Weighted Multiclass Intrusion Detection System

Varsha Dange[1] , *Soham* Phadke[1*] , *Tilak* Solunke[1] , *Sidhesh* Marne[1] , *Snehal* Suryawanshi[1] and *Om* Surase[1]

Department of Multidisciplinary,
Engineering
Vishwakarma Institute of Technology Pune, India

Abstract. Attackers are continuously coming up with new attack strategies since cyber security is a field that is continually changing. As a result, it's important to update and enhance the system frequently to ensure its efficiency against fresh threats. Unauthorised entry, usage, or manipulation of a computer system or network by a person or programme is referred to as an intrusion. There are numerous ways for an incursion to happen, including using software flaws, phishing scams, or social engineering techniques. A realistic solution to handle the risks brought on by the interconnectedness and interoperability of computer systems is to use deep learning architectures to build an adaptive and resilient network intrusion detection system (IDS) to identify and categorise network attacks. Artificial neural networks (ANNs) or deep learning can help adaptive intrusion detection systems (IDS) with learning capabilities identify well-known and unique or zero-day network behavioural patterns, which can significantly reduce the risk of compromise. The NSL-KDD dataset, which represents both synthetically manufactured attack actions and real-world network communication activity, is used to show the effectiveness of the model. Model trained with this dataset to detect a wide range of attack patterns, which help in building an effective IDS.

Keywords: Deep Learning, Decision Tree, KNN, Naive Bayes, Random Forest.

1 Introduction

To monitor network traffic for any signs of malicious or suspicious activity, a security tool known as an intrusion detection system (IDS) is utilised. Its objective is to identify potential network assaults or security breaches and notify security staff. IDS analyses network traffic and then compares the findings to a database of recognised attack signatures and suspicious activity. The system generates an alarm and tells security personnel to take appropriate action if it detects traffic that resembles a known attack signature or displays indicators of anomalous behaviour.

Systems based on signatures and anomaly detection are the two most important categories of IDS.A signature-based IDS compares network traffic to a database of recognised attack signatures or patterns. It triggers an alarm if it discovers traffic that resembles a known attack signature. On the other hand, anomaly-based IDS examines network data and establishes a baseline of acceptable behaviour. Any traffic that departs from that norm is therefore flagged as potentially suspicious.

*Soham Phadke: chintamani.soham21@vit.edu

Some IDS can be set up to automatically respond to threats by blocking malicious traffic or isolating offending systems from the network, for example. An essential piece of security technology, intrusion detection systems help organisations identify and respond to potential network assaults and security lapses. Businesses may respond swiftly to suspected breaches and identify risks in real-time by keeping an eye on network traffic for odd activities. Both internal and external intrusions are possible. Internal intrusions happen when an authorised user abuses their access privileges or when a compromised account is used to access systems or networks, whereas external intrusions happen when a hacker from the outside manages to obtain unauthorised access to a system or network. For businesses, intrusions can have detrimental effects such as sensitive data theft, operational disruption, financial loss, and reputational damage. Systems for identifying and reducing the risk of intrusions, preserving the security and integrity of computer networks, and preventing intrusions are all crucial instruments. There may occasionally be overlap between normal and abnormal behaviours since it is difficult to tell one from the other. Intelligent intrusion detection systems are useful in this situation. These systems are a more accurate and effective kind of intrusion detection since they study network traffic patterns using machine learning techniques and identify any unusual conduct. IDS works in conjunction with procedures that examine and keep track of various network events to find instances of malicious behaviour and security issues of any kind. It serves as a detecting mechanism on the network and is positioned outside of the real-time communication spectrum. Use SPAN or TAP ports to examine copies of inline network packets to make sure any incoming communication is not malicious.

Therefore, setting up an IDS program allows the system to:

- Analyse network packets to spot attack patterns
- Observe how users behave.
- Determine the unusual traffic behaviour Make sure that user and system behaviour does not violate security guidelines.

Host-based intrusion detection systems (HIDS), network-based intrusion detection systems (NIDS), anomaly-based intrusion detection systems (AIDS), and signature-based intrusion detection systems (SIDS) are the four categories into which IDS can be divided.

A network intrusion detection system (NIDS) and keeping track of network traffic packets at various points for any potential intrusions or anomalies may use both hardware (sensors) and software (console). Network intrusion detection systems (NIDS) continuously examine network traffic for known attack patterns, port scans, and unauthorised access attempts. They can be placed inside a network to monitor internal traffic or at important network points like the perimeter.

Host intrusion detection system (HIDS) Conversely, keep an eye on what's happening on specific hosts, such servers or workstations, and search for any indications of infiltration, like logins, file alterations, and system calls. HIDS only monitors activity on one particular computer or server, known as a host, at a time. Despite being limited to a single system, it offers more capabilities than NIDS due to its ability to access encrypted information that travels across the network, including file attributes, registries, and system configuration databases.

In order to detect intrusions, **anomaly-based intrusion detection systems (AIDS)** monitor network traffic. Security engineers and specialists develop a baseline to evaluate the

protocols, bandwidth, ports, devices utilised, and other aspects of the corporate network's health.

Signature-based intrusion detection systems (SIDS)work by keeping an eye on and detecting signatures on data packets travelling via your network. IDS tools check data packets against previously encountered/drawn attack signatures or against known harmful attack qualities in a database.

There are numerous techniques for intrusion detection in network systems. However, deep learning's popularity has increased due to its ability to handle enormous amounts of data and extract complex patterns from it. Deep learning-based intrusion detection and prevention systems (IDPS) can identify novel and unidentified attack types by analysing network traffic and system logs. The major objectives of IDPS are to respond quickly to invasions and to deliver a full study of intrusion detection, intrusion forms, and their detection. Data collection, feature selection, and conversion are the three core IDPS components and the decision engine Data collection comprises gathering logs, network traffic, and system event data related to network and system activity. Feature selection/conversion entails choosing pertinent features from the gathered data and turning them into a decision engine-friendly format. The decision engine examines the data and determines whether an intrusion has taken place and what steps should be taken. Deep learning techniques, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, have been used in IDPS to improve the accuracy of intrusion detection and prevention. These algorithms may distinguish between good activity and bad behaviour by extracting complex patterns and features from the data.

