International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimizing Network Security using LSTM Algorithm for Traffic Classification on UNSWNB15 and KDDCUP99 Dataset

Tarun Sharma *[1], Dr. Rais Abdul Hamid Khan [2]

Submitted: 07/10/2023 **Revised**: 26/11/2023 **Accepted**: 08/12/2023

Abstract: Various systems depend on the capability to classify network traffic for tasks such as detecting intrusions, enforcing policies, and managing traffic. Machine Learning (ML) and especially Deep Learning (DL) based classifiers have shown exceptional accuracy in classifying network traffic, despite the fact that many applications encrypt their network data and others change their port numbers constantly. This paper proposes a classification approach using graph convolution and Long-Short Term Memory (LSTM) to handle coupled network data flows. In order to analyse the spatial topological and temporal properties of the LSTM, the traffic flow data has to be preprocessed. Ultimately, the approach is evaluated on a portion of the UNSWNB15 and KDDCUP99 datasets to quantify its effectiveness. The proposed methodology has shown its ability to effectively derive potential attributes from network traffic data via successful experimentation. The recommended strategy is shown to be effective and performs better than other methods such as feature selection, bidirectional LSTM (BiDLSTM), and CNN-LSTM in terms of classification performance.

Keywords: *Machine Learning, Deep Learning, Long-Short Term Memory, bidirectional LSTM, UNSWNB15, KDDCUP99.*

1. Introduction

The classification of network traffic is necessary for a wide variety of applications, including administration and security systems. For example, it enables network managers to prioritise certain kinds of traffic and to use suitable procedures in order to customise quality of service and security regulations to the needs of each application [1]. Since a long time ago, both the business world and the academic community have shown a significant interest in the subject matter. New applications and encryption protocols have evolved, bringing with them new challenges [2]. This is happening at the same time as the Internet and mobile technologies are continuing to advance.

Deep learning approaches have a tendency to perform better than feature selection methods when it comes to the extraction of discriminative features from network traffic [3]. As a result of the adaptability of deep learning approaches, it is possible to extract features from network data without having to resort to a long sequence of challenging operations such as feature engineering. It is possible to extract local characteristics of network traffic by using filters and standard extraction techniques such as convolutional neural networks (CNNs) [4, 5]. The technique necessitates the transformation of the data on network traffic from its initial, one-dimensional format into a space that is two-dimensional [6].

[1]*Research Scholar, Department of Computer Science & Engineering*
[2]*Research Supervisor, Department of Computer Science & Engineering*
[1,2] *Dr. A.P.J Abdul Kalam University, Indore, MP, India, 452016*
* *Corresponding Author : Tarun Sharma*
Email: Sharma_tarun_in@yahoo.com

For the purpose of providing a method for classifying network traffic, this research makes use of convolution of graphs that include both long-term and short-term memory functions. This strategy makes use of the LSTM technique to extract the temporal aspects of information on network traffic, and it makes use of the strong topology extraction capacity of the graph convolution method to recover the spatial aspects of the information under consideration. In particular, the following are the primary methods in which this work presents its contribution:

By combining graph convolution with long short-term memory (LSTM), we propose a method for classifying network traffic that improves accuracy, abnormal traffic detection, and false alarm rates for regular traffic.

We compute metrics for each category, then evaluate them against deep learning models' feature selection techniques and other models to determine how well the proposed model classifies network traffic models are explained in the result section. The UNSW-NB15 is used as a reference data set for the assessment procedure.

The remaining parts of the paper are structured as follows. The second part of the paper examines the literature on the topic. The suggested traffic categorization model's comprehensive building procedure is presented in Section III. Evaluation and comparison of experimental results and productivity are reported in Sections IV and V. The last section of the paper provides a summary and forecast.

2. Literature Work

The machine learning (ML) strategies for traffic categorization were dissected in detail. The author provides the traffic obfuscation methods that may aid in the development of a more accurate classifier for the convenience of researchers. Key discoveries and open research issues for network traffic categorization are addressed, and suggestions for future research areas are provided. Overall, this study is a necessary addition to the literature since it compiles the most recent findings from studies on traffic categorization [7] and addresses gaps in the coverage of earlier studies.

We investigate the challenge of classifying nano-network traffic collected at the micro/nano-gateway and apply five supervised machine learning methods. Experimentally comparing and contrasting the presented models reveals the best classifier for nano-network traffic, with high accuracy and performance scores [8].

Three techniques are the main focus of this study: The procedure began with creating a picture representation of the road sound dataset sequences. Then, a model for feature extraction was suggested using a convolutional neural network. Lastly, a hybrid approach was used for classification, combining a convolutional neural network with other machine learning models. We have collected data on traffic noises from an uneven urban road at different times of the day (including morning and evening) to see whether our theories hold water. Specifically, the implementations have shown promising outcomes, with accuracies varying between 92% and 95% when determining traffic volumes over time [9].

To detect and categorize diverse traffic flows in 5G network slicing, a framework based on the multi-lane Capsule Networks (CapsNet) deep learning approach has been developed. In addition, the author employs deep learning methods to compare the model with the literature mentioned above. Compared to other classifiers in the literature [10], the experimental findings show substantial performance improvement, with an accuracy of 97.3975%.

In this research, we create two unique traffic categorization methods to help with this problem. The first implements the Random Forest technique on the plaintext bytes of TLS Hello messages. It is quite easy to implement and works well for categorizing traffic based on throughput. Additionally, the classification quality is improved while processing times are reduced by a factor of three compared to state-of-the-art techniques. The second method enhances the first by paying special attention to the handshake's information. As a result, it can rapidly extract information from the transaction and obtain the best possible categorization quality across the board. In addition to accurately classifying ECH traffic [11], its error rate is three times lower than that of state-of-the-art algorithms.

A novel approach to decrypting network data and detecting encrypted, tunnelled, and anonymous communication is presented in this paper. The proposed identification method decrypts encrypted traffic flows to identify anonymous network traffic and separate VoIP and non-VoIP calls using the much-desired deep learning techniques. Three categories have been established from the gathered data: VPN Voice over IP (VoIP), VPN Data Only (Data Only), and TOR Voice over IP (VoIP). After exhaustive testing, we found that our identification engine can withstand VPN and TOR connection disturbances [12].

Based on simulation results, the enhanced Harris Eagle algorithm, when combined with fuzzy clustering, achieves better intra-class compactness and inter-class separation on the data traffic sample set than the grey wolf algorithm, the conventional fuzzy clustering method, and the particle swarm algorithm-based clustering method. Consequently, clustering's accuracy and recall are enhanced to about 90% [13].

In trials, the proposed method was able to effectively extract potential attributes from data on network traffic. It proves that the suggested method works and beats other classification performance strategies, such as CNN-LSTM, Bidirectional LSTM (BiDLSTM), and feature selection [14].

Using a CNN classifier yields the best results in terms of classification accuracy. The CNN classifier is improved with the ability to reject sessions whose patterns do not match those learned during training in order to do a fine-grained, live, unsupervised traffic deconstruction for the four radio cells that are being observed [15].

