.99 | 0.92 | 0.99 | 0.99 | 0.9725 |

Table 5(a) Precision of UnGrouped Multiclass classification with different Activation Functions

| Mulicalss Ungrouped | Activation Function | LSTM Units | Dense Units | Batch Size(64/128) | Precision | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | Avg Preciso n |
| Mulicalss Ungrouped | Relu | 50 | 32,16,8 | 64 | 0.98 | 0.99 | 0.98 | 0.94 | 0.99 | 0.98 | 1 | 0.98 |
| Mulicalss Ungrouped | Linear | 50 | 32,16,8 | 64 | 0.98 | 0.97 | 0.97 | 0.92 | 0.99 | 0.98 | 0.99 | 0.9718875 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Muliclass Ungrouped | Sigmoid | 50 | 32,16,9 | 64 | 0.98 | 0.95 | 0.95 | 0.93 | 0.98 | 0.98 | 0.99 | 0.967725 |
| Muliclass Ungrouped | tanh | 50 | 32,16,10 | 64 | 0.98 | 0.99 | 0.97 | 0.94 | 0.99 | 0.98 | 0.98 | 0.9756625 |
| Muliclass Ungrouped | Leaky Relu | 50 | 32,16,11 | 64 | 0.98 | 0.99 | 0.96 | 0.91 | 0.99 | 0.98 | 1 | 0.9745125 |

Table 5(b) Recall of UnGrouped Multiclass classification with different Activation Functions

| Muliclass Ungrouped | Activation Function | LSTM Units | Dense Units | Batch Size(64/128) | Recall | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | Avg Recall |
| Muliclass Ungrouped | Relu | 50 | 32,16,8 | 64 | 0.99 | 0.72 | 1 | 0.92 | 0.99 | 1 | 0.99 | 0.944285714 |
| Muliclass Ungrouped | Linear | 50 | 32,16,8 | 64 | 0.99 | 0.72 | 1 | 0.92 | 0.99 | 0.99 | 0.98 | 0.941428571 |
| Muliclass Ungrouped | Sigmoid | 50 | 32,16,9 | 64 | 0.99 | 0.5 | 1 | 0.91 | 0.98 | 0.99 | 0.99 | 0.908571429 |
| Muliclass Ungrouped | tanh | 50 | 32,16,10 | 64 | 0.99 | 0.73 | 1 | 0.91 | 0.99 | 0.99 | 0.99 | 0.942857143 |
| Muliclass Ungrouped | Leaky Relu | 50 | 32,16,11 | 64 | 0.99 | 0.69 | 1 | 0.92 | 0.99 | 0.99 | 0.99 | 0.938571429 |

Table 5(c) F1-Score of UnGrouped Multiclass classification with different Activation Functions

| Mulicl ass Ungrou ped | Activat ion Functi on | LSTM Units | Dense Units | Batch Size(64/ 128) | F1 Score | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | Avg F1 Score |
| Mulicl ass Ungrou ped | Relu | 50 | 32,16 ,8 | 64 | 0. 99 | 0. 85 | 0. 98 | 0. 92 | 0. 99 | 0. 99 | 0. 99 | 0.95857 1429 |
| Mulicl ass Ungrou ped | Linear | 50 | 32,16 ,8 | 64 | 0. 99 | 0. 83 | 0. 98 | 0. 92 | 0. 99 | 0. 98 | 0. 99 | 0.95428 5714 |
| Mulicl ass Ungrou ped | Sigmoi d | 50 | 32,16 ,9 | 64 | 0. 98 | 0. 65 | 0. 97 | 0. 92 | 0. 98 | 0. 99 | 0. 99 | 0.92571 4286 |
| Mulicl ass Ungrou ped | tanh | 50 | 32,16 ,10 | 64 | 0. 99 | 0. 84 | 0. 98 | 0. 93 | 0. 99 | 0. 99 | 0. 99 | 0.95357 1429 |
| Mulicl ass Ungrou ped | Leaky Relu | 50 | 32,16 ,11 | 64 | 0. 99 | 0. 81 | 0. 98 | 0. 91 | 0. 99 | 0. 98 | 0. 99 | 0.95 |

It has been concluded from the tables that Relu performed best amongst all the activation functions as it help in faster learning. Hence, multiclass grouped classification with LSTM Layer-2 performs the best.