2 Literature Review

The research suggests a random forest-based intrusion detection method for power system networks. The random forest decision tree is produced using the power system network intrusion subsample. The edge function improves the random forest model. The effectiveness of the random forest algorithm-based intrusion detection solution for power systems has been proven through experiments. Utilising the power system network's lowest state vector and calculating the measurement residual of the attack, the accuracy of the approach is evaluated. The results indicate a strong capability to detect network intrusions in the power system. [1] An unsupervised learning approach to the intrusion detection problem was investigated and shown using the NSL KDD dataset. The sensitivity and cost of a hierarchical agglomerative clustered SOM network were examined. One choice is to take into account a different clustering method for SOM networks, such as fuzzy c-means. [2]

Numerous supervised, unsupervised, and hybrid models have demonstrated strong accuracy, precision, recall, and F1 scores when it comes to detecting IoT breaches. Deep learning algorithms are excellent at spotting unexpected attacks, handling massive data volumes, and adapting to evolving attack patterns, surpassing traditional intrusion detection methods such as rule-based and anomaly-based IDS.Intrusion detection in the Internet of Things (IoT) can be made far more effective and efficient by incorporating deep learning techniques. Deep learning algorithms are a useful way for IoT intrusion detection, the authors

concluded.[3]. The technology was very accurate in identifying network intrusions, 99.73%. Based on their findings, the authors concluded that the system is adept at detecting network breaches and holds the promise of enhancing network security.[4]

They proposed a technique that, in terms of F1-score, accuracy, and precision, outperformed traditional IDS systems. Leveraging deep learning in their IDS design holds the potential to improve network security. However, when developing IDS systems for real-world applications, it is essential to consider the balance between detection precision and false alarm rates. [5] When compared to other conventional intrusion detection systems, the system obtained greater accuracy, precision, and F1-score. Less false alarms were generated by the system. CNNs and LSTMs were very effective at detecting network intrusions. The experiments' outcomes showed how effectively the system's techniques operated.[6] The NSL-KDD dataset and a network security mechanism called BAT were employed. The approach included dataset preprocessing, utilising feature selection techniques, and using the BAT model to distinguish between regular and abnormal network traffic. The results of the study demonstrate that the BAT model surpasses other state-of-the-art deep learning models in terms of accuracy, precision, and recall rates. The model exhibits a minimal false positive rate and demonstrates the capability to detect various types of network threats. [7]

The research introduces a hybrid framework for utilising deep learning in intrusion detection for healthcare systems. To find anomalies, a stacked autoencoder is used in conjunction with supervised and unsupervised methods. The framework is evaluated using a healthcare-related dataset. This study uses the NSL-KDD dataset and a network security mechanism named BAT. The method uses feature selection techniques, dataset preparation, and the BAT model to differentiate between typical and aberrant network traffic. The research outcomes demonstrate that the BAT model surpasses other advanced deep learning models concerning accuracy, precision, and recall rates. Moreover, The model has a low false positive rate and can recognise many kinds of network threats. [7]

According to the research, a hybrid approach can be used to implement deep learning-based intrusion detection in healthcare systems. To find anomalies, a stacked autoencoder is used in conjunction with supervised and unsupervised methods. The framework is evaluated using a healthcare-related dataset. They compared the dataset they used with the publicly available.[8] The research suggested a hybrid approach for deep learning-based intrusion detection in healthcare systems. It used momentum to prevent overfitting, specified dropout rates, and "categorical cross entropy" with the Adam optimisation method. On the UNSW-NB15 dataset, the approach achieved a high accuracy of 95.6% for the 25% testing dataset. However, As the identification rate for underrepresented classes was lower, there is space for improvement in resolving class imbalance. [9]

The methodology employed in the presented study comprises classifying packets as legitimate or malicious based on the data content utilising four classifiers (RF, DT, SVM, and MLP). Numerous classifiers were assessed in a study using the NSL-KDD dataset, and it was discovered that the Random Forest (RF) classifier produced the highest accuracy, exceeding 99%, especially when utilising the first feature subset. Moreover, the study demonstrated that lowering the amount of features in the dataset enhanced the computational

efficiency and model accuracy, particularly for RF and DT classifiers[10]. The use of deep learning techniques in intrusion detection systems (IDS) is thoroughly examined and assessed in this study. In a detailed mapping effort, the scientists looked at 114 articles that were released between 2012 and 2018. They found that IDS has made substantial use of deep learning, and that this has led to positive outcomes in terms of recognising various forms of attack. The review of the current state of the field, outstanding research topics, and potential future research orientations forms the report's concluding section. [11].

When compared to traditional machine learning techniques, the evaluation findings demonstrate that the suggested IDS model employing NDNN (Nested Deep Neural Network) achieves a high detection rate while maintaining low False Detection Rate (FDR) and Missed Alarm Rate (MAR). The NDNN-based IDS outperforms other machine learning algorithms in terms of overall performance. It is demonstrated that the suggested DNN-based IDS is a workable paradigm for intrusion detection, particularly in multi-feature datasets.[12]. The study offers a novel stacked NDAE (Nested Denoising Autoencoder) and RF (Random Forest) classification algorithm-based NDAE (Nested Denoising Autoencoder) strategy for unsupervised feature learning. This method showed a reduction in training time and an accuracy improvement of up to 5%.The proposed classifier was assessed and implemented in TensorFlow, which makes use of graphics processing units (GPUs), using the benchmark KDD Cup '99 and NSL-KDD datasets. [13]In this study, a feature extraction method based on wrappers was created and combined with a wireless network data intrusion detection system (IDS).According to the results, the FFDNN algorithm performed better than other techniques in terms of binary classification accuracy, achieving 94.34% on validation data and 87.48% on test data. Additionally, the WIFE-FINN method outperformed conventional machine learning (ML) strategies in binary and multiclass classification contexts. The simulations showed that the suggested method produced outstanding overall accuracy rates for binary classification of 99.66% and multiclass classification of 99.77%. [14]

The researchers outlined a variety of performance indicators, including F-measure, recall, accuracy, and precision. In order to comprehensively examine the performance effectiveness of their low-dimensional features in both binary and multiclass classification, they carried out a series of tests using the NSL-KDD dataset. They purposefully contrasted the performance of their suggested approach throughout their research with that of other widely used methodologies, such as naive Bayesian, RF, and others [15].