Reviewing the existing research in the topic, this article presents an overview of AL and places it within the framework of NTC. Furthermore, difficulties and unanswered questions about categorizing network traffic using AL are highlighted. In addition, several experiments are carried out as a means of providing a technical overview, demonstrating the extensive potential of AL in NTC. Simulation results demonstrate that accuracy may be achieved using less data when using AL [16].

Since our method generates a unique cost matrix for each division, the costs associated with each category of misclassification are distinct. The author applies the suggested cost-sensitive learning approach to stacked autoencoder and convolution neural networks, two deep learning classifiers, to evaluate its usefulness. In our trials on the ISCX VPN-nonVPN dataset, we found that the suggested methodology outperformed three state-of-the-art NTC algorithms [17] regarding classification performance for low-frequency classes.

Important characteristics are initially extracted from network traces via processing. After reviewing previous survey studies, the author settled on cutting-edge machine-learning techniques to categorize IoT traffic. The author then compared the results of several machine learning methods regarding classification precision, speed, training duration, etc. Finally, the author recommended an appropriate machine-learning algorithm for various applications [18].

Learned side-channel features from header segments in particular improve performance. The last step is to find the label for the packet using the softmax algorithm. Another benefit of EBSNN is that it can classify network traffic patterns only by looking at the first packets. When compared to state-of-the-art algorithms, EBSNN performs better on application identification and website identification tasks, according to extensive testing conducted on real-world datasets [19].

Our traffic classification approach achieves a flow classification accuracy of up to 97.7 percent utilising only 9 initial packets of flows, which is much better than state-of-the-art systems based on machine learning. The authors demonstrate that a flow classification accuracy rate of 96.6% may be achieved with as little as 0.5% of all flows used for GMM training. Our method, which only requires the first six packets in a flow, achieves a Half Total Error Rate (HTER) of 7.65 or lower [20].

For the purpose of evaluating the suggested model's efficacy, confusion matrices, multiple classification metrics, and reciprocal operating characteristic (ROC) curves are used to assess performance. Although the suggested model is lower in size, experimental results show that it outperforms well-known models for anomalous traffic identification in terms of classification impact [21].

3. Proposed Work

First, let's talk about the GC MODEL. It has been shown that Graph Convolutional Networks (GCNs) are useful for learning graph representation [22],[23] because of their ability to excerpt longitudinal aspects of topological arrangements. GC (Simple_Graph_Convolutional) [24] is an optimization that builds on GCN that does away with The Effect of Nonlinearity on GCN and drastically cuts down the amount of time it takes to run the model by doing the calculations ahead of time.

Model LSTM is a specialized thoughtful of RNN that is often used to address the RNN dependence issue over the long term [25], [26]. Through a more intricate hidden layer unit structure, LSTM can circumvent the gradient disappearance issue. In Fig.1, we see the LSTM's fundamental building block.

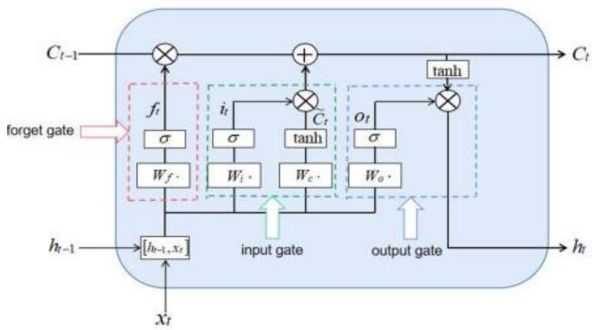


Fig 1. The framework of the LSTM unit.

Differentiating features of the LSTM model include image-recording devices such as the forget gate, input gate, and output gate. These three gate structures provide the following purposes:

The first kind of gate is called a forget gate, which determines the probability with which the current LSTM unit forgets the state of the higher hidden unit.

Second, the sequence's input is processed by the input gate.

Finally, the third gate is the output gate, which reveals the concealed state ht at time t.

C. Using a Generalized Classifier of Graph Convolution (GC) and a Short-Term Long Memory (LSTM) to Design, a Classification Model for Network Traffic

Figure 2 shows the layers of the GC-LSTM model that has been suggested. An output layer, a fully connected layer, a GC graph convolutional layer, and an LSTM layer make up these layers. The raw data is processed before using traffic flow correlations to construct the topological graph. Following data cleaning and preparation, the GC model is used to create a spatial representation. The LSTM layer then uses the output of GC to create a temporal representation. Following the LSTM layer, an output layer and a fully connected layer are affixed to the model during the subsequent training phase.

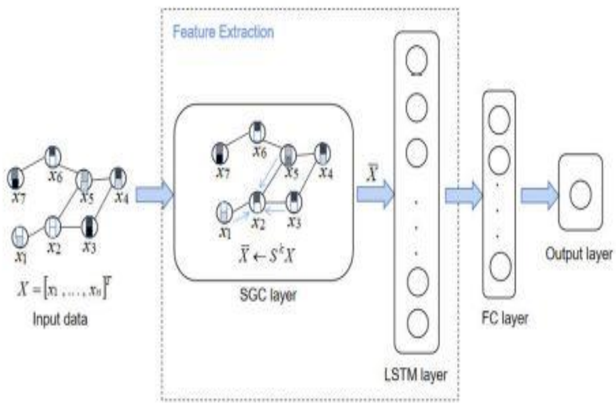


Fig 2. Framework for a GC-LSTM Model.

One Processing of Incoming Information Because various features employ different measurement techniques, it is important to normalize the data for numerical characteristics to remove the influence of measurement.

The GC-LSTM Feature Extraction Layer strongly reflects the GC layer, particularly the local smoothing of nodes and their neighbours.

Layer 3 and Training Process Interconnected Method 1 is a brief overview of the GC-LSTM model-based traffic categorization algorithm. The LSTM layer's output is sent into the fully connected layer, with 64 nodes and a similar number of connections. The major focus of this article is to determine whether aberrant traffic patterns have any discernible geographic patterns. To that end, we'll focus on a binary-classification experiment, where a sigmoid function is a viable option for the output layer's activation function.

Algorithm 1 : GC-LSTM Training

```
Input: Sampled UNSW-NB15 dataset, RMSProp, lr,
        batch_size, dropout , KDDCUP99
Output: GC-LSTM Model
load dataset
for data in training and test sets do
    Extract Features(X)
    Extract Labels(Y)
end
scale features with X̂_ij = (X_ij − MEAN_j) / STD_j
establish matrix A and D based on connection rules
calculate S based on Ã and D̃, initialize H = S
for i from 2 → k do
    H = HS
end
get the output X̄ = HX
input X̄ into the LSTM layer
add a fully connected layer, whose value is 32
add a dropout, whose value is 0.1
get cross-entropy loss by z_i and y_i
update parameters by RMSProp with loss
```

D. Analysis of the Model

Using benchmark datasets like UNSW-NB15 [27] and KDDCUP99 [28] has led to several important improvements in network security. However, recent studies have shown that these figures don't correctly reflect traffic or the incidence of risks like low-occupancy attacks in the actual world of networks. The India Cyber Security Centre gathers data sets more representative of the actual status of the Internet, such as UNSW-NB15 and KDDCUP99. This is why the UNSW-NB15 and KDDCUP99 datasets were used in the experiment reported here. Stratified sampling of 20% of the data from the UNSW-NB15 and KDDCUP99 datasets was used in the experiments; Later objective 20% of the data_set was sampled, and there is a decrease in the number of samples available in a particular attack class. As such, the binary-classification problem is a focal point of the experiment designed to confirm the hypothesis. Test and training set traffic flow distribution samples are shown in Table 1.