We compared our model with several state-of-the-art methods (Table 6) on the CICDDoS2019 dataset and found that our model performs best. Experimental results show that this paper has higher performance parameters.

Table 6 Comparison with other state-of-the-art methods on CICDDoS2019 Evaluation Dataset

| **Study** | **Year** | **Feature Selection** | **Machine Learning Algorithms** | **Classification** | **Performance Parameters (Accuracy)** |
|---|---|---|---|---|---|
| Gaur et al. [13] | 2021 | Chi-Square, Extra Tree, ANOVA | Random Forest, DT, KNN, XGBoost | Binary | XGBoost + ANOVA Accuracy = 98.34% |
| Abbas et al. [15] | 2021 | No | Random Forest | Binary | Random Forest = 99.976% Partial dataset (PORTMAP, LDAP) |
| Alamri et al. [16] | 2021 | No | LR, RF and XGBoost | Binary and Multiclass | XGBoost = 99.7% (Binary) XGBoost = 91.3% (Multiclass) |
| Rahman et al. [19] | 2020 | No | LR, DT, SVM | Binary | SVM = 97.1% |
| Chesney et al. [20] | 2021 | No | LR | Binary | LR = 99.70% Partial Dataset (only LDAP file) |
| Ferrag et al. [21] | 2021 | No | CNN | Binary and Multiclass | CNN = 99%(Binary) CNN = 90%(Multiclass) |
| Sanchez et al. [22] | 2021 | No | RF | Binary | RF = 99% |
| Elsayed et al. [23] | 2020 | No | RNN with Autoencoder | Binary | Proposed Model = 99% |
| Manikumar et al. [25] | 2020 | Extra Tree-Based Classifier | KNN, DT and RF | Binary | RF = 95.19% (without feature selection) RF = 96.74% (with extra tree) |

## Conclusion

Detection of DDoS attacks is very essential to protect networks. However, due to the lack of availability of intrusion detection systems and real-time data, there are significant hindrances to the detection of DDoS attacks. This paper proposes M-LSTM model to early detect the

DDoS attacks (Binary and Multiclass Classification). M-LSTM model starts with LSTM model on binary classification. Later, multiclass data is checked for grouped and ungrouped data. We also investigated the contributions of M-LSTM model on Precision, Recall and F-1 Score in multiclass classification of data. Experimental results show that Layer-2 LSTM Multiclass grouped classification yield maximum values of Precision, Recall and F-1 Score as 98.75%, 97.5% and 98% respectively.

