DETECTION OF THE NETWORK INTRUSION TRAFFIC USING DEEP LEARNING

NADIA ALSABBAGH
Nad.q.alneeme@gmail.com
Department of Computer Engineering
Yildiz Technical University
Istanbul, Turkey

HASAN HÜSEYIN BALIK
hasanbalik@gmail.com
Department of Computer Engineering
Istanbul Aydin University
Istanbul, Turkey

Abstract— In this paper, due to limited memory, processing power, and lack of security features in many IoT devices, these devices are particularly vulnerable to network intrusion attacks. To address this problem, the paper stresses the importance of implementing effective security measures and demonstrating the potential of learning technologies to detect and prevent network intrusion attacks on these devices. The proposed framework is useful for enhancing device security and reducing the risk of attacks and data breaches. We used an intrusion detection system with artificial intelligence (AI) by using DNN and then tested with the dataset (KDD Cup_99) we utilized to address attacks on the network. Subsequently, we preprocessed the dataset by employing normalization and one-hot encoding for input into the DNN model. The refined data underwent the application of the DNN algorithm to construct a learning model, and the validation was conducted using the complete dataset. The data is split into 80% training, 20% testing and taking from the training dataset 20% for evaluation of the result. Accuracy, Detection rate, loss & Precision were calculated to confirm the detection efficacy of the LSTM and MLP at (binary and multiclass) models, which we found to generate performance-acceptable results for intrusion detection. The average Accuracy of the four models is 99.99% and loss is minimum.

Index Terms— ***Binary class, Deep Learning, Intrusion Detection System (IDS), LSTM, MLP, Multiclass classification.***

1. INTRODUCTION

With the recent years of information technology (IT), the complexity of numerous information devices has increased, connected concurrently, persisting to produce and save significant digital data .

Accordingly. This introduces challenges to individuals and corporations. Given these points, the approaches to attack detection should also be more competent and practical than before to combat attack hackers, which are also developing persistently .For detecting abnormal demeanours (attacks); the current technologies commonly employ detection methods based on event examination with preset rules. So, insufficient data regarding real attacks or inaccurate identification of attacks in the security domain; result in financial losses and the limited utilization of such systems . Consequently, their primary function is the automated gathering and analysis of varied security event data for risk assessment. To ensure essential computing capacity, automatic intrusion detection systems must leverage cloud computing resources and acquire insights into attack methodologies for adaptive responses in the security industry. This study introduces an examination of a system for detecting abnormalities based on artificial intelligence (AI), employing a deep neural network (DNN). The objective is to create a rapid and efficient intelligent intrusion detection system capable of addressing emerging threats.

We used The KDD Cup 99 dataset for the study, then evaluated the preprocess by applying the normalize and One-hot-encoding and split 80% training. however, with LSTM and MLP for (binary and multiclass); the accuracy, detection rate, loss, Precision, and F1-score of the intrusion detection process were calculated to ascertain its

effectiveness. The average accuracy is 99.99% and the loss was decreased.

2. RELATED WORKS

2.1 Intrusion Detection Research

The intrusion detection system (IDS) observes potential threats through the analysis of attack patterns. Due to its reliance on a set of rules, it tends to exhibit a relatively elevated rate of false detections. Past research has aimed at pinpointing attack patterns through precision learning to mitigate the occurrence of false detections. This study explores and evaluates an intrusion detection model that employs a blend of various machine learning algorithms, including LSTM and MLP.

In recent years, research has been carried out in the field of intrusion detection systems utilizing deep learning, specifically employing the artificial neural network (ANN) algorithm. This algorithm represents a progression beyond conventional machine learning, where the processes of pattern extraction and learning are distinct. In contrast to prevalent intrusion detection approaches generating rules or models for identifying malicious attack patterns, the deep learning method establishes relationships directly from secured data to identify anomalous risks. H. Wang and Q. Gao, Ni. Gao, Li. Gao (2014) conducted an experimental of intrusion detection utilizing deep belief networks (DBNs) demonstrated a notable improvement in accuracy, surpassing 6% compared to the established ANN model, as evidenced in a test utilizing the KDD Cup 99 [1].

K. Al. and C. Purdy (2016) suggested the utilization of a Restricted Boltzmann Machine (RBM) featuring a single concealed layer along with a multiclass soft-max. They assessed their proposed approach using the KDD Cup_99 dataset, asserting a 97.90% detection rate and a 2.47% false negative rate [2].

Concerning multiclassification, Hasan, M., Milon Islam, M.d., Islam, I., & Hashem, M. M. A. (2019) employed various machine learning techniques. The study evaluated models like “logistic regression (LR), decision tree (DT), RF, and artificial neural network (ANN) using an exclusive dataset crafted by the researchers, not accessible to the public. The research findings indicated that RF emerged as the most effective model for multiclass classification" [3].

Arnaud Rosay, Florent Carlier and Pascal Leroux (2020) employed "An intrusion detection system employing a multilayer perceptron neural network achieves accuracy exceeding 99% while maintaining a false positive rate below 0.7%" [4].

Andrew Churcher, William J. Buchanan, and colleagues (2021). conducted an "experimental investigation into the categorization of attacks in IoT networks through the application of various machine learning (ML) techniques, including k-nearest neighbour (KNN), support vector machine (SVM), decision tree (DT), naive Bayes (NB), random forest (RF), artificial neural network (ANN), and logistic regression (LR) within intrusion detection system (IDS). Through evaluation based on metrics such as accuracy, precision, recall, F1 score, and log loss, the study found that in the context of HTTP distributed denial-of-service (DDoS) attacks, the random forest (RF) achieved an accuracy of 99%. Additionally, results from simulations indicated that RF consistently outperformed other ML algorithms in binary classification, showcasing higher precision, recall, F1 score, and lower log loss across various attack types. Nevertheless, in multiclass classification scenarios, KNN emerged as the superior ML algorithm, achieving an accuracy of 99%, which surpassed RF by 4%" [5].

2.2. Artificial Neural Network (ANN)

In the field of machine learning, the artificial neural network algorithm has been developed in a manner inspired by biological neural networks, using statistical learning techniques. It has evolved with the integration of the back-propagation algorithm, dropout, and the rectified linear unit (ReLU) function. In an artificial neural network, computations occur within nodes arranged in layers, mimicking human neural networks. These nodes activate based on input values multiplied by node weights, with adjustments allowing for different weight assignments. The multiplied values are then summed, passed through an activation function, and utilized for classification or regression analyses. Recently, the ANN algorithm has shown effectiveness across various domains, demonstrating success in tasks such as recognition, reasoning, and prediction. ANNs are the most commonly used soft computing technique in IDSs [6],[7],[8],[9],[10].

Jin Kim, Nara Shin, Seung Yeon Jo, and Sang Hyun Kim (2017) "implemented a DNN model for an artificial intelligence intrusion detection system. Their findings revealed an impressive overall accuracy and detection rate of approximately 99%. Furthermore, they accomplished a low false alarm rate of 0.08%, indicating minimal chances of incorrectly categorizing normal data as malicious attacks "[11].

Tran Nguyen Ngoc, Nathan Shone, Vu Dinh Phai, and Qi Shi (2018), "present a deep learning technique for intrusion detection. They employed a deep learning classification model constructed using stacked NDAEs and evaluated using the benchmark KDD Cup '99 and NSL-KDD datasets" [12]. They have discussed the problems faced by existing NIDS techniques, and the NDAE method for unsupervised feature learning. Then constructed a classification model from stacked NDAEs and the RF classification algorithm. The accuracy and training time reduction of up to 98.81% [12].