4. Implementation

4.1 The Laboratory Setup

The x360 Touchscreen 2-in-1 is a 14-inch full-screen IPS display with touch capabilities that can be folded into a tablet. This undertaking made use of Python on a laptop with a 512GB solid-state drive (SSD) and a 10th-generation Core i7-10510U processor. The operating system of option was Windows 10 Home 64 Bit. Varieties of processing units: Eight megabytes of L3 cache and a clock speed of 1.6 GHz make up the four CPU cores; Intel's Turbo Boost Technology allows the base frequency to go up to 4.9 GHz. Intel Iris Plus Graphics and HD Audio are features of this system. One such camera is HP's HD TrueVision. Some of the Python packages utilised in this technique include NumPy, Pandas, SciPy, PyTorch, Plotly, Keras, and OpenCV-python.

4.2 Dataset

Table 1: Twenty percent of the datasets were randomly picked for use as training and test data, respectively.

UNSW-NB15 [27]		
	Normal	Abnormal
Training set	1,05,204	70,136
Test Set	49,399	32,932

KDDCUP99 [28]		
	Normal	Abnormal
Training set	99,436	45,326
Test Set	34,256	26,354

Table 2. A Look at UNSW-NB15 vs. KDD CUP 99 [12]

Parameters	KDDCUP99 [28]	UNSW-NB15 [27]
No. of networks	2	3
No. of distinct ip address	11	45
Simulation	Yes	Yes
The duration of data collected	5 weeks	16 hours, 15 hours
Format of data collected	3 types (tcpdump, BSM and dump files)	Pcap files
Attack families	4	9
Feature Extraction tools	Bro-IDS tool	Argus, Bro-IDS and new tools
No. of features extraction	42	49

A comparison of the KDDCUP99 and UNSW-NB15 datasets is shown in Table 2 [32]. The table below shows eight differentiating factors for each data set: network count, IP address count, data type, data production time, output format, attack vectors, feature extraction techniques, and feature count. We can find many distinct families of assaults characteristic of contemporary low-footprint attacks in the UNSW-NB15 dataset.

4.3 Visualization

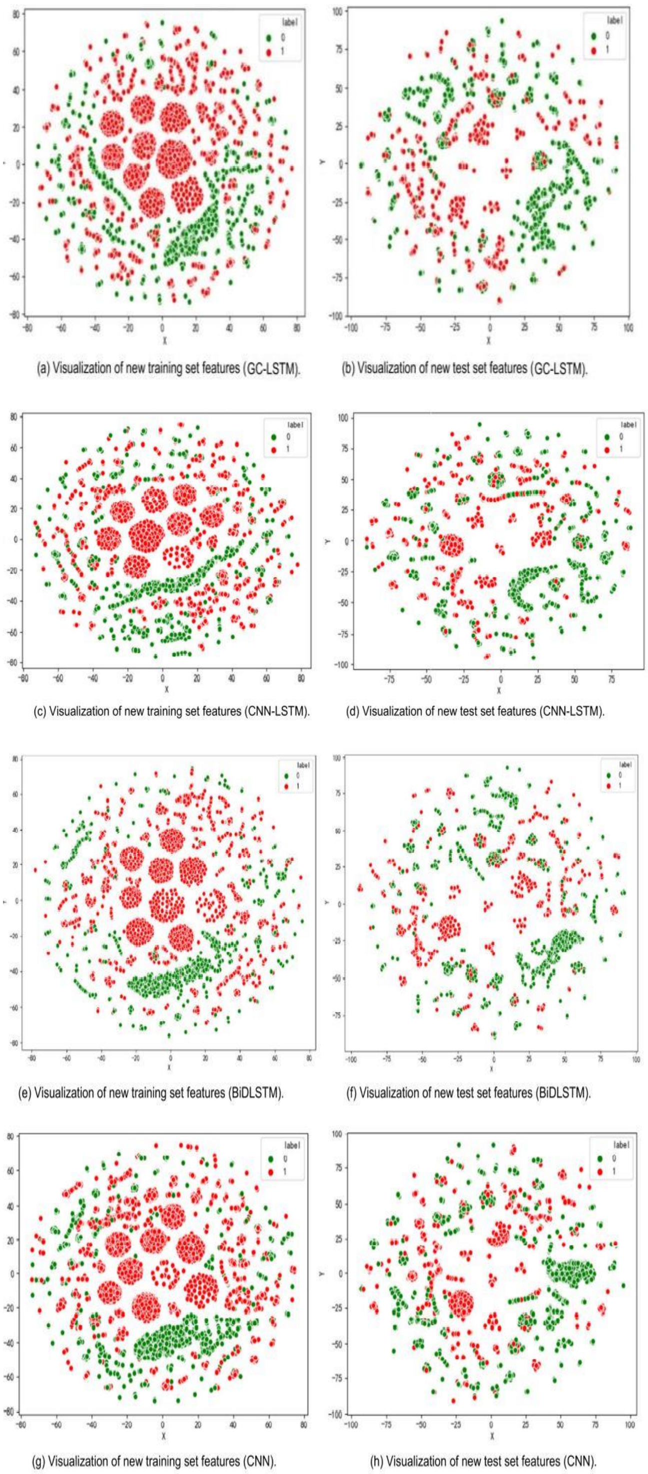


Fig 3. Displaying the features collected by each deep learning model during training and testing using t-SNE.

Each model extracted novel structures from the unique preparation and examination sets, and the outcomes of this process are shown in Figure 3 using the t-SNE approach. There is no obvious way to tell how well any of the four feature extraction models perform just by looking at the picture; instead, we will use several different categorization criteria to evaluate their effectiveness.

From the image above, it is clear that the GC-LSTM model performs worse than the CNN-LSTM model in terms of recall and accuracy for the abnormal class. Still, it outperforms the additional three representations in terms of additional metrics. The CNN with LSTM approach outperforms the CNN approach and the BiDLSTM approach across the board. Although there isn't much difference, the BIDLSTM model outperforms the CNN model overall.