Their study's main objective was to identify malicious behaviour in IoT networks, and to that end, they cleverly created a complex deep neural network (DNN) with an amazing 20 hidden layers. The DNN demonstrated exceptional accuracy, surpassing the remarkable threshold of 90%, and was remarkably adept at accurately detecting aggressive behaviour in the network under analysis. In their investigation, they thoughtfully incorporated three distinct data sets for training purposes, ensuring robustness in their model's performance, and expertly established separate groups for conducting rigorous testing and validation processes [16].

The comprehensive findings arising from their extensive research endeavour revealed that the combined utilisation of BGRU and MLP exhibited unparalleled effectiveness in accurately identifying and classifying a multitude of assaults, especially good

at spotting probing and denial of service (DoS) attacks. Moreover, the system that adopted the (B)GRU + MLP configuration greatly benefitted from accelerated convergence rates, culminating in expedited and more efficient computations. It is important to note, however, that Attacker to root (U2R) and remote to local (R2L) attack detection did not produce the best results due to the low number of records available and the RNN-based system's intrinsic affinity for effectively handling time-series jobs. [17].

The researchers demonstrated a novel and extremely efficient intrusion detection technique based on dimensionality-reduced convolutional neural networks (CNN), showcasing its impressive capability for precisely identifying and analysing network intrusion data. With an impressive accuracy rating of 94.0%, a detection rate of 93.0%, and an astonishingly low false alarm rate of 0.5%, the suggested model exhibited exceptional performance across various evaluation metrics, underscoring its potential for robust real-world applications [18].

The proposed model underwent extensive testing across various phases, each of which produced consistently high accuracy results, and It currently boasts a remarkable 99.62% overall accuracy rate and a remarkably low false positive rate of 1.57%. Phase 1 of the model managed to reduce data analysis by an impressive 19.69%, while achieving an accuracy rate of 99.81%. Similarly, Phase 2 boasted an impressive accuracy of 99.5%, supported by a staggering 504 leaves, and Phase 3 achieved a commendable accuracy rate of 99.11% [19].

The researchers used feature engineering in conjunction with the CNN model to creatively reduce the dimensionality of the data in the context of wireless intrusion detection utilising WLAN data. The outcomes were nothing short of exceptional, with the proposed method outshining other existing models across various evaluation metrics, including accuracy, precision, recall, and F1 score. The model's accuracy, precision, recall, and F1 score all attained noteworthy values of 98.8%, 98.9%, and 98.8%, respectively. Consequently, the proposed approach demonstrated its unparalleled effectiveness, particularly in the realm of wireless network security, and conveyed tremendous potential for real-world applications [20].

The researchers used a variety of deep learning models to rigorously study numerous datasets in their pursuit of improving intrusion detection, including well-known datasets like NSL-KDD, UNSW-NB15, and CICIDS2017. Drawing from their extensive analysis, the researchers firmly established that deep learning models, with their inherent adaptability and capacity for extracting intricate patterns and features, significantly outperformed conventional machine learning algorithms in accurately detecting network breaches. The overarching implications were clear—deep learning methods stood as a more robust and reliable alternative in the domain of cybersecurity, ensuring improved accuracy and precision in identifying potential security threats [21].

Additionally, the authors conducted a thorough comparison of multiple CNN models, meticulously evaluating diverse CNN architectures, including VGG, Inception, and ResNet. Among these, the VGG16 model emerged as the undisputed champion, impressively

achieving an accuracy rate of 96.08% on the test set, solidifying its position as a formidable performer in the domain of intrusion detection [22].

As the literature review went on, the datasets NSL KDD, KDD Cup 99, and UNSW-NB15 came out on top in terms of being widely used. The majority of research publications predominantly employed conference papers as the primary form of disseminating findings. Interestingly, only a limited number of attempts were made to harness ensemble learning on datasets such as Honeypot, CAIDA, and AWID, indicating potential areas for further exploration and research [23].

Delving further into the domain of encrypted communications, the study introduced an innovative deep learning method christened DeepPacket, specifically tailored to classify encrypted data across various communication models, including SSL, SSH, and VPN. The experimental findings were awe-inspiring, with an average accuracy of 98%, presenting enormous potential for diverse applications, including network security, traffic engineering, and service quality control [24].

In their research, the team proposed an intricately designed 5-layered architecture, meticulously tailored to effectively detect intrusion in vast datasets. Recurrent neural networks with long short-term memory (RNNBiLSTM) were used in both directions, and the architecture showed hitherto unheard-of accuracy rates, sensitivity ratios, and specificity ratios. The findings demonstrated the system's remarkable efficacy in fortifying IoT networks against potential security breaches, ensuring the integrity and safety of the network infrastructure [25].

Further enriching their research portfolio, the team devised a trustworthy and adaptive anomaly detection framework model, which effectively managed automatically deployed application-level firewalls. The proposed method demonstrated astounding testing accuracy rates across various datasets, with values reaching an impressive 99.51% and 99.91% [28]. This was accomplished by leveraging the strengths of the Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) as well as machine learning techniques like the Support Vector Machine (SVM) and K-Nearest Neighbour (K-NN).

In exploring the effectiveness of Principal Component Analysis (PCA) in enhancing detection precision, the authors skillfully conducted two tests, yielding promising results that showcased the potential of PCA to bolster precision in specific scenarios [29].

The extensive survey paper offered a comprehensive overview of various intrusion detection approaches, emphasising the need for developing a universal technique capable of securing networks effectively across diverse environments. The pursuit of such a technique remains an imperative aspect for strengthening the resilience of network security [30].