**References**

1. M. Simon, L. Huraj, and M. Cernansky, "Performance Evaluations of IPTables Firewall Solutions Under DDoS Attacks," Journal of Applied Mathematics, Statistics and Informatics, vol. 11, no. 2, pp. 35–45, 2015.
2. P. Shamsolmoali and M. Zareapoor, "Statistical-Based Filtering System Against DDoS Attacks in Cloud Computing," in 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1234–1239, IEEE, 2014.
3. K. Kalkan and F. Alag¨oz, "A Distributed Filtering Mechanism Against DDoS Attacks: ScoreForCore," Computer Networks, vol. 108, pp. 199–209, 2016.
4. K. Singh, P. Singh, and K. Kumar, "A Systematic Review of IP Traceback Schemes for Denial of Service Attacks," Computers & Security,vol. 56, pp. 111–139, 2016
5. Varun, B. N. ., S. . Vasavi, and S. . Basu. "Python Implementation of Intelligent System for Quality Control of Argo Floats Using Alpha Convex Hull". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 5, May 2022, pp. 60-64, doi:10.17762/ijritcc.v10i5.5554.
6. Anuradha and A. Singhrova, "A host based intrusion detection system for DDoS attack in WLAN," 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), 2011, pp. 433-438, doi: 10.1109/ICCCT.2011.6075142.
7. A. Navaz, V. Sangeetha, and C. Prabhadevi, "Entropy Based Anomaly Detection System to Prevent DDoS Attacks in Cloud," International Journal of Computer Applications, vol. 65, pp. 42–47, Jan. 2013.
8. J. Li, M. Liu, Z. Xue, X. Fan, and X. He, "RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things," IEEE Access, vol. 8, pp. 36191–36201, 2020.
9. M. E. Ahmed and H. Kim, "DDoS Attack Mitigation in Internet of Things Using Software Defined Networking," in 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), pp. 271–276, IEEE, 2017.
10. E S Alghoson, O Abbass. Detecting Distributed Denial of Service Attacks using Machine Learning Models. Int. J. Adv. Comput. 2021, 12,12.
11. G S Kushwah, V Ranga. Distributed denial of service attack detection in cloud computing using hybrid extreme learning machine. Turk. J. Electr. Eng. Comput. Sci. 2021, https://doi:10.3906/elk-1908-87
12. A Chartuni and J Márquez. Multi-Classifier of DDoS Attacks in Computer Networks Built on Neural Networks" Appl. Sci. 2021.11,10609. https://doi.org/10.3390/app112210609
13. DC Can, Le, HQ., Ha, QT. Detection of Distributed Denial of Service Attacks Using Automatic Feature Selection with Enhancement for Imbalance Dataset" In: Nguyen, N.T., Chittayasothorn, S., Niyato, D., Trawiński, B. (eds) Intelligent Information and Database

Systems. ACIIDS 2021. Lecture Notes in Computer Science (), 12672. Springer, Cham. https://doi.org/10.1007/978-3-030-73280-6_31
14. Chaudhary, D. S. . (2022). Analysis of Concept of Big Data Process, Strategies, Adoption and Implementation. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(1), 05–08. https://doi.org/10.17762/ijfrcsce.v8i1.2065
15. V Gaur, R Kumar. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. Arab J Sci Eng. 2021. https://doi.org/10.1007/s13369-021-05947-3
16. V Odumuyiwa, R Alabi. DDoS Detection on Internet of Things Using Unsupervised Algorithms. J. Cyber Secur. Mobil.10, 3. URL 10.13052/jcsm2245-1439.1034
17. I S A Abbas, S Almhanna, S. Distributed Denial of Service Attacks Detection System by Machine Learning Based on Dimensionality Reduction. In: International Conference of Modern Applications on Information and Communication Technology (ICMAICT) 22-23 October 2020, University of Babylon, Babylon-Hilla City, Iraq: IEEE
18. H A Alamri, V Thayananthan, J Yazdani, J. Machine Learning for Securing SDN based 5G network. Int. J. Comput. Appl., 174, 14. URL 10.5120/ijca2021921027
19. D Parfenov, L Zabrodina, A Zhigalov, I Bolodurina, Research of multiclass fuzzy classification of traffic for attacks identification in the networks. J. Phys. Conf. Ser. 1-7.2020. URL 10.1088/1742-6596/1679/4/042023
20. M Shurman, R Khrais, A Yateem. DoS and DDoS attack detection using deep learning and IDS. Int. Arab J. Inf. Technol. **17**(4A), p.655–661.2020. https:// doi.org/ 10. 34028/ iajit/ 17/ 4A/ 10
21. M. Dursun and N. Goker, “Evaluation of Project Management Methodologies Success Factors Using Fuzzy Cognitive Map Method: Waterfall, Agile, And Lean Six Sigma Cases”, Int J Intell Syst Appl Eng, vol. 10, no. 1, pp. 35–43, Mar. 2022.
22. M A Rahman. Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms Int. J. Smart Home. 14, 2, p.15-24.2021. URL 10.21742/IJSH.2020.14.2.02
23. S Chesney, K Roy, S Khorsandroo. Machine Learning Algorithms for Preventing IoT Cybersecurity Attacks. In: Arai K., Kapoor S., Bhatia R. Intelligent Systems and Applications. IntelliSys. 2020. Advances in Intelligent Systems and Computing, vol 1252. Springer (Book)
24. M S Ferrag, L Shu, H Djallel, K K R Choo. Deep Learning-based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0. Electronics. 2021. 10, 11. URL 10.3390/electronics10111257.
25. O R Sanchez, M Repetto, A Carrega, R Bolla, R. Evaluating ML-based DDoS Detection with Grid Search Hyperparameter Optimization. In: 2021 7th International Conference on Network Softwarization (NetSoft), pp. 402-408, 28 June-2 July 2021, Tokyo, Japan: IEEE.
26. M S Elsayed, N A L Khac, S Dev, A D Jurcut. DDoSNet: A Deep-Learning Model for detecting network attacks. In: 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 31 August-03 September 2020, pp.391-396. Cork, Ireland: IEEE.
27. M V O Assis, L F Carvalho, J Lloret, Jr M L Proença. A GRU deep learning system against attacks in software defined networks. J. Netw. Comput. Appl. 177. URL 10.1016/j.jnca.2020.102942.