5. Result

A confusion matrix is a common tool in classification issues [31] because it represents the proportion of data samples that were properly and wrongly labelled by the classifier. Think of the atypical group as a plus and the typical group as a minus. Then Table 3 displays the confusion matrix in its form:

Table 3. The mess of Confusion Matrix

		Predict	
		abnormal class (Positive)	normal class (Negative)
Actual	abnormal class (Positive)	**TP**	**FN**
	normal class (Negative)	**FP**	**TN**

The percentage of correctly labelled samples measures the accuracy of a prediction.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

In the equation below, r denotes the percentage of legitimately abnormal samples relative to total anomalous traffic records, which is a measure of abnormal class accuracy:

$$precision = \frac{TP}{TP + FP}$$

Measured as a percentage of all abnormal samples, recall of abnormal class indicates how many records were properly identified as abnormal. Another name for this is Detection Rate (DR), and it is calculated using the following formula:

$$DR = recall = \frac{TP}{TP + FN}$$

An all-encompassing measure of accuracy and recall, the f 1 score is written as follows:

$$f1 - score = \frac{2 * precision * recall}{precision + recall}$$

An equation describing the false alarm rate may be found below:

$$FAR = \frac{FP}{FP + TN}$$

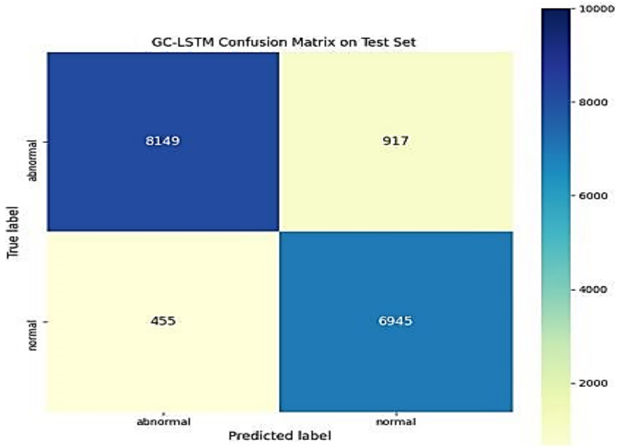


Fig 4. Xgboost's new test set data was collected using GC-LSTM, including its confusion matrix.

Figure 4 displays the metrics, and Table 5.4 displays the confusion matrix for the Xgboost model applied to the new test set retrieved using GC-LSTM.

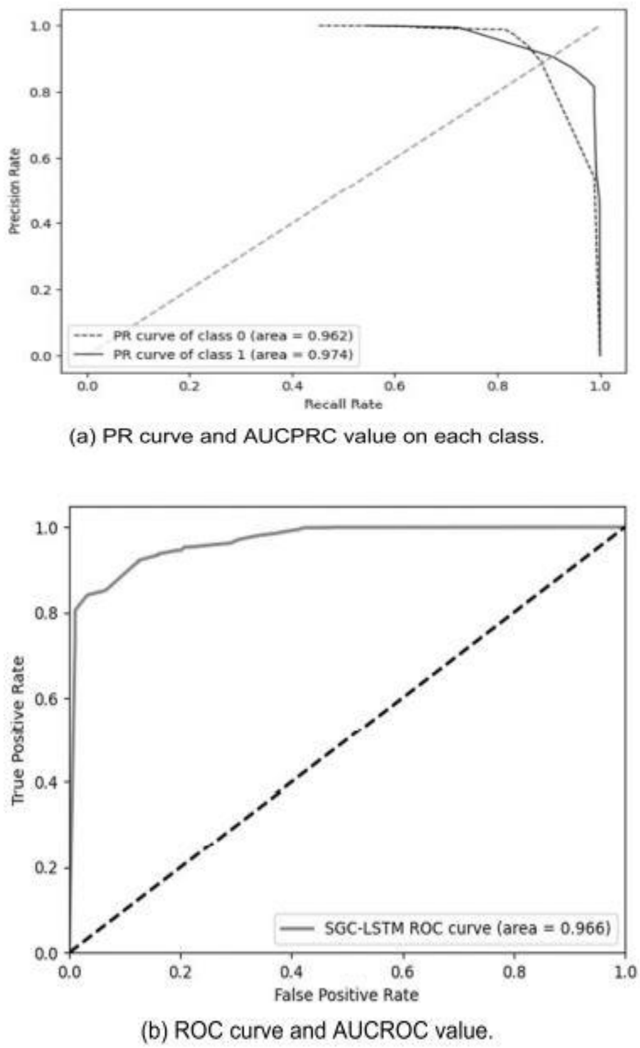


Fig 5. Xgboost's GC-LSTM-extracted evaluation curves for a fresh test dataset.

The ROC_curve of the Xgboost method on the novel examination established retrieved using GC_LSTM (a) and the PR_curve of each class (b) are shown above.

You can see that the model's AUCPRC score is more than 0.95 across the board in Fig.5 (a), albeit it excels at class 1 data (abnormal class). Later in this paper, we'll conduct experimental comparisons. The proposed approach is evaluated by first comparing the SC-LSTM model's retrieved features with those selected by the feature selection method. The outcomes are then contrasted with those produced using other deep learning methods.

Table 4: Measurements of the Xgboost method using the novel examination data gathered with the help of the proposed GC-LSTM.

UNSW-NB15 Dataset			
Class	Precision	Recall	F1-Score
normal(0)	0.91	0.96	0.94
abnormal(1)	0.98	0.92	0.93
macro avg	0.95	0.95	0.95
weighted avg	0.95	0.95	0.95

KDDCUP99 Dataset			
Class	Precision	Recall	F1-Score
normal(0)	0.89	0.96	0.92
abnormal(1)	0.96	0.91	0.9
macro avg	0.93	0.93	0.93
weighted avg	0.93	0.93	0.93

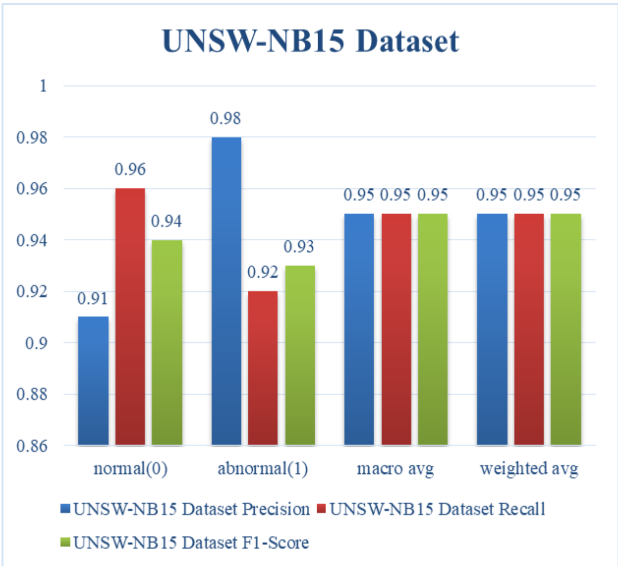


Fig 6. Measurements of the Xgboost method using the novel examination data gathered with the help of the proposed GC-LSTM in UNSW-NB15 Dataset

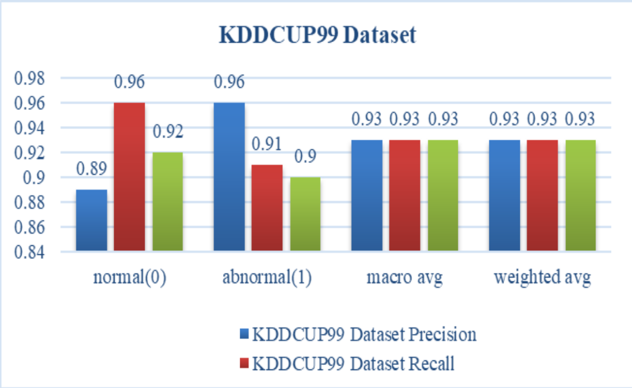


Fig 7. Measurements of the Xgboost method using the novel examination data gathered with the help of the proposed GC-LSTM in KDDCUP99 Dataset.