3 Methodology/Experimental

3.1 System Architecture

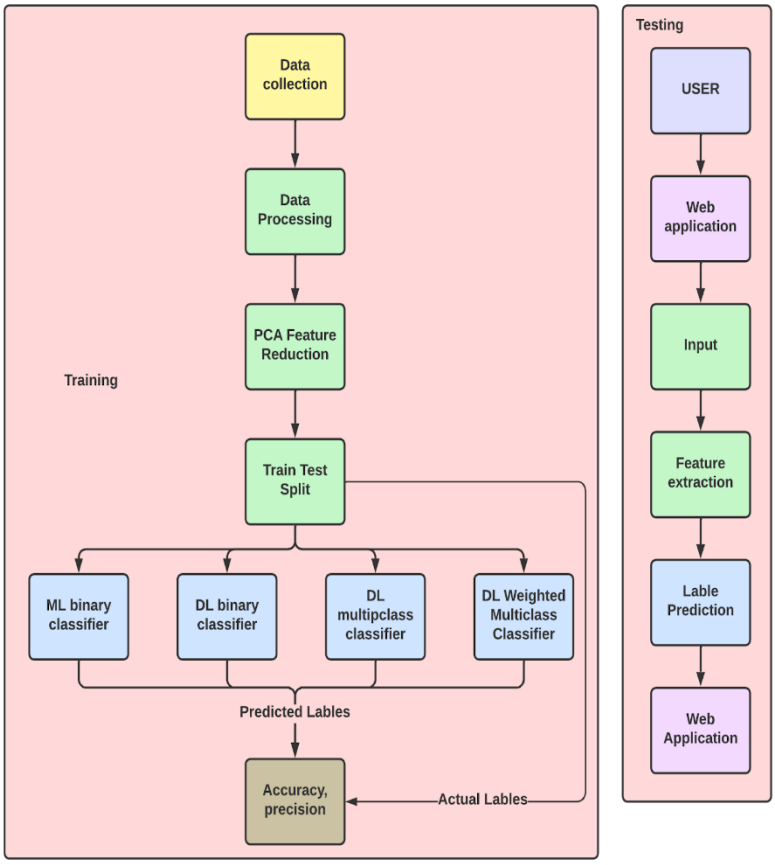

Fig. 1. System Architecture of intrusion detection system

Data collection for this research was conducted using the NSL-KDD dataset. In the field of cybersecurity, the NSL-KDD dataset is a frequently used benchmark dataset for evaluating intrusion detection systems. It is a revised version of the original KDD Cup 1999 dataset, intended to address some of the biases and shortcomings of the latter. This dataset is made up of network traffic data, which includes many sorts of network attacks as well as routine activity. The study used this information to design and test an effective intrusion detection model capable of differentiating between benign network traffic and malicious attacks.

Data processing involved removing null values from the dataset, splitting numeric and categorical data and then scaling the numeric variables. The encoding technique used to

encode the categorical data was one hot encoding and the output label was also encoded using label encoding.
Principle component analysis was used for feature reduction. PCA is a statistical method for reducing the number of dimensions in high-dimensional data by selecting the features that best represent the dataset. The features are chosen based on the variance they produce in the output. The first principal component is the trait that causes the most variation. The characteristic that accounts for the second biggest variance is regarded as the second principle component, and so on. It is vital to note that the primary components have no relationship with one another.

3.2 **Components**

TensorFlow:
It is a complete open-source machine learning platform. Google Brain created it, and it is extensively used in both research and industry. TensorFlow provides a flexible architecture that allows you to deploy computations on different hardware, such as CPUs, GPUs, or specialized hardware like TPUs (Tensor Processing Units).It offers a rich ecosystem of tools, libraries, and resources to aid in the execution of diverse machine learning tasks. TensorFlow uses a dataflow graph model to represent computations and allows you to define, optimise, and execute complex mathematical operations efficiently.

Keras:
Built on TensorFlow (as well as other deep learning libraries like Theano and Microsoft Cognitive Toolkit), Keras is a high-level neural network API. To create and train deep learning models, Keras offers a user-friendly and straightforward interface. It allows you to quickly prototype and experiment with different network architectures and configurations. Keras offers a rich set of pre-built layers, loss functions, optimizers, andvarious elements that facilitate the creation and training of neural networks.Recurrent neural networks (RNNs), convolutional neural networks (CNNs), and other sophisticated features are supported as well.

Matplotlib:
Matplotlib is a popular Python charting package. It includes a variety of tools for constructing different sorts of visualisations, such as line plots, scatter plots, bar graphs, histograms, and more. Matplotlib is highly customizable, allowing you to control every aspect of the plot, such as colors, labels, titles, and axes. Matplotlib is very customisable, enabling you to change the colours, labels, titles, and axes of the plot. Matplotlib provides a pyplot interface that is similar to MATLAB, making it easy to get started with plotting in Python.

Flask:
Flask is a lightweight web framework for Python. It allows you to build web applications and APIs quickly and easily. Flask follows the WSGI (Web Server Gateway Interface) specification and can be used with any web server that supports WSGI, such as Gunicorn or uWSGI. Flask provides essential features for web development, such as routing, request handling, template rendering, and session management. It also supports extensions for adding

additional functionality, such as database integration, authentication, and form handling. Flask is often used for building small to medium-sized web applications or as a foundation for larger web frameworks.

3.3 Machine Learning Algorithms

K-Nearest Neighbours:
It is a non-parametric, instance-based machine learning approach. Regression and classification issues are addressed by it. Based on the class or value of the data point's k closest neighbours in the training dataset, it predicts the class or value of a new data point. KNN is suitable for a variety of applications because it is straightforward and intuitive and doesn't require explicit model creation during training. On the other hand, the choice of hyperparameter k can significantly affect performance, and the computational complexity may be a problem for large datasets. KNN is used in pattern recognition, recommender systems, anomaly detection, medical diagnosis, and other areas.

Decision Tree:
A well-known and simple to understand machine learning approach called a decision tree can be applied to both classification and regression problems. Each core node represents a feature-based judgement, and each leaf node represents a predicted class label or value, resulting in a tree-like model. During training, the algorithm divides the input recursively based on the most important features, seeking to maximise information gain or minimise impurity at each stage. This approach generates a flowchart-like structure with clear and explicit decision rules, allowing Decision Trees to be easily interpreted and visualised. They can, however, suffer from overfitting, particularly on large datasets, which can be minimised by pruning approaches or ensemble methods such as Random Forests.

Naive Bayes:
A probabilistic machine learning technique called Naive Bayes is based on the Bayes theorem. In light of the occurrence of another event, it calculates the likelihood of one event occurring. The Naive Bayes model is simplified and more effective calculations are made possible by the "naive" assumption that all features are independent of one another. The technique evaluates the probabilities of distinct class labels during training based on the frequency of feature occurrences in the training data. When new data is introduced, it computes the likelihood of each class label for that data point and assigns the label with the highest probability as the predicted class. Naive Bayes is very useful for text classification issues like spam detection and sentiment analysis. It's also popular in natural language processing, document classification, and recommendation systems. Though it does not capture intricate feature interactions, Naive Bayes is computationally efficient, simple to construct, and can serve as a robust foundation model for many classification tasks.