Cosimo I., Ahsan Adeel, Francesco Carlo Morabito, and Amir Hussain (2019), discussed "The Intrusion Detection System (IDS) utilizes a combination of data analytics, statistical methods, and advancements in machine learning theory to extract enhanced and more interconnected attributes. To validate the effectiveness of the IDS, testing is conducted using the NSL-KDD database as a benchmark. They found that the developed IDS outperforms deep learning and traditional shallow machine learning" [13].

Ayei E. Ibor, Florence A. Oladeji, OB. Okunoye, Charles O. Uwadia (2019) discusses "the use of unsupervised statistical and supervised deep learning techniques to predict attacks by mapping hyper-alerts to class labels of attacks and proposes a model for predicting cybera_ttacks using a hybrid approach" [14]. This improves the procedures of extracting features and transforming them, aiming to provide an organized understanding of the dynamic profiles within a network. "The model incorporates such modules as alert normalization, dimensionality reduction, prediction and reporting using unsupervised feature filtering and supervised deep learning techniques for prediction of attack types. The choice of both supervised and unsupervised methods is significant for constructing hyper-alerts, which can be mapped to labelled classes of attacks, with the added advantage of defining new classes of attacks as the model learns non-linear representations of the feature space" [14].

Kumar Sahu, S. Sharma, M. T., and Rohit Raja (2021), discussed the extraction and classification of accurate data representation using the long memory model was discussed where twenty devices were used from the Internet of Things (LSTM). The accuracy of the experimental study in detecting the attack was %96. "The suggested approach employs a convolutional neural network to capture an accurate data representation and categorize it through a long memory model" [15].

Yogita Shewale, Shailesh Kumar, and Satish Banait (2023) utilized "MLP and LSTM classifiers to establish Intrusion Detection Systems using the CICDDoS2019 dataset. They assessed performance by considering accuracy and loss metrics and observed that the LSTM classifier outperformed the MLP classifier in terms of accuracy and loss during both training and validation phases".[16]

3. METHODOLOGY

Despite being present for more than seventy years, connection structures have gained prominence in artificial intelligence research recently, thanks to novel architecture and the inclusion of graphical processing units (GPUs). Deep learning encompasses a diverse array of algorithms and configurations rather than being a singular strategy, offering solutions to a wide range of problems.

While not a recent concept, deep learning is currently undergoing substantial growth, propelled by the combination of deeply layered neural networks and the accelerated execution provided by GPUs. The surge is further driven by the availability of vast amounts of big data. Deep learning relies on training neural networks with example data and rewarding them based on their training success, making a larger dataset advantageous for constructing these structures. Deep learning employs numerous architecture and algorithms, with supervised and unsupervised learning categorizing these structures. Among the oldest and most widely used methods in various applications are long short-term memory (LSTM) and MultiLayer Perceptron (MLP). Figure. 1 illustrates the framework of the proposed system of this paper.

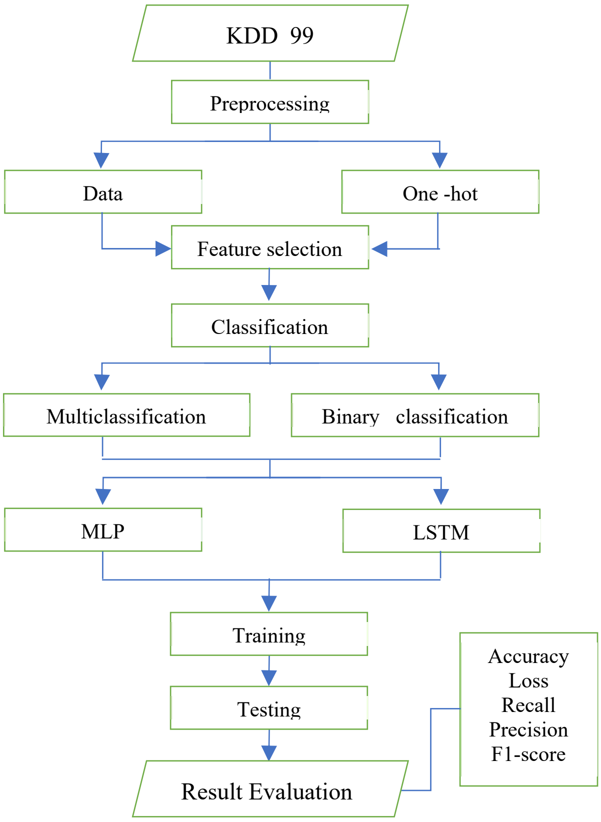


Fig. 1: A framework of the proposed system

3.1 Dataset

This research utilized the KDD CUP 99 dataset, accessible in both CSV and JSON formats. This dataset was employed during the model and evaluation phases, offering modifiability, extensibility, and reproducibility [17].
"The KDD CUP 99 dataset, introduced in 1999, has garnered widespread popularity. Developed by MIT Lincoln Labs, its primary purpose is to provide a standardized dataset featuring a diverse range of cyber-attacks, facilitating comprehensive research and assessment of intrusion detection. The dataset comprises nine weeks of unprocessed TCP dump data from a simulated U.S. Air Force network, inclusive of various attacks. To enhance usability, packets associated with the same connection are consolidated into connection records" [18].
Specifically, "the training data, derived from seven weeks of collection, is processed into approximately five million connection records, while the test data, collected over two weeks, comprises around two million connection records "[18]. The training data is intended for the machine learning technique learning phase, whereas the test data serves to evaluate the performance of a fully trained solution.
Each connection is categorized as either normal or one of four attack types: Denial of Service (DoS), network probe (Probe), Remote to Local (R2L), and User to Root (U2R), as shown in Table I. A DoS attack aims to render a machine or service unavailable, while a probe attack involves malicious network activities, such as port scanning, to understand the network's architecture. R2L attacks occur when an attacker gains local access to a system through the network, and U2R attacks exploit system vulnerabilities to acquire user privileges. Notably, the dataset exhibits strong imbalance, with significantly more DoS attacks than U2R attacks. Each connection record contains 7 discrete and 34 continuous attributes for a total of 41 attributes. Additionally, each connection is characterized by 41 derived attributes, as outlined in Table II; by the classifier we analyze these features, whether numerical or symbolic, to differentiate between normal connections and attacks. Despite its popularity, the KDD Cup 99 dataset faces criticism for various deficiencies highlighted in analyses [18] [19]. Problems include a substantial number of redundant and duplicated records.

Table I: Attack profiles of DoS, R2L, U2R and Probe classes.

Attack class	Attack profile
DoS	neptune, back, pod, land, smurf, tear, drop, apache2, mailbomb, processtable, udpstorm, snmpgetattack.
R2L	ftp_write, imap, guess_passwd, multihop, phf, warezclient, spy, warezmaster, named, sendmail, snm, pguess, xlock, xsnoop, warm.
U2R	loadmodule, buffer_overflow, root, kit, perl, xterm, sqlattack, httptunnel.
Probe	nmap, ipsweep, satan, portsweep, mscan, saint.

Table II: Examples of KDD CUP 99 attributes [19].

Attribute name	Type	Description
Protocol type	Discrete	type of the protocol, e.g. tcp, udp, etc.
Service	Discrete	network service on the destination, e.g., http, telnet, etc.
Flag	Discrete	normal or error status of the connection.
Logged in	Discrete	1 if successfully logged in; 0 otherwise.
Land	Discrete	1 if the connection is from/to the same host/port; 0 otherwise
Duration	Continuous	Length (number of seconds) of the connection
Dst bytes	Continuous	number of data bytes from destination to source.
Src bytes	Continuous	number of data bytes from source to destination.
Urgent	Continuous	number of urgent packets.
wrong fragment	Continuous	number of "wrong" fragments.
Hot	Continuous	number of "hot" indicators.
Num failed logins	Continuous	number of failed login attempts.