Table 4 and figure 6 , 7 shows that there is minimal difference between both the normal class and the abnormal class present in the sample set following GC-LSTM feature extraction.

Table 5: Performance Evaluation of the Proposed GC-LSTM and Feature_Selection Approach using the below Metrics.

UNSW-NB15 Dataset				**KDDCUP99 Dataset**			
Methods	**AUCPRC_0**	**AUCPRC_1**	**AUCROC**	**Methods**	**AUCPRC_0**	**AUCPRC_1**	**AUCROC**
Proposed GC-LSTM	0.984	0.995	0.989	**Proposed GC-LSTM**	0.961	0.974	0.963
Feature Selection [150]	0.922	0.959	0.957	**Feature Selection [150]**	0.893	0.943	0.948

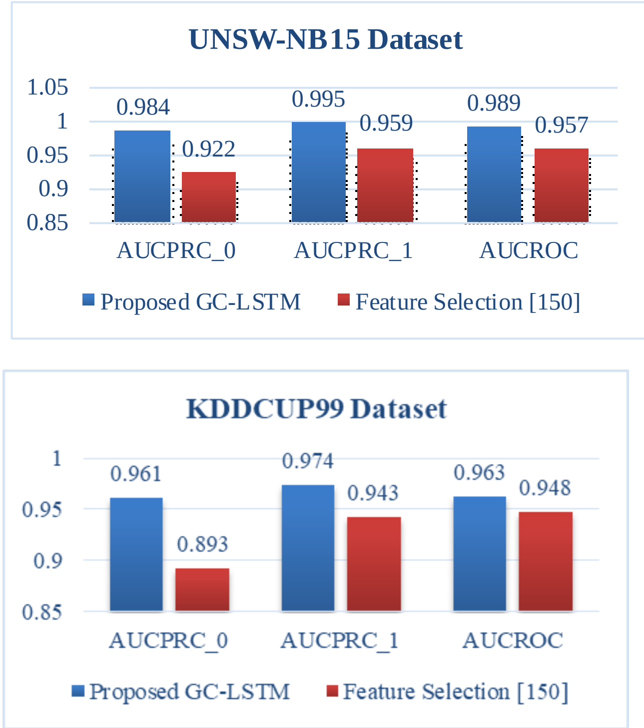


Fig 8: Performance Evaluation of the Proposed GC-LSTM and Feature_Selection Approach using the below Metrics in UNSW-NB15 Dataset and KDDCUP99 Dataset.

Here, the xgboost model is used to assess the newly recovered features via the usage of GC-LSTM and the feature subset generated through the feature selection approach. The recommended GC-LSTM and the technique for feature selection were evaluated using AUCPRC and AUCROC, and the results of these evaluations are shown in Table 5 and figure 8.

Table 6: Accuracy, Detection Rate, and Fault-Tolerance of the Proposed GC-LSTM vs a Feature Selection Approach

UNSW-NB15 Dataset				**KDDCUP99 Dataset**			
Methods	**Accuracy (%)**	**DR (%)**	**FAR (%)**	**Methods**	**Accuracy (%)**	**DR (%)**	**FAR (%)**
Proposed GC-LSTM	96.45	91.85	3.8	**Proposed GC-LSTM**	94.36	87.42	5.2
Feature Selection [20]	89.31	84.38	18.65	**Feature Selection [20]**	84.29	81.94	22.68

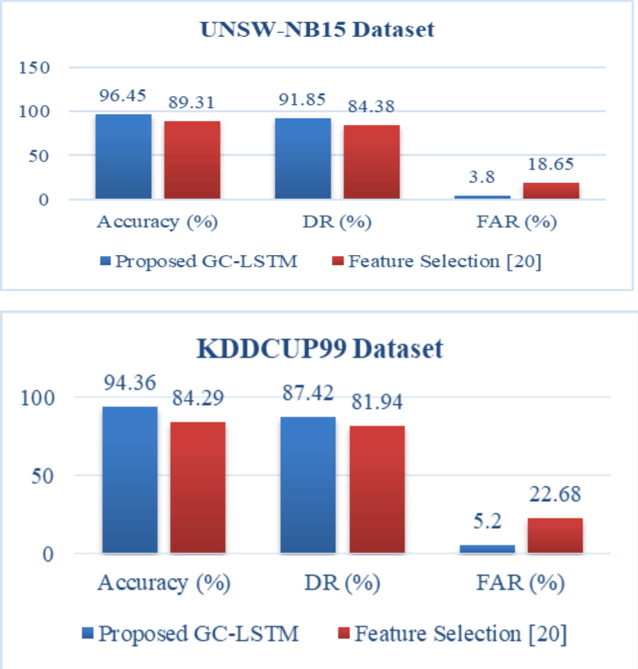


Fig 9: Accuracy, Detection Rate, and Fault-Tolerance of the Proposed GC-LSTM vs a Feature Selection Approach in UNSW-NB15 Dataset and KDDCUP99 Dataset.

Table 6 and figure 9 displays the results of a comparison between GC-LSTM and a feature selection approach in terms of accuracy, DR, and FAR. The table clearly shows that the GC-LSTM model is more accurate than the feature selection approach by roughly 7%. Regarding DR, the two approaches are almost identical. However, the GC-LSTM technique is 60% more efficient than the feature selection approach.

Table 7: Evaluation of the Proposed GC-LSTM compared to various deep learning approaches based on the AUCPRC and AUCROC measures.