Random Forest:
The machine learning ensemble learning method known as Random Forest (RF) is frequently applied in classification and regression applications. It is a collection of decision trees, each

of which was trained using a randomly chosen feature set and subset of the training data. The RF combines the predictions from all the trees throughout the prediction phase to arrive at a final prediction, either by voting (for classification) or averaging (for regression). Random Forest's main strength is its capacity to avoid overfitting and increase generalisation due to the randomization inherent in data sampling and feature selection. It can handle big and high-dimensional datasets, is less sensitive to noisy data, and offers useful information on feature relevance.

3.4 Model Architecture

Traditional machine learning based classification algorithms have an underlying probability model using which classification is done. These models are effective only when the underlying conditions and assumptions of the probability model are satisfied. This user is required to have good knowledge about the properties of data and that of the model before its application.

Neural Networks are an effective alternative to machine learning based models. Since they are naturally self-adaptive, they can learn the characteristics of the data without needing to be explicitly specified. They are non-linear models, which makes them versatile in simulating complicated interactions between group members and object properties in the real world. On the other hand, neural networks based classification models(ANN classification models) have been successfully applied to a variety of real world classification problems in the field of business, science and industry and have shown promising results and ability to work with more skewed data.

Internally the mathematical function inside a neuron is given by:

$$F(x) = w.X + c \tag{1}$$

where M is the number of input attributes and X is a 1D tensor with size M*1. c is the bias, and w is a weight matrix of size 1*M. The output of these neurons is passed through a ReLu function. This is a special type of function that output the input value if it is greater than 0, in all other cases its output is 0.

An ANN based classification model consists of various such neurons which take in the input attributes and give out the output label. A loss function is then used to determine the loss once this label is compared to the actual label. Cross entropy loss function was employed in this study. The weights and bias of the neural network are then calibrated using the loss value from the loss function to get results that are more accurate.

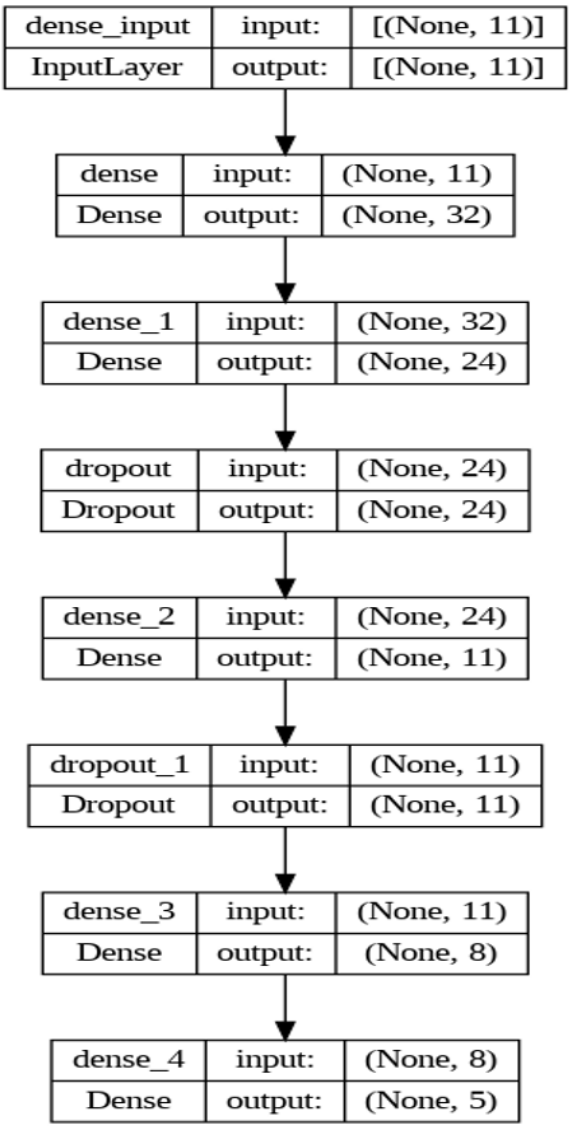

Fig. 2. Neural Network Architecture

Model Architecture:
The model is named "sequential," which implies that the layers are stacked sequentially.
The model consists of six layers: dense, dense_1, dropout, dense_2, dropout_1, and dense_3, followed by the final output layer dense_4.

Layer Details:
dense: This is a fully connected layer (Dense) with 32 output units/neurons. The output shape of this layer is (None, 32), indicating that it produces a tensor of shape (batch size, 32).
dense_1: Another fully connected layer with 24 output units. The output shape is (None, 24).
dropout: Dropout is a regularisation strategy that prevents overfitting by randomly setting a percentage of input units to 0 during training. The output shape is identical to the preceding layer. (None, 24).
dense_2:a layer with 11 output units and full connectivity. The final form is (None, 11).
dropout_1: Another dropout layer. The output shape remains the same as the previous layer, (None, 11).
dense_3: a layer with eight output units and full connectivity. The final form is (None, 8).
dense_4: The final output layer with 5 output units. The output shape is (None, 5).

Parameter Count:
The parameter count shows how many trainable parameters there are in the model. It represents the learnable weights and biases within the layers.
All of the model's 1,592 total parameters in this instance are trainable.

Trainable vs. Non-trainable Parameters:
The weights and biases that are adjusted during training in order to reduce the loss function are referred to as trainable parameters.
Non-trainable parameters refer to any other internal parameters that are not updated during training.There are no non-trainable parameters in this model.

In summary, this model is a sequential neural network with several fully connected layers and dropout regularization. It takes input data and processes it through the layers to generate predictions. The model has a total of 1,592 trainable parameters, which are updated during training to optimize its performance on the given task.

In case of Weighted Multiclass classifier model the underlying ANN model is same, the only modification made to it was that each class was assigned a weight to it using the following equations

$$W_{class1} = (1/F_{class1}) * (total/2.0) \quad (2)$$

Here, W stands for class 1 weight, and F for class 1 frequency. The training dataset's total observation count is represented by the term total. This weighted strategy was investigated here because it was found that the training dataset's amount of observations of each type of attack is significantly unbalanced. Thus, to have proper representation of classes with low number of observations in the learning of the classifier, weights were attached to the classes. Thus, class having lower number of observations was given higher priority in the training phase.