3.2 Data Preprocess

Figure 2, Table III shows, the dataset that we used in this paper includes traffic with both regular and irregular connections on the network. Each record shows 23 attack patterns or normal traffic and fields for 41 other features, having a mixture of numeric values and symbolic values. The neural network is based on numeric values.

Table III: Number of Attack labels

Name of Attack	Smurf	neptune	normal	back
No. of Attack	280790	107201	97277	2203
Name of Attack	Satan	ipsweep	portsweep	Warezclient
No. of Attack	1589	1247	1040	1020
Name of Attack	teardrop	pod	nmap	guess_passwd
No. of Attack	979	264	231	53
Name of Attack	buffer_overflow	land	warezmaster	Imap
No. of Attack	30	21	20	12
Name of Attack	rootkit	loadmodule	ftp_write	Multihop
No. of Attack	10	9	8	7
Name of Attack	Phf	perl	spy	
No. of Attack	4	3	2	

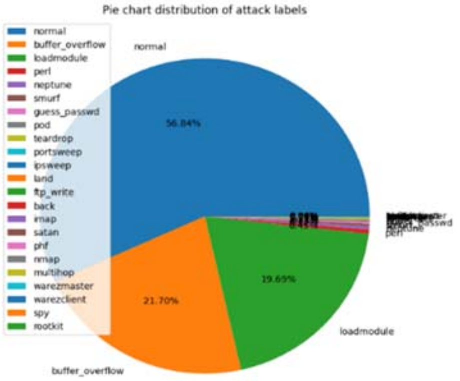


Fig. 2: Distribution of attack labels

3.2.1 Data Normalization

We used normalization because integer values were mixed with floating point values between 0 and 1, which made learning difficult, as in Table IV and Table V.

Numeric attributes values K_{ij} were transformed into the range 0 to 1 using the min-max normalization technique, it aims to standardize features so that they are on a comparable level, which enhances the model's performance and training consistency as per:

$$\widetilde{K}_{fj} = \frac{K_{fj}-\min(K_f)}{\max(K_f)-\min(K_f)} \tag{1}$$

The max and min values of the numeric attribute K_f denote a $\max(K_f)$ and m (K_f) respectively, while $\widetilde{K}_{fj}$ represents the normalized feature value within the range of 0 to 1.

3.2.2 One-Hot-Encoding

We created new variables that take on values 0 and 1 to represent the original categorical values. For example, if protocol type, service and flag (K2, K3, K4, respectively), the K2 feature (protocol type) has three attributes: tcp, udp and icmp. Applying the one-hot-encoding technique they were converted into binary vectors: [1,0,0], [0,1,0], [0,0,1], respectively. Similarly, also K3 and K4 features (service and flag) we transformed into one-hot-encoding vectors. Overall, the 41-dimensional features were mapped into 122-dimensional features (38 continuous and 84 with binary values related to the features K2, K3 and K4). As seen in Table VI.

4. BINARY AND MULTICLASSIFICATIONS

Each algorithm has been trained and tested in terms of binary and multiclass classification. To gain additional insight into where each algorithm provides the finest performance, the algorithms have been trained and tested in terms of binary and multiclass classification. The results from this study will allow a better overview of these available algorithms for their adaptability in a specific scenario to achieve an optimal performance whether it is a binary or multiclass classification. Additionally, this paper provides a strong basis for further research into the integration of LSTM and MLP with intrusion detection and to present findings which can aid when selecting an appropriate LSTM and MLP algorithm for the requirements of effectiveness.

4.1 Binary Classification

Binary classification predicts one of two outcomes ("positive" and "negative") based on input features. The process involves gathering a labeled dataset for training machine learning algorithms like logistic regression or decision trees. After training, the model predicts the class of new instances by applying a threshold to the probability estimate of the positive class. Evaluation metrics such as accuracy, precision, recall, and F1 score assess the model's performance on a test dataset. The goal is to develop a model that excels on new, unseen data [20]. As in Figure 3:

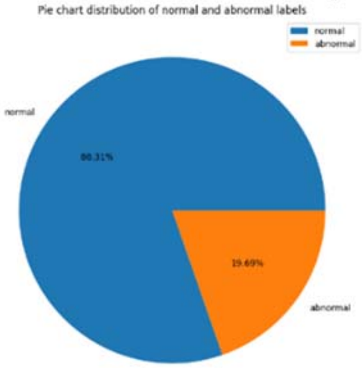


Fig. 3: Distribution of Binary labels

4.2 Multiclass Classification

Multiclass classification involves predicting one of several outcomes based on input features. It deals with more than two possible outcomes, each represented by a distinct label. To address this, a labeled dataset is collected and used to train machine learning algorithms like decision trees, random forests, k-nearest neighbors, or neural networks. Once trained, the model predicts the class of new instances by outputting probability estimates for each class label, choosing the one with the highest probability. Evaluation metrics such as accuracy, precision, recall, and F1 score assess the model's performance on a separate test dataset, aiming for accurate classification of new, unseen data [20]. Look at figure 4:

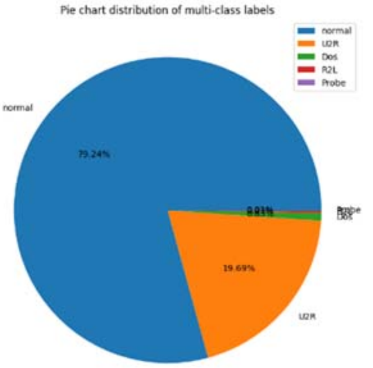


Fig. 4: Distribution of Multiclass labels

4.3 Classification Techniques

Two developed deep models utilizing MultiLayer Perceptron (MLP) and Long Short-Term Memory (LSTM) architectures have been created for the identification of normal and abnormal classifications within the KDD_99 dataset, encompassing Normal, Denial of Service (DoS), Remote to Local (R2L), and Probe categories. Details of these models are provided in the subsequent sections:

4.3.1 LSTM (Long Short-Term Memory) Classifier

This study, Figure 5.1a shows the designed LSTM classifier. A model based on long short-term memory (LSTM) was created. LSTM serves as the memory block in a recurrent neural network (RNN) [21]. The standard LSTM architecture, illustrated in Figure 5.1b and Figure 5.1c, comprises a cell (C_t'), an input gate (i_t), an output gate (o_t) and a forget gate (f_t)Within a layer of LSTM units, the model can grasp prolonged dependencies among time steps in a data sequence. This LSTM layer possesses two states: the hidden state (or output state) housing the output at time step t, and the cell state, preserving information acquired from preceding time steps. The hidden and cell states are updated at each time step t through the utilization of the mentioned gates.

It consists of an input layer, one LSTM layer and an output layer. Specifically, for a fair comparison, the LSTM layer used 64 cells(neurons) for encoding the input information. which represents the number of memory units or cells in the LSTM layer. and input dimensions of 118 and 93 at binary and multiclassification respectively.

The output was fed into the dense fully connected layer which has 2 or 5 neurons with "*sigmoid*" "and "*SoftMax*" activation functions (binary and multiclassification respectively).

The dropout layer with a dropout rate of 0.2 to reduce overfitting.

The LSTM model was trained using a comparative model: "*ADAM"* (adaptive moment

estimation) optimizer, the loss: "*Binary crossentropy*" and "*categorical crossentropy"* in Binary class and multiclassification, respectively) and metric "*accuracy*".

The definite must make the loss "*binary crossentropy"* when we have two categorical. at the same time, making the loss" *categorical crossentropy*" when we have more than two categorical.

The input dataset splits to 80% and 20% for training and testing respectively then splits to 75% and 25% for training and testing with test size (0.2) and (0.25) and random state (42). Then training with epochs (20), batch size (500) and validation rate (0.2). These values were selected based on practical experimentation conducted through various tests.