28. D V V S Manikumar, B U Maheswari. Blockchain Based DDoS Mitigation Using Machine Learning Techniques. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 794-800, 15-17 July 2020Coimbatore, India:IEEE.
29. J Li, M Liu, Z Xue, X. Fan, X. He. Rtvd: A real-time volumetric detection scheme for DDoS in the internet of things. IEEE Access, 8, p. 36191-36201. 2020. URL 10.1109/ACCESS.2020.2974293.
30. Y Jia, F Zhong, A Alrawais, B Gong, X Cheng. Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet Things J. 7, 10, 9552-9562. URL 10.1109/ACCESS.2020.2974293.
31. I Sharafaldin, A H Lashkari. S Hakak and A A Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, pp. 1-8, 1-3 October2019, Chennai, India: IEEE.
32. TH Vuong, C V N Thi, QT. Ha (2021) N-Tier Machine Learning-Based Architecture for DDoS Attack Detection. In: Nguyen N.T., Chittayasothorn S., Niyato D., Trawiński B. (eds) Intelligent Information and Database Systems. ACIIDS 2021. Lecture Notes in Computer Science, vol 12672. Springer, Cham. https://doi.org/10.1007/978-3-030-73280-6_30.
33. V. Gaur and R. Kumar, "HCTDDA: Hybrid Classification Technique for Detection of DDoS Attacks," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), 2021, pp. 1-5, doi: 10.1109/ISCON52037.2021.9702399.
34. V. Gaur and R. Kumar, "DDoSLSTM: Detection of Distributed Denial of Service Attacks on IoT Devices using LSTM Model," 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2022, pp. 01-07, doi: 10.1109/IC3IOT53935.2022.9767889.
35. V. Gaur and R. Kumar, "FSMDAD: Feature Selection Method for DDoS Attack Detection," 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 939-944, doi: 10.1109/ICEARS53579.2022.9752308.
36. P. Modiya and S. Vahora, “Brain Tumor Detection Using Transfer Learning with Dimensionality Reduction Method”, Int J Intell Syst Appl Eng, vol. 10, no. 2, pp. 201–206, May 2022.
37. V. Gaur and R. Kumar, "ET-RF based Model for Detection of Distributed Denial of Service Attacks," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022, pp. 1205-1212, doi: 10.1109/ICSCDS53736.2022.9760938.