4 Results And Discussion

NSL-KDD is the utilised dataset. It was created using statistics from the KDD Cup 1999, which was created based on network traffic data from a simulated environment. The dataset contains 125,973 records each of which represents a network connection or an instance. The dataset is labelled, with each instance belonging to one of the following classes:

- Normal: Represents normal network connections or instances.
- DOS (Denial of Service): Represents instances of denial-of-service attacks.
- Probe: Represents instances of probing attacks.
- U2R (User-to-Root): Represents instances of unauthorised access to superuser privileges.
- R2L (Remote-to-Local): represents incidents where a distant machine gained unauthorised access to a local machine.

The dataset includes a total of 41 features, representing various characteristics of the network connections or instances.

Machine learning methods like k-Nearest Neighbours, Deep neural networks, decision trees, random forests, and Gaussian naive bayes were used. Dimensionality reduction was accomplished using principal component analysis. There were 122 original features and 20 reduced features in total.

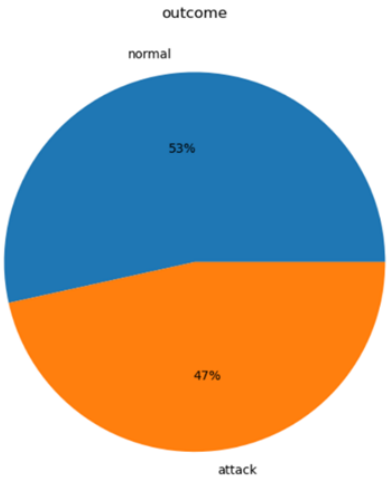

Fig. 3. The outcome class division

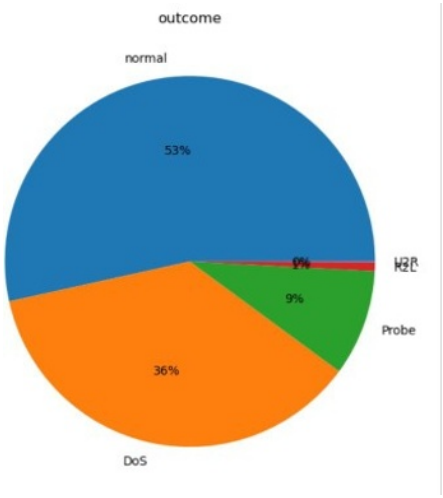

Fig. 4. Distribution of different attacks and normal observations in the dataset.

Machine learning algorithms and a multilayer perceptron were used for binary classification of attack and normal type.

Table 1. Performance comparison table

Algorithm	Accuracy	Precision	Recall
KNN	98.93%	99.08%	98.64%
Naive bayes	91.60%	90.86%	89.29%
Decision Tree	98.606%	98.66%	98.35%
Random Forest	98.90%	99.31%	98.33%
Deep NN	97.25%	98.02%	97.17%

The evaluation parameters like loss and accuracy were measured and they were used to analyse the model.

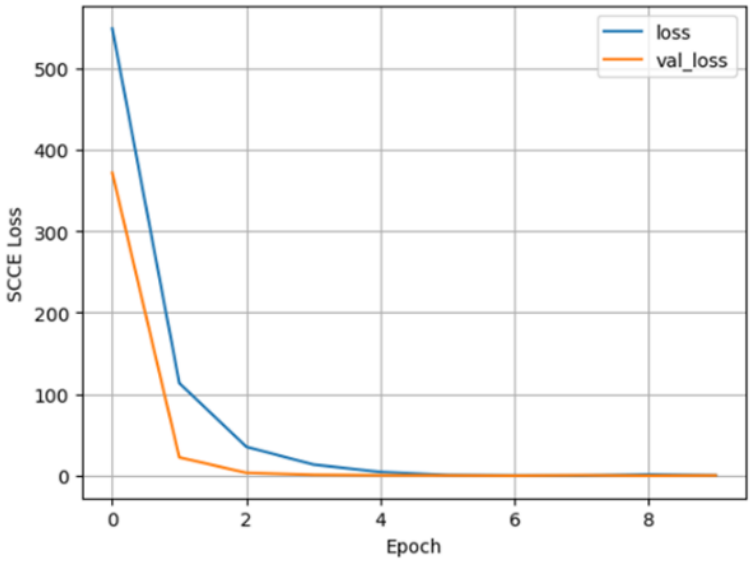

Fig. 5. loss vs validation loss for neural network

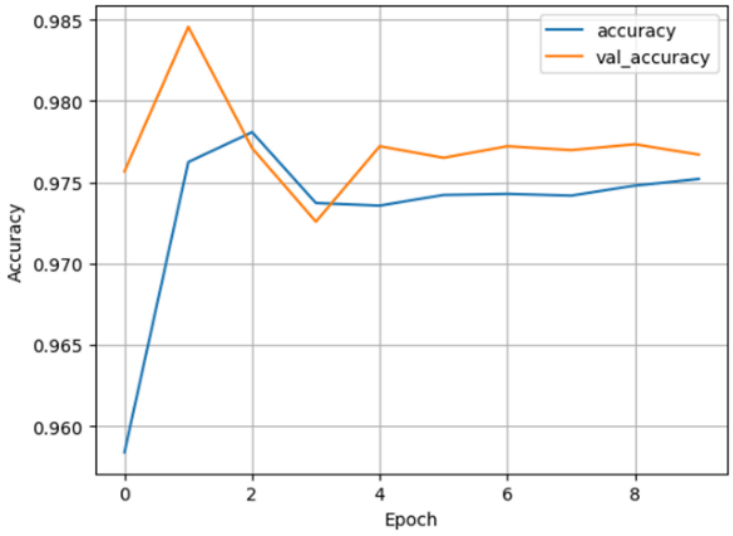

Fig. 6. accuracy vs validation accuracy for neural network

In the above figures we can see that the loss, both training and validation significantly comes down and the accuracy is also increased hence indicating that the model is a good model.