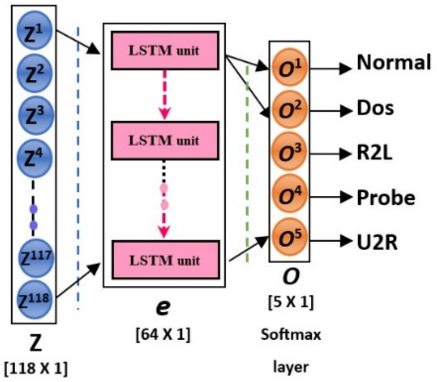


Fig. 5.1a Proposed LSTM Classifier

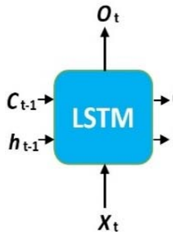


Fig. 5.1b
LSTM cell

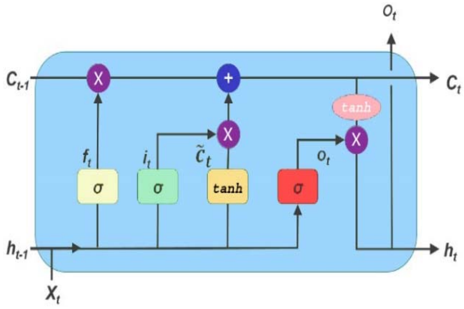


Fig. 5.1c
Architecture of a standard LSTM unit

the corresponding equations for a single timestep.

$$f_t = \sigma_g(W_f \times x_t + U_f \times h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i) \quad (3)$$

$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o) \quad (4)$$

$$C_t' = \sigma_c(W_c \times x_t + U \times h_{t-1} + b_c) \quad (5)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot C_t' \quad (6)$$

$$h_t = o_t \cdot \sigma_c(c_t) \quad (7)$$

Where the

σ_g: sigmoid	σ_c: tanh
h_t : is the hidden state	· : element wish multiplication
f_t : forget gate	i_t : input gate
o_t : output gate	c_t : call state

4.3.2 MLP (MultiLayer Perception) Classifier

This study, Figure 6 shows the designed MLP classifier. The MultiLayer Perceptron (MLP) is a feedforward neural network algorithm that employs a supervised learning algorithm for its training [22]. This widely utilized artificial neural network type is employed in tasks related to both classification and regression. It consists of interconnected nodes (neurons) organized into multiple layers, where each layer transmits its output to the subsequent layer.

It consists of an input layer, with 5 hidden layers and the activation*:" ReLU"* With" *dense"* starts with 118 inputs.

the output was fed into the dense fully connected layer which has 2 or 5 neurons (with "*sigmoid*" and "*SoftMax*" activation functions for binary and multiclassification, respectively).

The dropout layer with a dropout rate of 0.2 to reduce overfitting.

The MLP model was trained using comparative model: "*ADAM"* (adaptive moment

estimation) optimizer, the loss: ("*Binary crossentropy*" and "*categorical crossentropy*" in Binary class and multiclassification, respectively) and metric "*accuracy*".

The definite must make the loss" *binary crossentropy* "when we have two categorical. while making the loss "*categorical crossentropy*" when we have more than two categorical.

The input dataset splits to 80% and 20% for training and testing respectively then split to 75% and 25% for training, and testing with test size: (0.2) and (0.25) and random state (42). Then training with epochs (20), batch size (500) and validation rate (0.2). These values were selected based on practical experimentation conducted through various tests.

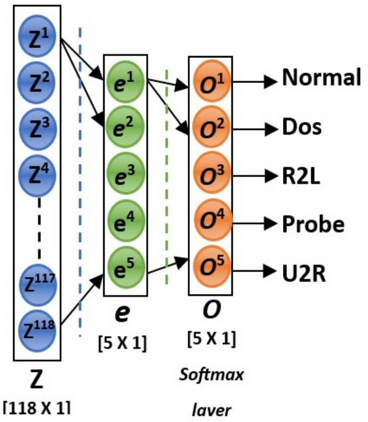


Fig.6: MLP architecture, it referred to multiclassification task

Rectified Linear Unit (ReLU) is a frequently employed activation function found in the concealed layers of neural networks. The mathematical representation of ReLU can be expressed as f(x) = max (0, x), with x denoting the neuron's input. Essentially, ReLU substitutes negative input values with zero while preserving positive values unaltered [23].
Using ReLU as an activation function in an MLP helps the network learn complex representations and speeds up the training process, as it allows for the efficient backpropagation of gradients during the training phase. Expressing it mathematically:

$$y_{out} = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{8}$$

The Sigmoid Function, with its "S" shaped curve, is used in logistic regression and basic neural networks for multianswer classification. Applied to raw output, it produces values within 0 to 1 or -1 to 1[24]. As in the following function:

$$f(x) = sigmoid(x) = \frac{1}{1+e^{-x}} \tag{9}$$

The SoftMax Function, also known as SoftArgMax or Normalized Exponential Function, transforms real number vectors into a probability distribution by normalizing them based on exponentials. It ensures that the elements range from 0 to 1 and sum up to 1, representing a probability distribution. The larger the input number, the higher the resulting probabilities [25]. As in the following:

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} for\ j = 1, \ldots, K \tag{10}$$

ADAM, an abbreviation for Adaptive Moment Estimation, is a widely employed optimization algorithm in the training of artificial neural networks. It is categorized within the family of stochastic gradient descent (SGD) optimization techniques and integrates concepts from two other well-known optimization algorithms, namely RMSprop and Momentum [26].

The random state parameter is crucial in initializing the internal random number generator, influencing how data is divided into training and testing indices. Specifying an integer value ensures result reproducibility, making the data splitting procedure consistent across runs. Without a specified integer, variability arises from the inherent shuffling mechanism, impacting the reliability of subsequent analyses. Explicitly assigning an integer to random state is recommended for control over randomization and result reproducibility in machine learning or data analysis [27].

In machine learning model training, the goal is to balance a deep understanding of the data to avoid overfitting. Overfitting occurs when the model not only captures the genuine patterns but also memorizes noise and outliers, resulting in impressive performance within the training set but limited generalizability [28].

4.4 Performance of The Model

To evaluate the performance of the model, accuracy and loss were calculated, and the degree of detection rate and Precision was recalled and scheduled, as in the following:
Accuracy (A)refers to the proportion of accurately identified outcomes, encompassing both attack and normal traffic. Within the realm of multiclass classification, precision aligns with the "*Jaccard index*", wherein it is calculated as the magnitude of the intersection divided by the magnitude of the union within the label sets.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{11}$$

Precision (P) (also called positive predictive value), alternatively referred to as positive predictive value, denotes the ratio of accurately identified attacks to the total number of identified instances, specifically highlighting the percentage of correctly identified attacks within the identified set.
Also, the percentage of records correctly classified as anomalies out of the total number of records classified as anomalies. Precision is calculated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

Detection rate (DR) is actual attack data are correctly categorized as attacks; also called Recall or True Positive Rate, is the percentage of records correctly classified as anomalies out of the total number of anomaly records. The Recall can be calculated as follows:

$$Recall\ Detection\ Rate\ (DR) = \frac{TP}{TP+FN} \tag{13}$$

F-measure (F) is a measure that combines both precision and detection rate, and it is calculated as follows:

$$F1\_score = 2 * \frac{Precision * Recall}{Precision+Recall} \tag{14}$$

"*TP* refers to predicted values that accurately match actual positive outcomes. *FP*, on the other hand, denotes predicted values that inaccurately identify actual negatives as positives. *FN* involves positive values being wrongly predicted as negatives. Lastly, *TN* represents predicted values that correctly identify actual negative outcomes" [29].
The accuracy, precision, recall, *F1- score* and loss; for the DNN model testing the proposed model (LSTM, and MLP) are summarized in Table VIII and Table IX. The model accuracy was more than 99% for all cases, which also heightened the detection rate.