UNSW-NB15 Dataset			
Methods	**AUCPRC_0**	**AUCPRC_1**	**AUCROC**
CNN [17]	0.984	0.981	0.984
BiDLSTM [18]	0.983	0.987	0.981
CNN-LSTM [19]	0.986	0.989	0.983
Propo	0.9	0.9	0.9

KDDCUP99 Dataset			
Methods	**AUCPRC_0**	**AUCPRC_1**	**AUCROC**
CNN [17]	0.961	0.958	0.953
BiDLSTM [18]	0.964	0.967	0.963
CNN-LSTM [19]	0.968	0.972	0.971
Propo	0.9	0.9	0.9

sed GC-LSTM	93	98	94

sed GC-LSTM	85	88	85

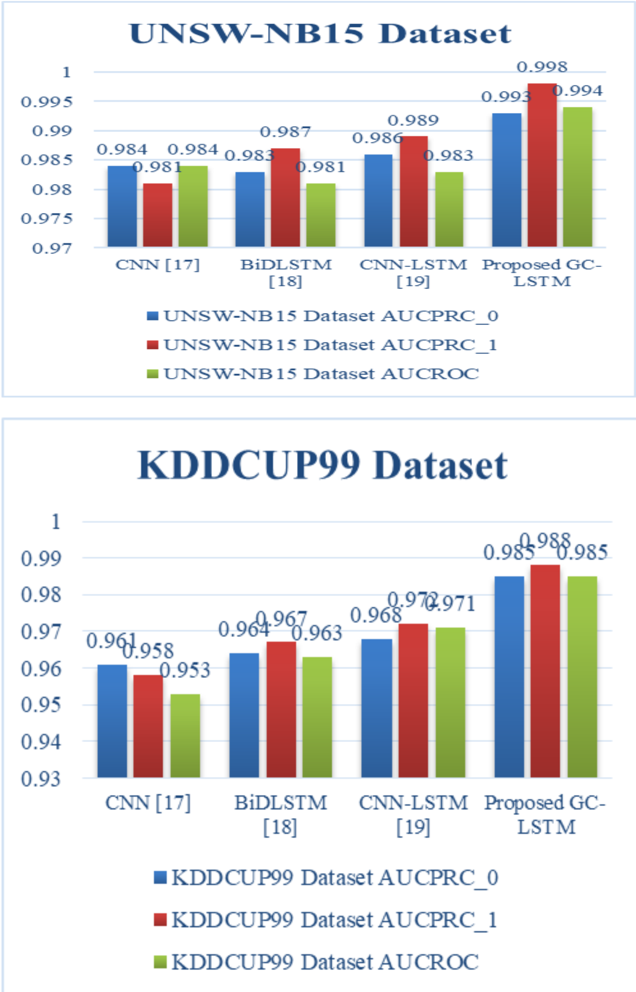
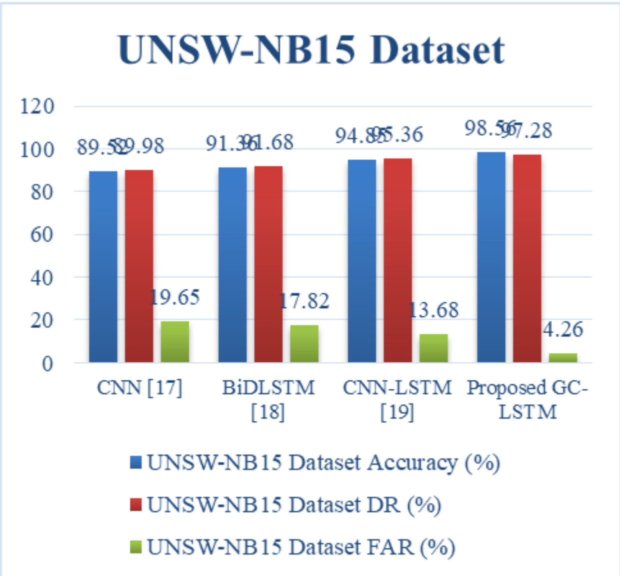


Fig 10: Evaluation of the Proposed GC-LSTM compared to various deep learning approaches based on the AUCPRC and AUCROC measures in UNSW-NB15 Dataset and KDDCUP99 Dataset.

Table 7 and figure 10 displays the results of AUCPRC and AUCROC comparisons for all models. Limited spatial aspects of movement flows are extracted by the CNN algorithm using multiple convolution kernels, while temporal features are extracted by the BiDLSTM model using the memory unit. The two models' results on the three benchmarks in the table are quite similar. Both the spatial feature extraction skills of CNN and the temporal feature extraction capabilities of LSTM are included in the CNN-LSTM model. CNN-LSTM outperforms CNN and BiDLSTM by around 0.2% across the board. Although effective in extracting features related to visual structure, the CNN model has its limits. The GC-LSTM model outperforms the CNN-LSTM model on three criteria, with an average 0.2% improvement.

Table 8: The accuracy, DR, and FAR of the proposed GC-LSTM compared to those of existing deep learning techniques.

UNSW-NB15 Dataset				**KDDCUP99 Dataset**			
Methods	**Accuracy (%)**	**DR (%)**	**FAR (%)**	**Methods**	**Accuracy (%)**	**DR (%)**	**FAR (%)**
CNN [17]	89.52	89.98	19.65	CNN [17]	85.82	86.79	21.53
BiDLSTM [18]	91.36	91.68	17.82	BiDLSTM [18]	89.73	90.28	18.69
CNN-LSTM [19]	94.85	95.36	13.68	CNN-LSTM [19]	92.69	91.63	15.27
Proposed GC-LSTM	98.56	97.28	4.26	Proposed GC-LSTM	97.12	96.83	5.83



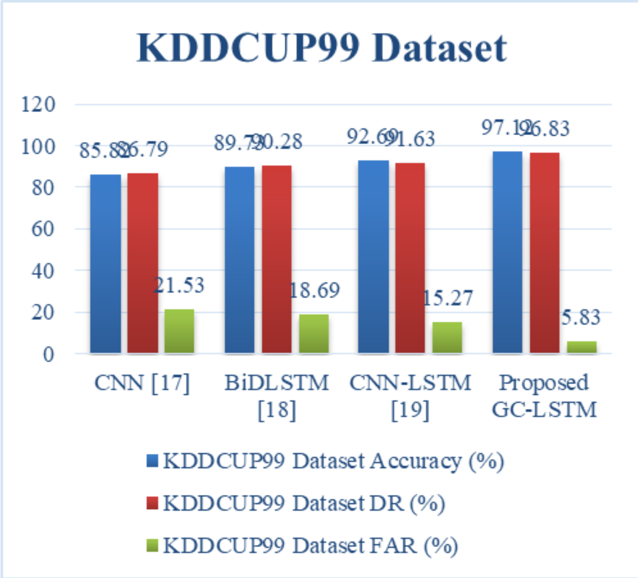


Fig 11: The accuracy, DR, and FAR of the proposed GC-LSTM compared to those of existing deep learning techniques in UNSW-NB15 Dataset and KDDCUP99 Dataset.

On the test set, these four feature extraction models are compared in terms of accuracy, DR, and FAR in Table 8 and figure 11. The CNN-LSTM approach outperforms both CNN and the BiDLSTM technique. With a DR of 98.56%, the CNN-LSTM approach outperforms the other three models by around 4.43% compared to the GC-LSTM approach. The GC-LSTM technique, on the other hand, outperforms the CNN-LSTM approach in terms of accuracy and FAR, with a 9.44% reduction in FAR.

Table 9: Metrics comparing the proposed GC-LSTM to various normal class models.