Multiclass classification included a deep neural network without weights and one neural network weights. It was found that there was no significant difference between both the neural networks.

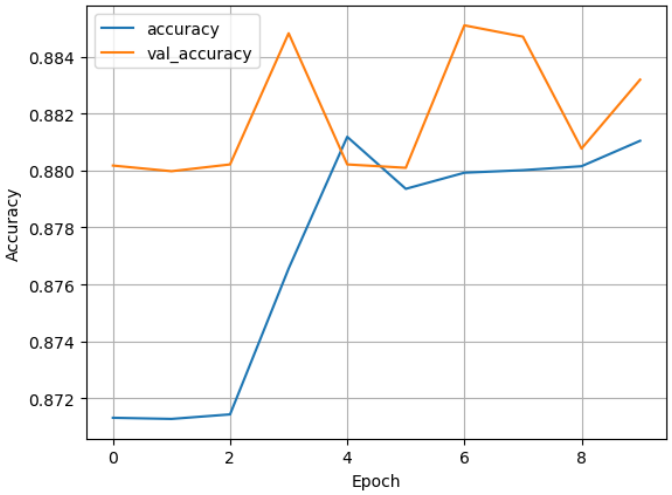

Fig. 7. non-weighted multiclass loss

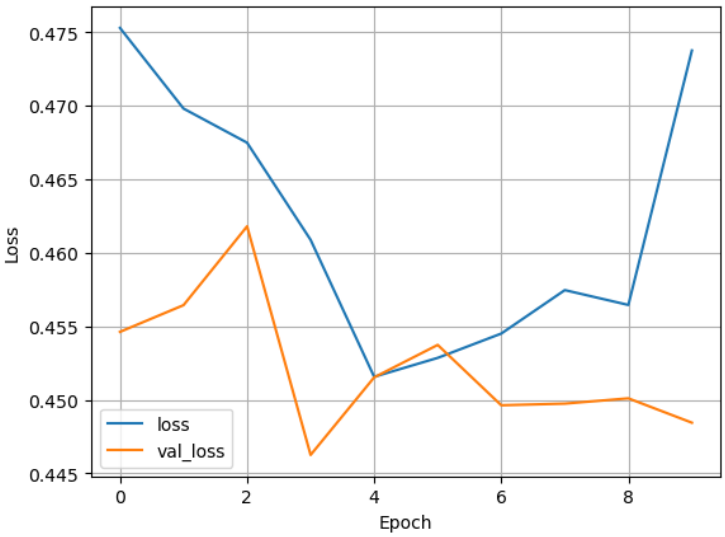

Fig. 8 non-weighted multiclass accuracy

In the above figures we can see that the loss is very volatile and the accuracy is lower than the accuracy we observed in the binary classification neural network. This demonstrates that multiclass classification is less trustworthy than binary classification.

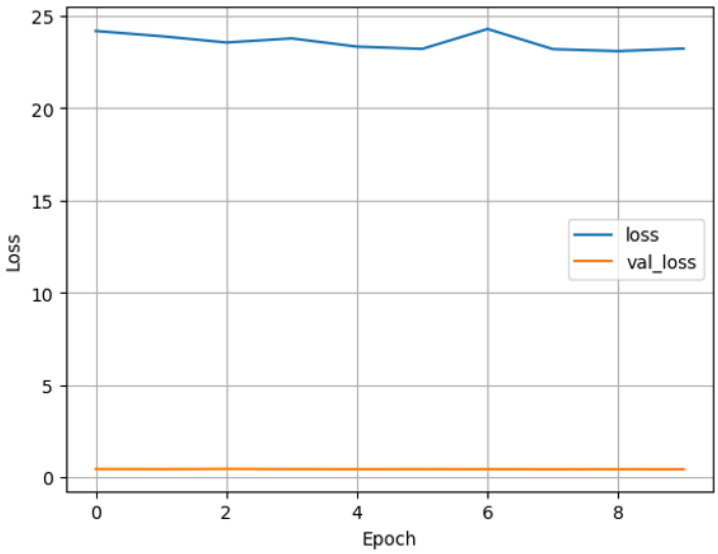

Fig. 9. weighted multiclass loss

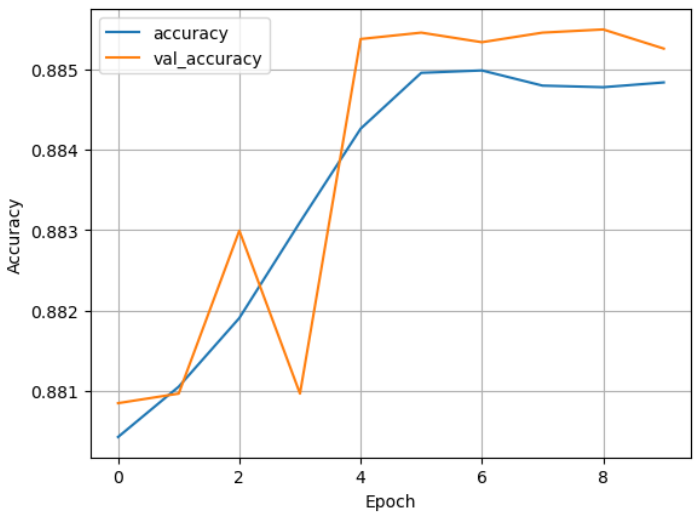

Fig. 10. weighted multiclass accuracy

Table 2. Multiclass classification comparison

Algorithm	Accuracy	Loss
Deep NN without weights	87.98%	0.4412
Deep NN with weights	88.54%	0.3664

The above weighted classification using neural network, it is observed that the difference between validation loss and training loss is high and also the overall training loss is extremely high which indicates that the model is not a very good performer and that the non-weighted model performs better than the weighted neural network.

5 Conclusion

Our study's objective was to develop an intrusion detection system based on deep learning that could identify different types of network attacks in real-time. We used a real-time dataset of 42 variables to collect attack information, and we used an artificial neural network (ANN) to train the deep learning model on this input data. The objective of our system was to identify common network traffic patterns and identify any outliers that would point to potential network attacks.

Using the Flask server, we created a web application that could handle real-time data and swiftly evade particular kinds of classified network attacks. It's crucial to remember that our technology cannot identify an attack's specifics; it can only determine whether incoming network information is normal or odd. In order to recognise and stop potential attacks, deep learning models have been applied in intrusion detection systems, which has been found to improve network security. The deep learning model's potential for future improvement could increase attack detection's accuracy and fortify the system's defences against network intrusions.

References

1. G. ZHU, H. YUAN, Y. ZHUANG, Y. GUO, X. ZHANG and S. QIU, "Research on network intrusion detection method of power system based on random forest algorithm," 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 2021, pp. 374-379, doi: 10.1109/ICMTMA52658.2021.00087.
2. Lin, E., Chen, Q. & Qi, X. Deep reinforcement learning for imbalanced classification. Appl Intell 50, 2488–2502 (2020). https://doi.org/10.1007/s10489-020-01637-z
3. B. R, S. Deepajyothi, P. G, D. T, P. Karthikeyan and V. S, "Survey on Intrusions Detection System using Deep learning in IoT Environment," 2022 International Conference on Sustainable

Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 195-199, doi: 10.1109/ICSCDS53736.2022.9760993.

4. Lirim Ashiku, Cihan Dagli,Network Intrusion Detection System using Deep Learning, Procedia Computer Science,Volume 185, 2021, Pages 239-247, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.(https://www.sciencedirect.com/science/article/pii/S1877050921011078)
5. Fernandez, Gabriel & Xu, Shouhuai. (2019). A Case Study on Using Deep Learning for Network Intrusion Detection.
6. S. Al-Emadi, A. Al-Mohannadi and F. Al-Senaid, "Using Deep Learning Techniques for Network Intrusion Detection," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 171-176, doi: 10.1109/ICIoT48696.2020.9089524.
7. T. Su, H. Sun, J. Zhu, S. Wang and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," in IEEE Access, vol. 8, pp. 29575-29585, 2020, doi: 10.1109/ACCESS.2020.2972627.
8. Akshay Kumaar M, Samiya D, Vincent PMDR, Srinivasan K, Chang CY, Ganesh H. A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning. Front Public Health. 2022 Jan 12;9:824898. doi: 10.3389/fpubh.2021.824898. PMID: 35096763; PMCID: PMC8790147.
9. Lirim Ashiku, Cihan Dagli,Network Intrusion Detection System using Deep Learning, Procedia Computer Science,Volume 185, 2021, Pages 239-247, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.(https://www.sciencedirect.com/science/article/pii/S1877050921011078)
10. A. S. Ahanger, S. M. Khan and F. Masoodi, "An Effective Intrusion Detection System using Supervised Machine Learning Techniques," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1639-1644, doi: 10.1109/ICCMC51019.2021.9418291.
11. S. Osken, E. N. Yildirim, G. Karatas and L. Cuhaci, "Intrusion Detection Systems with Deep Learning: A Systematic Mapping Study," 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 2019, pp. 1-4, doi: 10.1109/EBBT.2019.8742081.
12. Lirim Ashiku, Cihan Dagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185,2021, Pages 239-247,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.
13. N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.
14. Sydney Mambwe Kasongo, Yanxia Sun, A deep learning method with wrapper based feature extraction for wireless intrusion detection system, Computers & Security, Volume 92, 2020, 101752, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.101752.
15. M. Al-Qatf, Y. Lasheng, M. Al-Habib and K. Al-Sabahi, "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection," in IEEE Access, vol. 6, pp. 52843-52856, 2018, doi: 10.1109/ACCESS.2018.2869577.
16. Sarika Choudhary, Nishtha Kesswani, Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT, Procedia Computer Science, Volume 167, 2020,
17. C. Xu, J. Shen, X. Du and F. Zhang, "An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units," in IEEE Access, vol. 6, pp. 48697-48707, 2018, doi: 10.1109/ACCESS.2018.2867564.
18. Y. Xiao, C. Xing, T. Zhang and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 42210-42219, 2019, doi: 10.1109/ACCESS.2019.2904620.

19. K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilising support vector machines, decision trees, and naive Bayes for off-line analysis," SoutheastCon 2016, Norfolk, VA, USA, 2016, pp. 1-6, doi: 10.1109/SECON.2016.7506774.
20. S. M. Kasongo and Y. Sun, "A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System," in IEEE Access, vol. 7, pp. 38597-38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
21. Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, Helge Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study, Journal of Information Security and Applications, Volume 50, 2020, 102419, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2019.102419.
22. N. Khoza, M. Khosa, T. Mahlangu and N. Ndlovu, "Plant Seedling Classification Using Machine Learning," 2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2022, pp. 1-6, doi: 10.1109/icABCD54961.2022.9856067.
23. Bayu Adhi Tama, Sunghoon Lim, Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation, Computer Science Review, Volume 39, 2021, 100357, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2020.100357.
24. Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R. et al. Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput 24, 1999–2012 (2020). https://doi.org/10.1007/s00500-019-04030-2
25. JEYANTHI D, Indrani B. IoT Based Intrusion Detection System for Healthcare Using RNNBiLSTM Deep Learning Strategy with Custom Features. Research Square; 2022. DOI: 10.21203/rs.3.rs-2302072/v1.
26. JOUR, Uddin, M. Irfan, Alkahtani, Hasan, Aldhyani, Theyazn H. H. 2021 2021/07/07 Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms 5579851 2021 1076-2787 https://doi.org/10.1155/2021/5579851
27. N. Ahuja, G. Singal and D. Mukhopadhyay, "DLSDN: Deep Learning for DDOS attack detection in Software Defined Networking," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 683-688, doi: 10.1109/Confluence51648.2021.9376879.
28. Alzahrani, Mohammed Yahya, Alkahtani, Hasan, Aldhyani, Theyazn H. H., Al-Yaari, Mohammed 2020 2020/12/10 Adaptive Anomaly Detection Framework Model Objects in Cyberspace 6660489 2020 1176-2322 https://doi.org/10.1155/2020/6660489 10.1155/2020/6660489
29. Nskh, Praneeth & Varma, M & Naik, Roshan. (2016). Principle component analysis based intrusion detection system using support vector machine. 1344-1350. 10.1109/RTEICT.2016.7808050.
30. U. Bashir and M. Chachu, "Intrusion detection and prevention system: Challenges & opportunities," 2014 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2014, pp. 806-809, doi: 10.1109/IndiaCom.2014.6828073.