4.5 Performance of Confusion Matrix

Various metrics are employed to delineate the efficacy of a classifier. Table VII concisely encapsulates the four potential outcomes associated with a detection process.
A confusion matrix is a comparison between the true and predicted intrusion. The value of the predicted intrusion (cell value along the row of the confusion matrix) is normalized by the total number of true intrusions. The recall (i.e., the number of predicted values equal to the true values) of the model is the values along the main diagonal elements of the confusion matrix indicate. An ideal model would exhibit a "1" exclusively along the main diagonal and "0" elsewhere; the accuracy score is the number of correctly predicted instance

(sum of the diagonal terms) over the total number of instances (sum of the full matrix) as in following Figures:

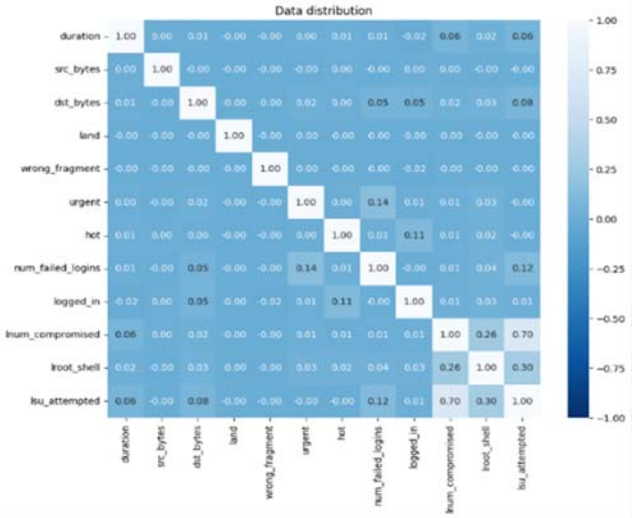


Fig. 7: Confusion Matrix of the Attacks distribution

Table VII: Confusion Matrix

		Actual	
		Normal	Attacks
Predicted	Normal	**True Positive**	False Positive
	Attacks	False Negative	**True Negative**

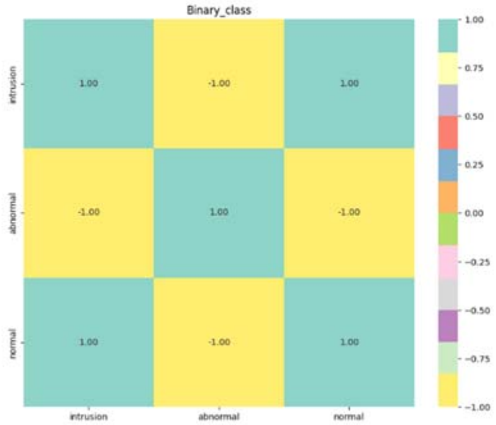


Fig. 8: Confusion Matrix of Binary labels

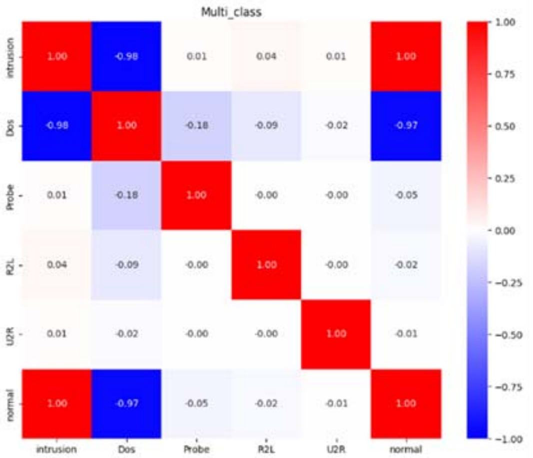


Fig. 9: Confusion Matrix of Multiclass labels

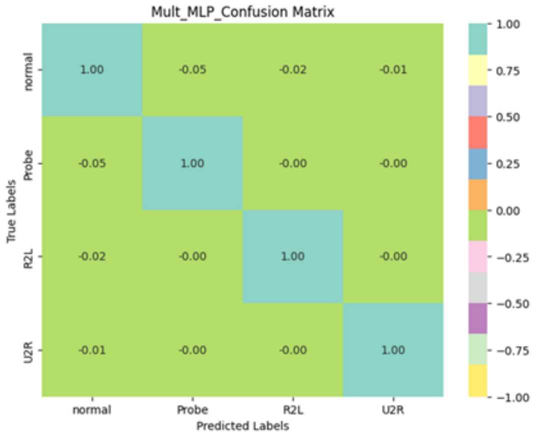


Fig. 10: Multiclass (MLP model)

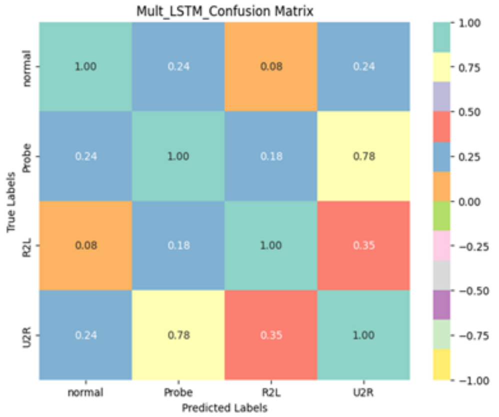


Fig. 11: Multiclass (LSTM model)

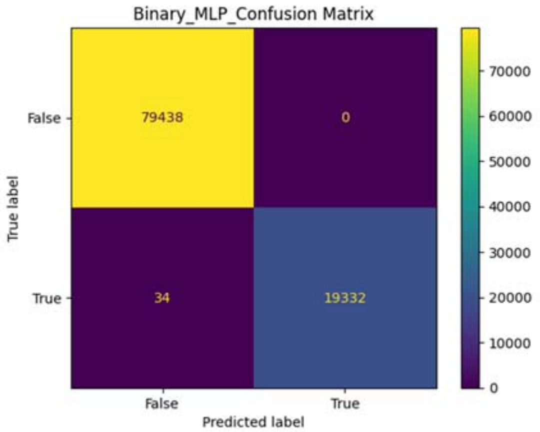


Fig. 12: Confusion Matrix of Binary MLP

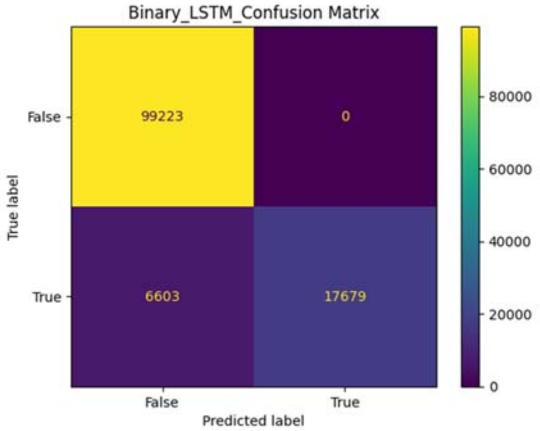


Fig. 13: Confusion Matrix of Binary LSTM

5. RESULT & DISCUSSION

The deep neural network necessitates numerous computations primarily involving matrix multiplications and additions. Therefore, employing a GPU is crucial to minimize the time spent on model training. In a MLP model at Binary Classification the effectiveness as; the average accuracy of 99.99352097511292 %, losses of 0.04522902600001544% and the time training is1008.4796767234802 seconds with the different hidden layer of our results are measured as in following figure:

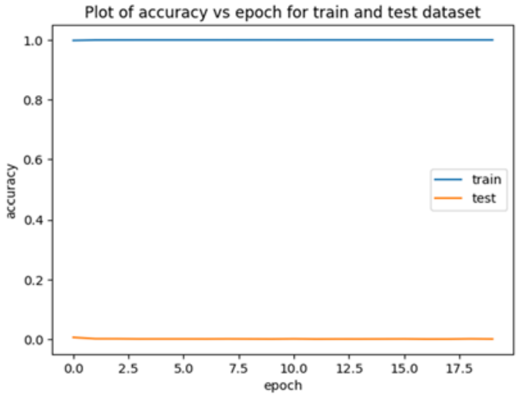


Fig. 14: plotting the accuracy vs epoch

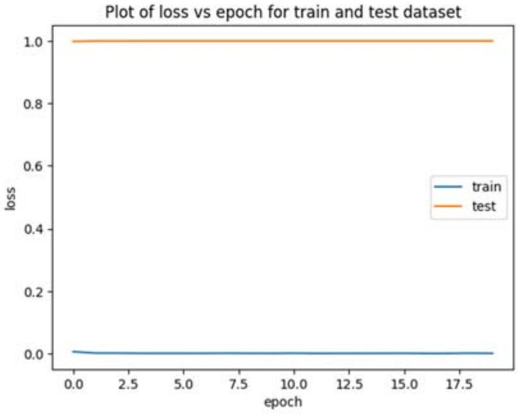


Fig. 15: plotting the loss vs epoch

In LSTM model (binary classification) the effectiveness as; the average accuracy of 99.83644485473633%, losses 0.7034842856228352% and the time training is 152.73029232025146 seconds with the different hidden layer of our results are measured as in following figures:

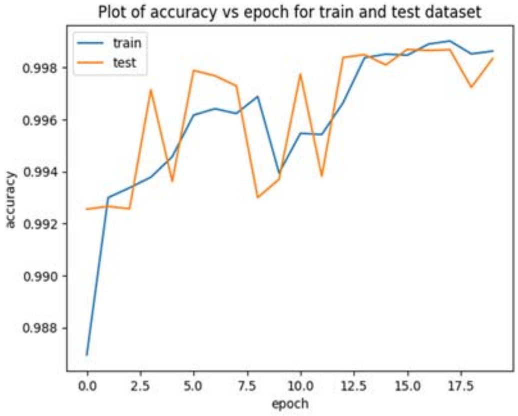


Fig. 16: plotting the accuracy vs epoch

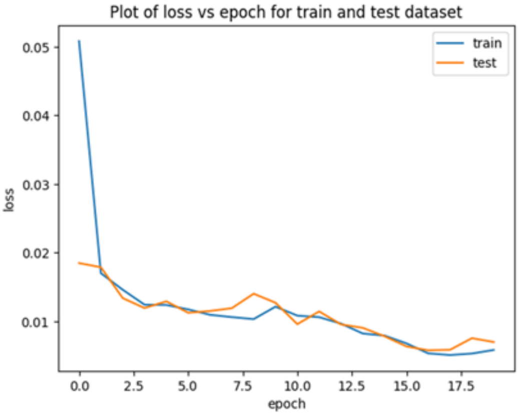


Fig. 17: plotting the accuracy vs epoch

while at MLP model in multiclass classification the effectiveness as; the average accuracy of 99.99271035194397 %, losses 0.06861906149424613%and the time training is 1057.5126361846924 seconds with the different hidden layer of our results are measured as in following figures:

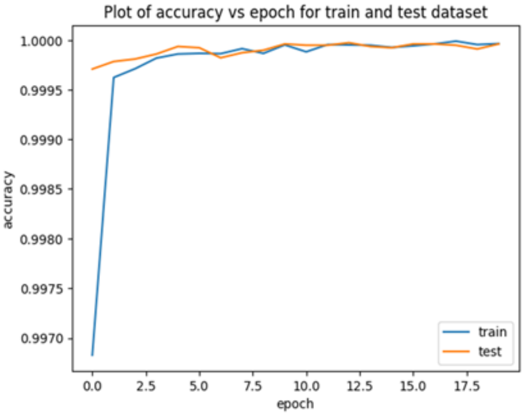


Fig. 18: plotting the accuracy vs epoch

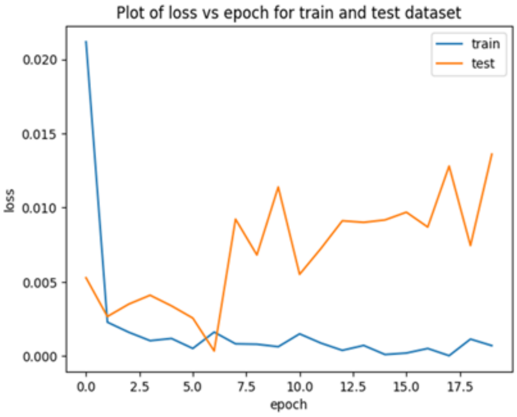


Fig. 19: plotting the accuracy vs epoch

In the LSTM model at multiclass classification the effectiveness as; the average accuracy of 99.99756813049316 %, losses 0.014561950229108334% and the time training is 150.2320737838745 seconds with the different hidden layer of our results are measured as in following figures:

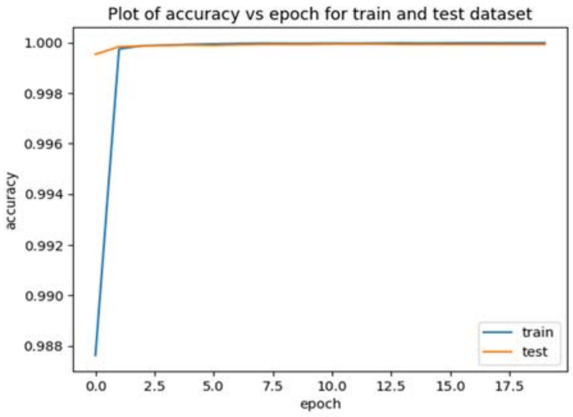


Fig. 20: plotting the accuracy vs epoch

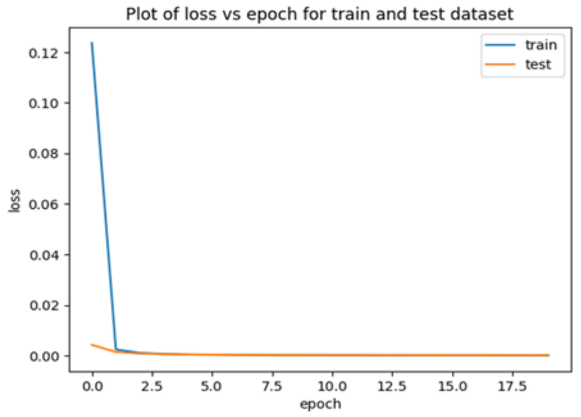


Fig. 21: plotting the Loss vs epoch

F1_ score used to get the equally weighted harmonic mean of precision and recall as in (14). This threshold gives a maximum score only if both false positives and false negatives are low, therefore finding the 'sweet spot'. Finally, we pick the threshold value with the highest F1_score.as shown in Tables (VIII), (IX).

The Receiver Operating Characteristic (ROC); curve illustrates the relationship between the true positive rate and the false positive rate, as depicted in Figures (22), (23). The numerical representation of the area under ROC curve signifies the likelihood that a positive instance chosen at random will be prioritized over a randomly selected negative instance.