UNSW-NB15 Dataset				**KDDCUP99 Dataset**			
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
CNN [17]	0.91	0.91	0.91	CNN [17]	0.89	0.91	0.91
BiDLSTM [18]	0.94	0.93	0.94	BiDLSTM [18]	0.92	0.92	0.92
CNN-LSTM [19]	0.96	0.96	0.95	CNN-LSTM [19]	0.95	0.94	0.94
Proposed GC-LSTM	0.99	0.97	0.96	Proposed GC-LSTM	0.98	0.97	0.97

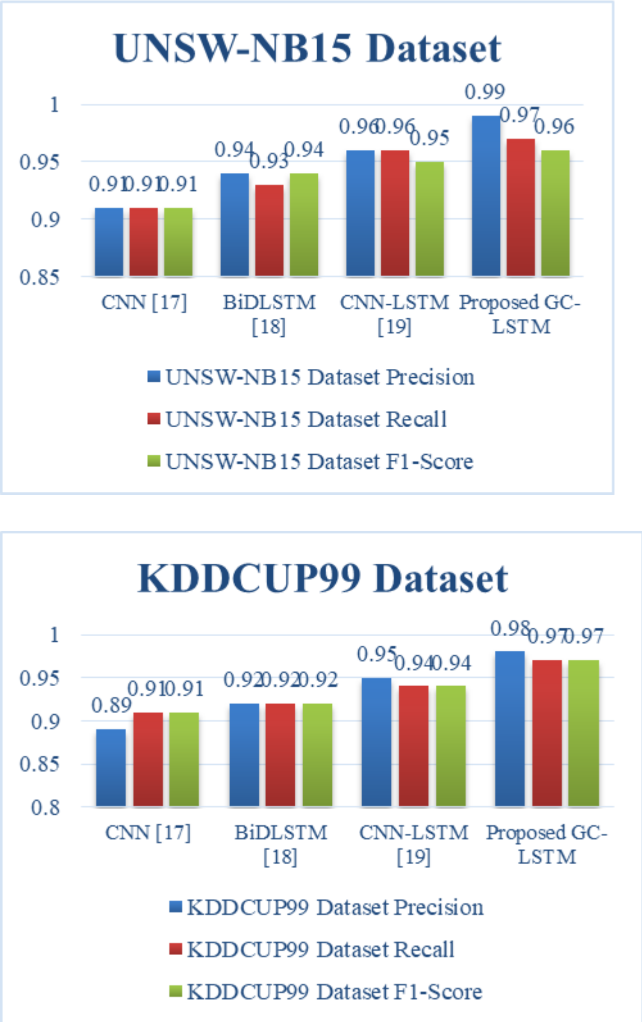


Fig 12: Metrics comparing the proposed GC-LSTM to various normal class models in UNSW-NB15 Dataset and KDDCUP99 Dataset.

Table 10: Evaluation of the Proposed GC-LSTM Model against Competing Models for the Abnormal Class.

UNSW-NB15 Dataset			
Class	**Precision**	**Recall**	**F1-Score**
CNN [17]	0.88	0.91	0.91
BiDLSTM [18]	0.92	0.91	0.92
CNN-LSTM [19]	0.94	0.93	0.92
Proposed GC-LSTM	0.98	0.96	0.94

KDDCUP99 Dataset			
Class	**Precision**	**Recall**	**F1-Score**
CNN [17]	0.86	0.88	0.89
BiDLSTM [18]	0.89	0.91	0.91
CNN-LSTM [19]	0.92	0.91	0.89
Proposed GC-LSTM	0.98	0.95	0.91

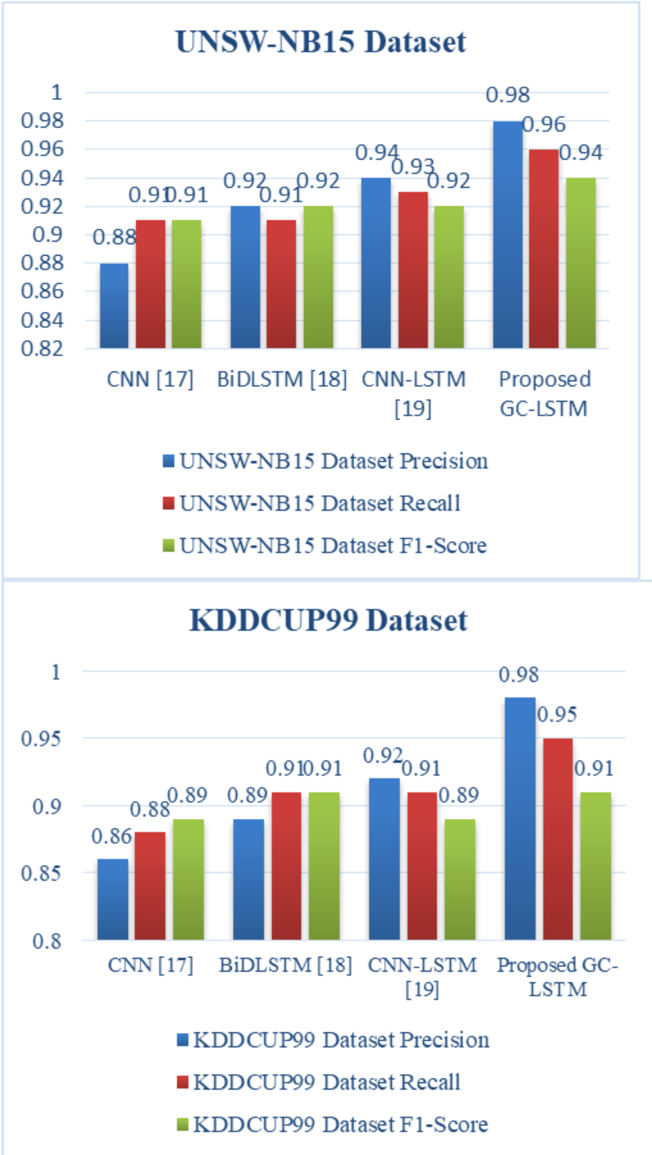


Fig 13: Evaluation of the Proposed GC-LSTM Model against Competing Models for the Abnormal Class in UNSW-NB15 Dataset and KDDCUP99 Dataset.

Tables 9 ,10 and figure 12. 13 provide measurements of four feature extraction models for the typical and pathological classes, whereas Table 8 displays the metrics of the basic classifier, Xgboost.

6. Conclusion

This study investigates the problem of traffic categorization, makes suggestions for how to set up the topological graph structure of network traffic, and offers a solution based on the proposed GC-LSTM. The GC layer is used to analyze the input and extract spatial characteristics, and then the LSTM model is used to extract probable temporal information. A portion of the UNSW-NB15 and KDDCUP99 data sets are used to compare the performance and efficacy of the proposed method to feature selection and other well-known deep learning techniques, including Convolutional Neural Networks, Bidirectional LSTM, and Convolutional Neural Network-LSTM. The experiment has certain flaws and room for improvement as well. For network traffic data, creating a topological graph with many nodes places a heavy load on the system's resources due to the increased number of undirected edges that must be constructed. Future research into the correlations between traffic flows and their normal and abnormal counterparts may be informed by this article's proposal of applying a graph convolution model in a network traffic environment.

References

[1] M. Sadeghzadeh, S. Shiravi and R. Jalili, "Adversarial Network Traffic: Towards Evaluating the Robustness of Deep-Learning-Based Network Traffic Classification," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1962-1976, June 2021, doi: 10.1109/TNSM.2021.3052888.

[2] Z. Bu, B. Zhou, P. Cheng, K. Zhang and Z. -H. Ling, "Encrypted Network Traffic Classification Using Deep and Parallel Network-in-Network Models," in IEEE Access, vol. 8, pp. 132950-132959, 2020, doi: 10.1109/ACCESS.2020.3010637.

[3] S. S. Sepasgozar and S. Pierre, "Network Traffic Prediction Model Considering Road Traffic Parameters Using Artificial Intelligence Methods in VANET," in IEEE Access, vol. 10, pp. 8227-8242, 2022, doi: 10.1109/ACCESS.2022.3144112.

[4] Y. Wang, X. Yun, Y. Zhang, C. Zhao and X. Liu, "A Multi-Scale Feature Attention Approach to Network Traffic Classification and Its Model Explanation," in IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 875-889, June 2022, doi: 10.1109/TNSM.2022.3149933.

[5] S. Fathi-Kazerooni and R. Rojas-Cessa, "GAN Tunnel: Network Traffic Steganography by Using GANs to Counter Internet Traffic Classifiers," in IEEE Access, vol. 8, pp. 125345-125359, 2020, doi: 10.1109/ACCESS.2020.3007577.

[6] S. Fathi-Kazerooni and R. Rojas-Cessa, "Countering Machine-Learning Classification of Applications by Equalising Network Traffic Statistics," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3392-3403, 1 Oct.-Dec. 2021, doi: 10.1109/TNSE.2021.3113656.

[7] M. S. Sheikh and Y. Peng, "Procedures, Criteria, and Machine Learning Techniques for Network Traffic Classification: A Survey," in IEEE Access, vol. 10, pp. 61135-61158, 2022, doi: 10.1109/ACCESS.2022.3181135.

[8] Galal and X. Hesselbach, "Machine Learning Models for Traffic Classification in Electromagnetic Nano-Networks," in IEEE Access, vol. 10, pp. 38089-38103, 2022, doi: 10.1109/ACCESS.2022.3165013.

[9] K. -H. N. Bui, H. Oh and H. Yi, "Traffic Density Classification Using Sound Datasets: An Empirical Study on Traffic Flow at Asymmetric Roads," in IEEE Access, vol. 8, pp. 125671-125679, 2020, doi: 10.1109/ACCESS.2020.3007917.

[10] Mareri, G. Owusu Boateng, R. Ou, G. Sun, Y. Pang and G. Liu, "MANTA: Multi-Lane Capsule Network Assisted Traffic Classification for 5G Network Slicing," in IEEE Wireless Communications Letters, vol. 11, no. 9, pp. 1905-1909, Sept. 2022, doi: 10.1109/LWC.2022.3186529.

[11] Shamsimukhametov, A. Kurapov, M. Liubogoshchev and E. Khorov, "Is Encrypted ClientHello a Challenge for Traffic Classification?," in IEEE Access, vol. 10, pp. 77883-77897, 2022, doi: 10.1109/ACCESS.2022.3191431.

[12] F. U. Islam, G. Liu, J. Zhai and W. Liu, "VoIP Traffic Detection in Tunneled and Anonymous Networks Using Deep Learning," in IEEE Access, vol. 9, pp. 59783-59799, 2021, doi: 10.1109/ACCESS.2021.3073967.

[13] Q. Liu, M. Li, N. Cao, Z. Zhang and G. Yang, "Improved Harris Combined With Clustering Algorithm for Data Traffic Classification," in IEEE Access, vol. 10, pp. 72815-72824, 2022, doi: 10.1109/ACCESS.2022.3188866.

[14] Y. Pan, X. Zhang, H. Jiang and C. Li, "A Network Traffic Classification Method Based on Graph Convolution and LSTM," in IEEE Access, vol. 9, pp. 158261-158272, 2021, doi: 10.1109/ACCESS.2021.3128181.

[15] H. D. Trinh, Á. Fernández Gambín, L. Giupponi, M. Rossi and P. Dini, "Mobile Traffic Classification Through Physical Control Channel Fingerprinting: A Deep Learning Approach," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1946-1961, June 2021, doi: 10.1109/TNSM.2020.3028197.

[16] Shahraki, M. Abbasi, A. Taherkordi and A. D. Jurcut, "Active Learning for Network Traffic Classification: A Technical Study," in IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 1, pp. 422-439, March 2022, doi: 10.1109/TCCN.2021.3119062.

[17] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC), pp. 218–224, Feb. 2020.

[18] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," Expert Syst. Appl., vol. 185, Art. no. 115524, Dec. 2021.

[19] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," Secur. Commun. Netw., vol.12, pp. 1–11, Aug. 20, 2020

[20] H. Alizadeh, H. Vranken, A. Zúquete and A. Miri, "Timely Classification and Verification of Network Traffic Using Gaussian Mixture Models," in IEEE Access, vol. 8, pp. 91287-91302, 2020, doi: 10.1109/ACCESS.2020.2992556.

[21] Yao, Y. Yang, K. Yin and J. Yang, "Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network," in IEEE Access, vol. 10, pp. 103136-103149, 2022, doi: 10.1109/ACCESS.2022.3210189.

[22] N. Khan, U. Chaudhuri, B. Banerjee, and S. Chaudhuri, "Graph convolutional network for multi-label VHR remote sensing scene recognition," Neurocomputing, vol. 357, pp. 36–46, May 2019.

[23] J. Wu, S.-H. Zhong, and Y. Liu, "Dynamic graph convolutional network for multi-video summarisation," Pattern Recognit., vol. 107, Nov. 2020, Art. no. 107382

[24] Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in Proc. 36th Int. Conf. Mach. Learn., vol. 97, Jun. 2019, pp. 6861–6871

[25] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524.

[26] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," IEEE Trans. Ind. Informat., vol. 17, no. 5, pp. 3469–3477, May 2021.

[27] https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15?resource=download

[28] https://kdd.ics.uci.edu/databases/kddcup99/task.html

[29] P. Singh and A. Tiwari, "A review intrusion detection system using KDD'99 dataset," Int. J. Eng. Res. Technol., vol. 3, no. 11, pp. 1103–1108, 2014

[30] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in Proc. ACM Workshop Wireless Secur. Mach. Learn., 2020, pp. 25–30.

[31] A. Salih and I. Duhok, "Evaluation of classification algorithms for intrusion detection system: A review," J. Soft Comput. Data Mining, vol. 2, no. 1, pp. 31–40, Apr. 2021.

[32] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC), Feb. 2020, pp. 218–224

[33] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524.

[34] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," Secur. Commun. Netw., vol. 2020, pp. 1–11, Aug. 20

[35] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," Expert Syst. Appl., vol. 148, Jun. 2020, Art. no. 113249.