To measure the effectiveness of a classifier that is able to distinguish between two classes: ROC-AUC and plot of TPR vs FPR. The area under carve close to 1 means that the classifier is good and the area close to 0.5 means that the classifier can't distinguish between the classes. A good classifier tries to maximize TP, TN and minimizing FP and FN. The ROC curve of MLP & LSTM as follows:

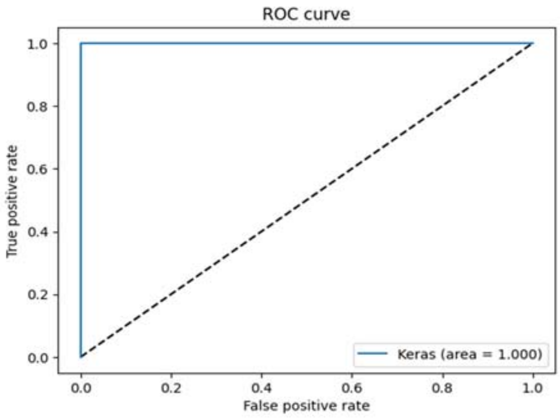


Fig. 22 The ROC curve of MLP model.

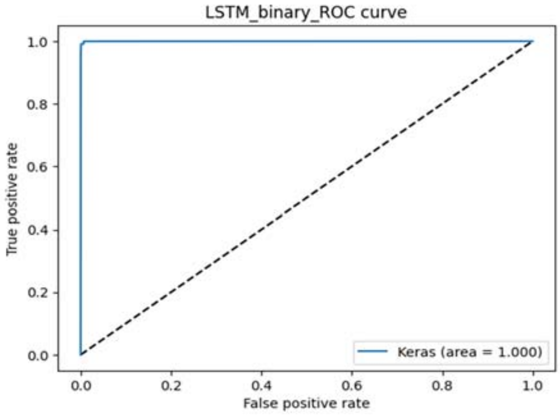


Fig. 23: The ROC curve of LSTM model.

6. CONCLUSION AND FUTURE WORK

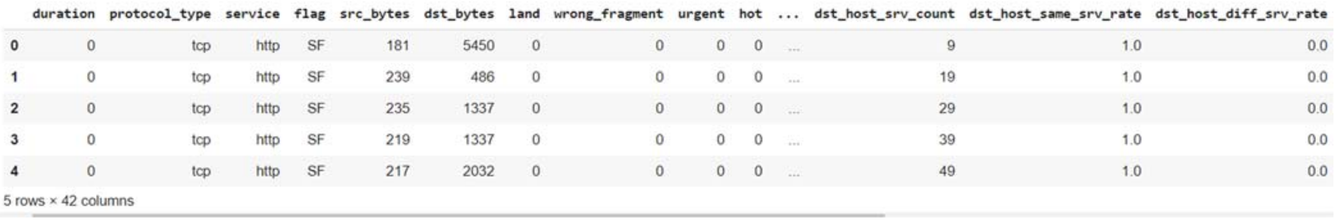
This paper is a study of intrusion detection systems with artificial intelligence using the DNN model. It used the KDD Cup_99 dataset for training and testing. The training data was created through data preprocessing. The dataset consisting of 80% of the reworked data was utilised as the training data.

The results of this study showcase remarkable performance metrics, with accuracy and detection rates consistently averaging an impressive 99.99%. Notably, the false alarm rate attained a remarkably low figure of 0.06%, underscoring the system's capability to discern normal data from potential attacks with exceptional precision. The Study in this domain has scrutinized and categorized individual traffic data.

However, as part of prospective endeavours, the exploration of real-time deep analysis of data emerges as a promising avenue. we underscored our trajectory by the pursuit of criteria facilitating faster decision-making processes, a concerted effort to mitigate computational complexity, and an orientation towards handling the challenges posed by big data in intrusion detection contexts.

For future work, we could use deep in real-time data, optimizing decision criteria, streamlining computational processes and addressing the complexities inherent in big data management.

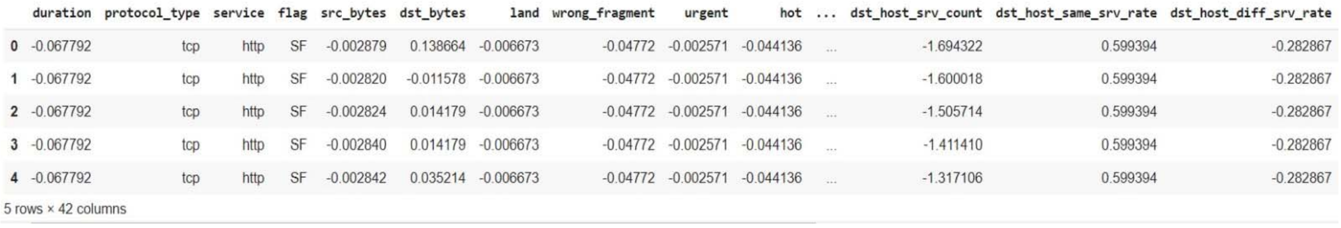
Table IV: Data before normalization



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	0.0
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	0.0
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	0.0
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	0.0
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	0.0

5 rows × 42 columns

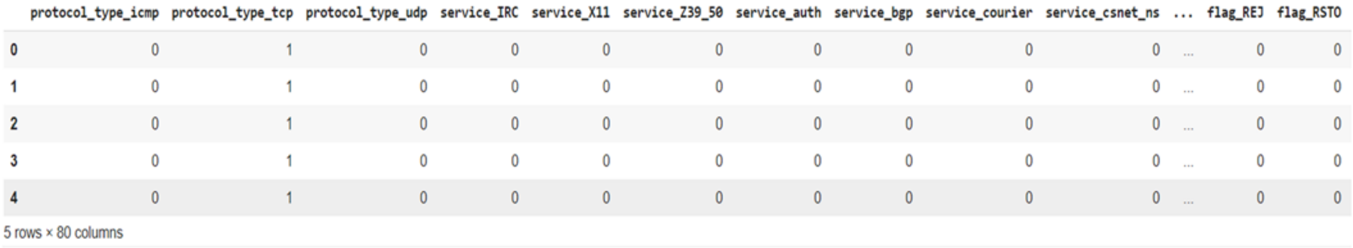
Table V: Data after normalization



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate
0	-0.067792	tcp	http	SF	-0.002879	0.138664	-0.006673	-0.04772	-0.002571	-0.044136	...	-1.694322	0.599394	-0.282867
1	-0.067792	tcp	http	SF	-0.002820	-0.011578	-0.006673	-0.04772	-0.002571	-0.044136	...	-1.600018	0.599394	-0.282867
2	-0.067792	tcp	http	SF	-0.002824	0.014179	-0.006673	-0.04772	-0.002571	-0.044136	...	-1.505714	0.599394	-0.282867
3	-0.067792	tcp	http	SF	-0.002840	0.014179	-0.006673	-0.04772	-0.002571	-0.044136	...	-1.411410	0.599394	-0.282867
4	-0.067792	tcp	http	SF	-0.002842	0.035214	-0.006673	-0.04772	-0.002571	-0.044136	...	-1.317106	0.599394	-0.282867

5 rows × 42 columns

Table VI: Convert categorical features to numerical using one-hot encoding



	protocol_type_icmp	protocol_type_tcp	protocol_type_udp	service_IRC	service_X11	service_Z39_50	service_auth	service_bgp	service_courier	service_csnet_ns	...	flag_REJ	flag_RSTO
0	0	1	0	0	0	0	0	0	0	0	...	0	0
1	0	1	0	0	0	0	0	0	0	0	...	0	0
2	0	1	0	0	0	0	0	0	0	0	...	0	0
3	0	1	0	0	0	0	0	0	0	0	...	0	0
4	0	1	0	0	0	0	0	0	0	0	...	0	0

5 rows × 80 columns

Table VIII: Schedule of performance of models (LSTM and MLP).
splitting the dataset 75% for training and 25% testing

Model	**LSTM**		**MLP**	
Class	**Binary**	**Multiclass**	**Binary**	**Multiclass**
Accuracy	99.83644485473633%	**99.99756813049316%**	99.99352097511292%	99.99271035194397%
Precision	0.9944152431011827	0.9943160776104983	0.999917620891342	**0.9999271249159912**
Recall (DR)	0.9972819372374598	**0.9999919031618153**	0.9997529033852236	0.9998785474272297
F1- score	0.9958465271209442	0.997145913650768	0.9998352553542009	**0.9999028355816101**
Loss	0.7034842856228352%	**0.014561950229108334%**	0.045229026000001544%	0.06861906149424613%
Time(s)	152.73029232025146	**150.2320737838745**	1008.4796767234802	1057.5126361846924

Table IX: Schedule of performance of models (LSTM and MLP).
splitting the dataset 80% for training and 20% testing

Model	**LSTM**		**MLP**	
Class	**Binary**	**Multiclass**	**Binary**	**Multiclass**
Accuracy	99.99089241027832%	**99.99696612358093%**	99.99493956565857%	99.98583197593689%
Precision	0.9995870541475249	0.9905063608384879	**1.0**	0.999858303897655
Recall (DR)	0.9999483631106062	**1.0**	0.9997418155530311	0.9998481842840371
F1- score	0.999767675985441	0.9952305406562414	**0.9998708911095618**	0.9998532440652406
Loss	0.030505415634252213%	**0.014984085282776505%**	0.03381400019861758%	0.46352697536349297%
Time(s)	**170.23747491836548**	184.8968095779419	1111.6987471580505	1111.741890668869

REFERENCE

[1] Gao, N., Gao, L., Gao, Q., & Wang, H. (2014). An Intrusion Detection Model Based on Deep Belief Networks. *2014 Second International Conference on Advanced Cloud and Big Data*. https://doi.org/10.1109/cbd.2014.41.

[2] K. Alrawashdeh and C. Purdy (2016) "Toward an online anomaly intrusion detection system based on deep learning," in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 195–200 https://ieeexplore.ieee.org/abstract/document/7838144.

[3] Hasan, M., Milon Islam, Md., Islam, I., & Hashem, M. M. A. (2019). Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches. Internet of Things, 7, 100059. https://doi.org/10.1016/j.iot.2019.100059.

[4] Rosay, A., Carlier, F., & Leroux, P. (2020). mlp4nids: an efficient mlp-based network intrusion detection for cicids2017 dataset. *machine learning for networking*, 240–254. https://doi.org/10.1007/978-3-030-45778-5_16.

[5] Churcher, A., Ullah, R., Ahmad, J., ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B., & Buchanan, W. J. (2021). An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks. *Sensors*, *21*(2), 446. https://doi.org/10.3390/s21020446.

[6] James Cannady, (1998) "Artificial neural networks for misuse detection," Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.

[7] K. Fox, R. Henning, J. Reed, and R. Simonian, (1990) "A neural network approach towards intrusion detection," Proceedings of 13th National Computer Security Conference, Baltimore, MD, pp. 125-134, 1990.

[8] H. Debar, M. Becker, and D. Siboni, (1992) "A neural network component for an intrusion detection system," Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, pp. 240 – 250, 1992.

[9] Srinivas Mukkamala, (2002) "Intrusion detection using neural networks and support vector machine," Proceedings of the 2002 IEEE International Honolulu, HI, 2002.

[10] R. Cunningham and R. Lippmann, (1999) "Improving intrusion detection performance using keyword selection and neural networks," Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Purdue, IN, 1999.

[11] Jin Kim, Nara Shin, Seung Yeon Jo and Sang Hyun Kim (2017). *Method of intrusion detection using deep neural network.* https://ieeexplore.ieee.org/abstract/document/7881684.

[12] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Sh, (2018) a deep learning approach to network intrusion detection, ieee transactions on emerging topics in computational intelligence, vol. 2, no. 1, february 2018.

[13] Cosimo Ieracitano, Ahsan Adeel, Francesco Carlo Morabito, Amir Hussain, (2019) A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach, Journal Pre-proof, 2019.

[14] Ayei E. Ibor, Florence A. Oladeji, Olusoji B. Okunoye, Charles O. Uwadia, (2019) deep learning model for predicting multistage cyberattacks, the journal of computer science and its applications vol. 26, no 1 june 2019.

[15] Kumar Sahu, Suraj Sharma, M. Tanveer, Rohit Raja, (2021)Internet of Things attack detection using hybrid Deep Learning Model, Computer Communications, Volume 176, 1 August 2021, Pages 146-154.

[16] Shewale, Y., Kumar, S., & Banait, S. (2023). Machine Learning Based Intrusion Detection in IoT Network Using MLP and LSTM. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(7s), 210–223. https://www.ijisae.org/index.php/IJISAE/article/view/2947/1525.

[17] KDD Cup 1999 Data. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data.

[18] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (n.d.). (2009) *A Detailed Analysis of the KDD CUP 99 Data Set.* https://www.ecb.torontomu.ca/~bagheri/papers/cisda.pdf.

[19] Somwang, P., & Lilakiatsakun, W. (2015). Anomaly Traffic Detection Based on PCA and SFAM. *The International Arab Journal of Information Technology*, *12*(3). https://www.iajit.org/PDF/vol.12%2Cno.3/6922.pdf.

[20] dhanushkumar_idk. (2023, september 8). classification algorithm. medium. https://medium.com/@danushidk507/classification-algorithm-13caad1742da.

[21] S. Hochreiter, J. Schmidhuber,(1997) Long short-term memory, Neural computation 9 (8) 1735–1780.

[22] artificial neural networks b. yegnanarayana. (n.d.). retrieved january 16, 2024, from https://content.kopykitab.com/ebooks/2016/06/7330/sample/sample_7330.pdf.

[23] Krishnamurthy, B. (2022, October 28). ReLU Activation Function Explained | Built In. Builtin.com. https://builtin.com/machine-learning/relu-activation-function.

[24] Types Of Activation Function in ANN.(2021,January20). GeeksforGeeks. https://www.geeksforgeeks.org/types-of-activation-function-in-ann.

[25] Basta, N. (2020, April 5). The Differences between Sigmoid and Softmax Activation Function. Medium. https://medium.com/arteos-ai/the-differences-between-sigmoid-and-softmax-activation-function-12adee8cf322.

[26] Gupta, A. (2021, October 7). A Comprehensive Guide on Deep Learning Optimizers. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers.

[27] Modasiya, K. (2022, November 28). What the heck is random_state? MLearning.ai. https://medium.com/mlearning-ai/what-the-heck-is-random-state-24a7a8389f3d.

[28] Brownlee, J. (2016, March 20). overfitting and underfitting with machine learning algorithms. machine learning mastery. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms.

[29] Singh, G. (2021, June 6). confusion matrix and cyber security.Nerd for Tech. https://medium.com/nerd-for-tech/confusion-matrix-and-cyber-security-62da9afb6eff.

NOTE: Required color printing.

Google Scholar

CERTIFICATE

OF PRESENTATION

SCIENCE CITE"

International Conference on Computer Engineering and Software Applications(ICCESA - 2024)

20th - 21st February 2024 | Virtual, Istanbul, Turkey

This is to certify that.. **HASAN HUSEYIN BALIK** ....affiliated with

.... Department of Computer Engineering Istanbul Aydin University Istanbul, Turkey ....has presented a paper titled

.... DETECTION OF THE NETWORK INTRUSION TRAFFIC USING DEEP LEARNING ....

....

at the conference organized by the Science Cite held on 20th - 21st February 2024 at Virtual, Istanbul, Turkey.

ResearchPEDIA.org

Dr. Samuel Moses
Director

SCIENCE CITE
SC
INTERNATIONAL CONFERENCE

Akash Shinde
Convener

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS