Design of a Hypermodel using Transfer Learning to Detect DDoS Attacks in the Cloud Security

Marram Amitha, Dr.Muktevi Srivenkatesh
Department of Computer Science, GITAM Deemed to be University, Visakhapatnam, India

***Abstract*—The present research proposes a detective approach to analyzing the performance of various algorithms used for more accurate detection of Distributed Denial-of-Service (DDoS) attacks in cloud computing. From the start, this study uses machine learning and deep learning to explore whether information security has evolved in recent years. The deployment of intrusion detection systems and distributed denial-of-service attacks are then discussed. The most common DDoS attack types were summarized. In addition, this study reviewed the existing approaches and techniques for DDoS attack detection. Various pre-processing subsystems as well as attribute-based selection techniques for preventing the detection of DDoS were briefly described. The proposed Intrusion detection system uses transfer learning for detecting DDoS attacks in the Networks. The proposed system used for the data set for the Network Intrusion Detection System is SDN Dataset which has more features and is suitable to use to detect in Network Intrusions. It contains 23 features that are used to detect Intrusions in the network SDN Dataset which consists of training and testing data to detect the attacks in the network. The detection and prevention subsystems through ML and DL strategies were briefly discussed. The proposed deep learning model for DDoS attack detection in cloud storage applications is explained. After that, various preprocessing strategies employed in the detection are described, among them rebalancing data, data cleaning, data splitting, and data normalization like min-max normalization. The author created a hypermodel that consists the parameters of baseline classifiers like Support Vector Machine, K-Nearest Neighbors Algorithm, XGboost, and other various machine learning models. The proposed model gives very good accuracy compared to other machine learning models.**

Keywords*—Machine learning; deep learning; support vector machine; k-nearest neighbors algorithm

I. INTRODUCTION

Intrusion relates to every collection of connected operations performed by a malevolent adversary that affects a target system. Regarding the detection of DoS and DDoS attacks, intrusive activities are usually adaptable and can be categorized based on the attacks.

The primary threat from these four intrusion activities involves a DoS and DDoS attack that either consumes computer and communication facilities or takes advantage of the system's vulnerabilities to make the system accessible for authorized users. This leads to a significant loss of resources, money, and data. When numerous systems overwhelm the internet connection of a targeted system, which includes one or more web servers [1], a Distributed Denial-of-Service (DDoS) assault occur. Such an attack usually arises with traffic overloading of the targeted system resulting from many compromised systems. Distributed Denial of Service attacks are different from different kinds of attacks in that can carry conduct a damaging attack on Internet-connected resources.

A. Phases of DDoS Attack

Two phases are followed in DDoS attacks. The attacker seeks to gain access to the network's vulnerable machines first. With the compromised hosts of other networks, the attacker or master establishes his own network and describes them as slaves. The 'intrusion phase' occurs when that occurs. The attacker then selects which victim server to target and starts delivering packets in that direction. With DoS attacks, which start from a single host, DDoS attacks originate from a number of dynamic networks that were previously hacked. The 'DDoS attack phase' [2] corresponds to what this is recognized as.

Based on the way methods work, DoS and DDoS attacks can be roughly classified into three categories (Ali et al., 2019). These involve assaults based on connection utilization, bandwidth consumption, and vulnerability exploitation.

B. Connection Consumption-based Attacks

A connection-oriented protocol is the Transmission Control Protocol (TCP). In advance of data exchange, it establishes a connection between the client and the server. A limited quantity of connection requests can be accepted and processed by any server. In an effort keep those with authorization considering accessing the service offered by the organization, the attacker establishes an enormous amount of connections with the server. The operating system's kernel [5] resources required for setting up connections have been drained by this type of attack. One of the most frequently used attacks under this category is the SYN Flood attack.

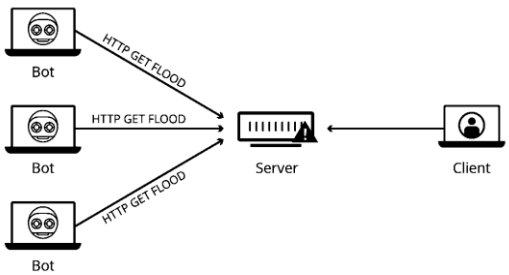

Fig. 1. Flood attack.

Fig. 1 shows the process of a SYN Flood attack, during which the attacker establishes an extensive amount of TCP connections with connections only partially accessible to deplete the connection pool. Due to how it operates, a cluster

of servers may be configured to function slower than normal through using slow network connections.

C. Bandwidth Consumption-based Attacks

Every network has an established amount of accessible bandwidth. The network's bandwidth limit has been exceeded, which will impact up the response times of the servers and the various devices linked to it. Attacks based on computation-based DoS and DDoS are started by making use of this primary bandwidth. In order to generate a massive flood, the attacker uses established handler machines to control an enormous quantity of preconfigured zombie devices connected to the internet. User Datagram Protocol (UDP) flood is a prevalent attack in this type of attack. Fig. 2 shows the visualization of continuous features with respect to packet count protocol and type of attack.

This paper explains the procedure of transfer learning to train a model to get higher accuracy. In this research, authors used A.DDoS attack SDN Datasets from Mendeley website.

II. Related Work

Deep learning is gaining popularity these days because of its higher accuracy and performance. Its implementations in this field are being researched by a community of scholars. Automotive architecture, healthcare, manufacturing, and law enforcement are some of the well-known realms. The study that has already been completed by various scholars is mentioned below. Asad et al. [2020] [1] equate the effects of the machine learning methodology to those of others The Naive Bayes categorization methodology and the decision-tree categorization methodology are two examples of machine learning techniques. The researcher mostly attempted to act in a version that was not online. If the scale of the dataset grows larger, the output disparity becomes more pronounced. Deep Intelligence was introduced by Bhuvaneswari Amma N.G et al [2023] [2]. The knowledge was derived using a radial base function with a variety of abstraction levels. The research was conducted on well-known datasets such as NSL-KDD and UNSW NB15 that included 27 functions. In comparison to other existing methods, the researcher believed that his method was more accurate. Muhammad Aamir et al. used a clustering technique to apply a feature selection process. Five related machine-learning methodologies were used to compare the algorithm. For preparation, SVM and RF methodologies were utilized. The best accuracy was attained by RF, which was about 96 percent.

Mishra et al. [3] classified packages depending on their characteristics. Through inspecting the IP header, the protection strategy attempts to identify IP addresses. These IP addresses are utilized to distinguish between spoofed and legitimate addresses. As the scale of the assault becomes larger, firewalls are ineffective. For separating the regular and assaulted traffic, Narasimha et al utilize anomaly identification and machine learning methodologies. Real-time datasets were utilized in the research. For classification, the well-known naive Bayes ML methodology was utilized. The outcomes were compared to those of other methodologies such as J48 and RF.

A. Haddaji et al [2023] [4] combined intellectual-stimulated computation and the entropy methodology in their research. For the classification, Support Vector Machine Learning was utilized. The platform's flow chart was being mined for information. In terms of identification precision, the outcomes were satisfactory. Omar E. Elejla et al. used an IPv6 classification strategy to incorporate a methodology for the identification of DDoS assaults. The findings were compared to 5 different well-known machine learning methodologies by the scientist. DT, SVM, NB, KNN, and NN were the methodologies utilized. The research was conducted on a well-known dataset. According to the source, the KNN methodology achieved a precision of about 85 percent.

Farhan ulla et al. [2023] [5] used the ML methodology to create an entropy-depend semi-supervised methodology. Unsupervised and supervised designs are used in this development. Unsupervised techniques have high precision and low false-positive rates. Supervised methods, contrarily, minimize the number of false positives. The datasets utilized in the research were NSL-KDD, UNB ISCX 12, and UNSW-NB15. For the recognition of the assault, Nathan Shone et al. use a DL methodology. It also utilized the NDAE function for unsupervised instruction. On the well-known KDD Cup 99 and NSL-KDD datasets, the proposed methodology was executed on a GPU utilizing TensorFlow. The researcher believed that he was able to get more precise identification outcomes.

III. Intrusion Detection System Model

The Internet is a global network of computers connected through various media and a standard protocol. Among many additional essential elements in modern life, people nowadays depend on the World Wide Web for their education, trade, social ability, and recreational activities. Evidently, the Internet brought perhaps the greatest advances in communication and computing.

Attacks on the web may involve in many different possible dangers, such as financial loss, identity theft, loss of confidential data or information, theft of network resources, damage to a person's brand and reputation, and a decrease in consumer confidence in online banking and e-commerce.

Most security issues differentiate themselves from their earlier equivalents in non-network infrastructures since data and business logic are located on a remote Network server with transparent supervisors. One of these attacks, the Denial of Service (DDoS) attack, has been extremely aggressive and extremely intrusive to web servers. Denial of Service (DDoS) assaults frequently target the server a network of computers that provide consumers a service. DoS attackers seek to consume servers that are operational in an approach that leads the service stop functioning because of an excessive number of outstanding requests in the service queue.

DDoS attacks can be performed on government departments, educational institutions, and home systems that have been hacked. These computer programs are referred to as bots. Usually, DoS assaults begin at the network layer by sending many UDP, SYN, or ICMP packets of data. Attackers migrate to the application layer and flood it with HTTP GET requests, which is referred to as application-layer DDoS, after

network layer attack fails. According to Bhardwaj et al. (2020) [6], DDoS attacks can use TCP SYN, UDP flood, DNS reflection, HTTP flood, and ICMP flood. Based on information research on security, distributed denial-of-service (DDoS) attacks recently cost companies and governments all through the world a large amount of revenue. On the other side, the attackers use more advanced techniques to amplify attacks and overload their targets by taking benefit of their geographic distribution and computing power provided possibly by the wide variety of devices and their various movements, which are frequently incorporated within IoT network scenarios. Therefore, a practical and effective DDoS detection method must be developed and has been optimized, and functional in IoT-based smart environments with major constraints on processor power, reaction time, and data processing volume. Thus, conventional IDSs might out to be completely appropriate for applications in the Internet of Things. IoT security is a continuously significant issue that requires the creation of mitigation techniques and an increasing understanding of IoT safety concerns system (Catak FO et al., 2020) [7]. In order to guarantee that client data is stored in a safe fashion, security is another crucial component of network-based IoT data. Network security is crucial for both commercial and personal users. Everyone wants to remain certain regarding the integrity of their personal data. Businesses are legally required to protect customer information, and certain industries require more stringent regulations around data storage. Various difficulties with the value multi-tenancy, data loss and leaking, network accessibility, identity management, harmful APIs, inconsistent service levels, patch management, and internal threats are associated with the issue of network computing security. Some safety features that cannot be sufficiently scalable, incompatible, and appropriate are missing from conventional basic cryptographic algorithms. With these requirements in mind, an IDS mechanism with the indicated encryption approach has been established to defend the network from DDoS attacks.

A. DDoS Attack SDN Dataset

Machine learning and deep learning algorithms use this smaller net emulator-generated data set that has been modified with SDN, to categorize traffic. At the start of the project, ten smaller networks with switches connected to a single Ryu controller are established. Network simulation was utilized for imitating malicious traffic, such as TCP Syn assaults, UDP flood incidents, and ICMP assaults, as well as normal traffic including TCP, UDP, and ICMP. A total of 23 features in everything, some of which have been determined and some of which were obtained from the switches, in the data set. Among of the characteristics that have been obtained are Switch-id, Packet_count, Byte_Count, Duration_sec, Duration_nsec, which is Duration in Nanoseconds, Source IP, Destination IP, and Total Duration. The port symbol while rx_bytes indicates the number of bytes received on the switch port, tx_bytes indicates the number of bytes carried from the switch port. The dt field shows the time and date prior to the are transformed to values, and a flow is tracked every 30 seconds. Packet per flow examines the entire amount of packet in a single flow, while bytes per flow examines the total amount of bytes in a single flow. The data transfer and reception accelerates are tx_kbps and rx_kbps, respectively, whereas port bandwidth is the product of tx_kbps and rx_kbps. The packet rate, which is determined as the number of packets sent per second, can be determined by dividing the number of packets provided per flow by the monitoring interval, Packetins message measure, and the total number of flow entries in the switch. The last column's category proof of identity, which decides whether the traffic type is malicious, is shown. Label 1 indicates malicious traffic, while Label 0 indicates benign traffic. The result of a 250-minute network simulation produced 1,04,345 rows of statistics. Repeating the simulation for a longer amount of time allows for the gathering of more data.

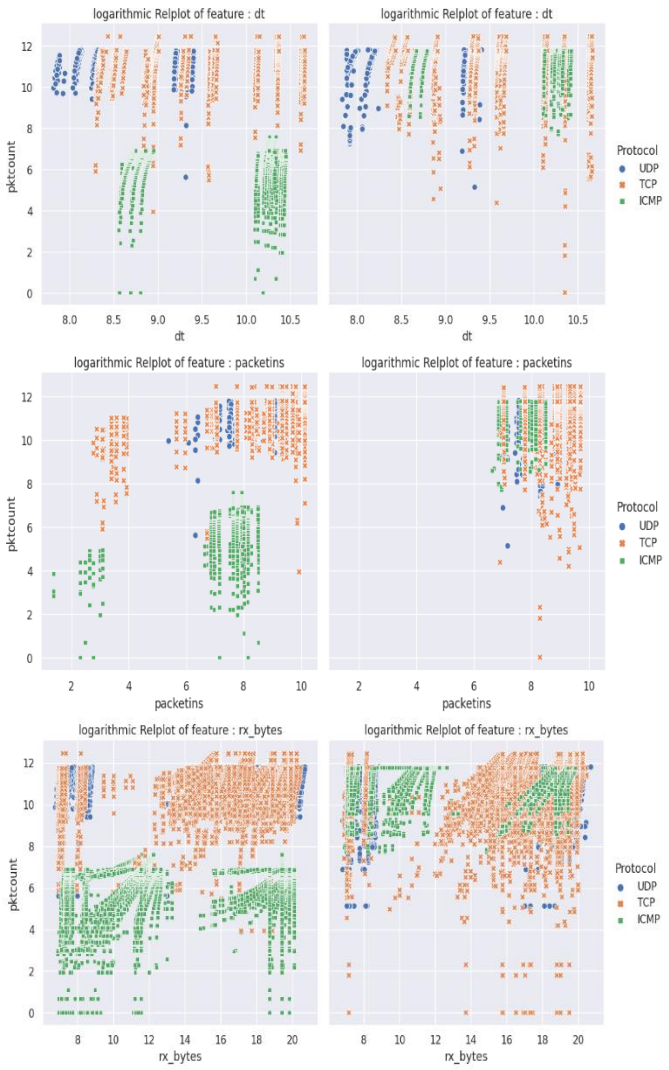

Fig. 2. Visualization of continuous features with respect to packet count, protocol, and type of attack.

IV. PROPOSED METHODOLOGY

A sophisticated model based on DLNN which employs optimized values in the hidden layers to differentiate among normal and attacked network data is developed for recognizing DDoS attacks.

A. Deep Learning Neural Network

One of the most advanced artificial intelligence methods for tackling computer vision's 11 challenges is the Deep Convolutional Neural Network (Deep CNN). As a feed-forward artificial neural network, the Deep CNN established a class of deep learning and has been utilized for different agricultural image classification research. Deep CNN's convolutional layer, which is vital, employs filters to extract data from the input images. An enormous amount of training data must be collected for the purpose of enhancing the performance of Deep CNN. Fig. 3 shows the architecture of the Deep CNN technique.

One of the primary advantages of using Deep CNN for image classification is avoiding the requirement for feature development. Deep CNN's numerous levels each contain multiple convolutions. They provide many kinds of visuals for the training data in the quicker, more detailed layers, acquiring to more intricate ones in the deeper layers. The pooling layers, which initially serve as methods to extract features from the convolutional layers' performance as feature extractors, are then used to decrease the dimensionality of the training data. In the words of Chen J et al. (2019) [8], the convolutional layers transform an assortment of lower-level features into additional discriminative features. The vital elements of Deep CNN are the convolutional layers in addition. In contrast to traditional machine learning, feature engineering is an essential part of deep learning. The down-sampling process gets carried out along the spatial dimensions through the pooling layer. It promotes having fewer parameter choices. The pooling component of the proposed model uses the max-pooling process. In the proposed Deep CNN model, max pooling outperforms average pooling in processing performance. Dropout, which explains removing entities from the network, is another important layer. It follows the overfitting reducing regularization strategy. Using dropout values that ranged from 0.2 to 0.8, the proposed model was trained and compared. Applying the convolutional and pooling layers results, the dense layer follows up with the classification.

Deep CNN is a highly iterative process that requires developing an assortment of models while deciding on the most effective one (Morgan Kaufmann, et al. 2019) [9].

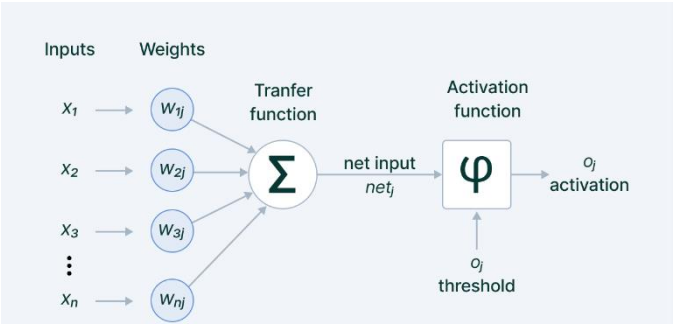

Fig. 3. The architecture of the deep CNN technique.

B. Transfer Learning

An approach for transmitting knowledge from one machine learning model to another is referred to as transfer learning by collecting bias and weight values from present models, it reduces the initial model construction phase of the new model. For instance, a machine learning model developed for task A could be used as a foundation for a model for task B. Current pre-trained models are used through transfer learning to gather knowledge that can be applied to new models.

In the research of Panigrahi R et al. (2018), AlexNet, Visual Geometry Group Network (VGGNet), Residual Network (ResNet), and Inception Network are among the most frequently utilized pre-trained deep learning models. A popular pre-trained deep convolutional neural network model is AlexNet.

The Alex Net comprises five convolutional layers with a Relu activation function and three fully connected layers. The Alex Net contains 62,000,000 trainable variables.

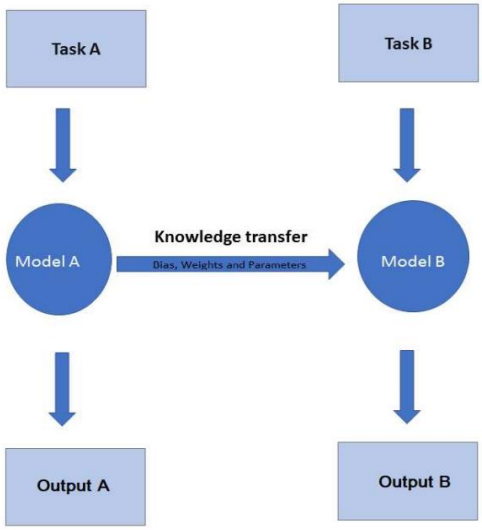

Fig. 4. Transfer learning techniques.

An example of an approach to transfer learning is shown in Fig. 4. In comparison with AlexNet, VGGNet increases performance while requiring less time to train. The convolutional and pooling layer kernels employed by VGGNet were lower than those utilized by AlexNet, indicating an important difference between the two networks. During the entire training phase, the kernel's size is fixed. VGG16 and VGG19 constitute two among the 14 distinct types of VGGNets. The number shows the number of network levels they are. There are 138 million trainable parameters in the VGG16. The ResNet addresses the vanishing gradient problem during the deep convolutional neural network training process. The ResNet makes use of a shortcut connection to improve network performance. In an entire network, there are only two pooling layers. ResNet18, ResNet50, and ResNet101 are the most common ResNet models. A total of eleven million trainable parameters in the ResNet18. In addition, the parallel kernel methods for handling flexible kernel values are presented for the inception net [10]. Google Net is the Inception Net's simplest iteration. In Google Net, there are 6.4 million trainable parameters.

DDoS volumetric attack constitutes the most harmful malicious internet traffic. This volumetric attack aims to

overwhelm the victim's computational capacity internet connections by having numerous attackers coordinate the transfer of a high rate of void data [11].

The proposed intrusion detection system for the detection of DDoS attacks and the Elliptic Curve Cryptography (ECC) - based secure encryption scheme. This method undergoes training and testing phases. In the training phase, the SDN dataset undergoes preprocessing where data nominalization, replacement of missing attributes and data normalization has been done. The proposed classification is carried out on the training phase to identify or classify the data to be normal or attacked data. Then the procedure is followed by preprocessing and classification as like in the training phase. Then if the classified data is normal, it can be further prevented from the attackers by encrypting the data using the ECC technique and stored in the network. If any need of encrypted real data in future will decrypt the data in the network and make use of it. Otherwise, the attacked (hacked) data is stored as a log file in the network for future attack detection. SDN-Data set which consists of forty-one features of is considered as normal and attacked type. Each feature is categorized into three types of attribute value types namely nominal, ordinal.

C. Hyperparameter Optimization

It is necessary for machine learning model training to offer optimal performance. Hyperparameters are parameters that are set before training begins and control aspects of the training process that are not learned from the data. These parameters can significantly affect a model's performance, generalization, and convergence [12].

Here are the steps and techniques involved in hyperparameter optimization:

Identify Hyperparameters: Start by identifying the hyperparameters that need to be optimized. Depending on the type of model you are training, these could involve learning rate, batch size, number of layers, number of units in each layer, dropout rates, regularization strength, etc.

D. Optimization Methods

Grid Search: For each hyperparameter, a grid of possible values needs to be specified, and all possible combinations should be thoroughly examined. It is easy to set up and can work well for a small parameter space, but it can be computationally expensive.

Random Search: Instead of trying all possible combinations, random search randomly selects parameter combinations to evaluate. It is more efficient than grid search for larger parameter spaces.

Bayesian Optimization: This is a more advanced approach that models the underlying function that maps hyperparameters to model performance. It uses this model to intelligently choose the next set of hyperparameters to evaluate, potentially reducing the number of evaluations needed.

Gradient-Based Optimization: Some libraries offer methods that use gradient-based optimization techniques to tune hyperparameters.

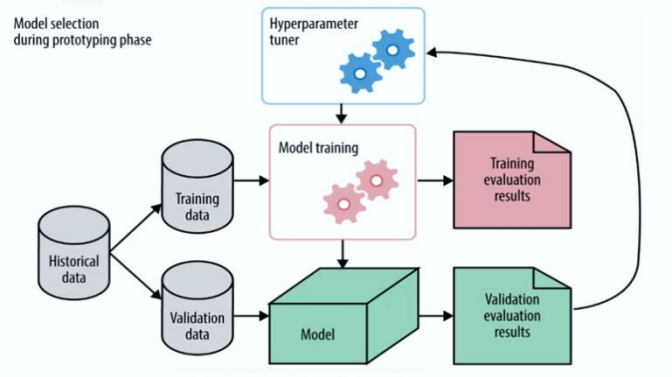

Fig. 5. Hyper parameter tuning process.

E. Implement Hyperparameter Search

For grid search or random search, loop through different hyperparameter combinations, train the model on the training set and evaluate it on the validation set using the chosen evaluation metric. For Bayesian optimization, use a library like scikit-optimize, or Bayesian Optimization.

Iterate and Tune: Based on the results of the validation performance, adjust the range or values of the hyperparameters and repeat the optimization process. Fig. 5 shows the hyper parameter tuning process.

Evaluate on Test Set: Once you have found the best hyperparameters using the validation set, evaluate the model on a separate test set that was not used during the optimization process to get an unbiased estimate of the model's performance.

V. Experimental Results

After Steps to be followed

- Feature extraction.
- Data modification
- Classification
- Decision making

A. Tools

We used i7 processor, a 256 SSD laptop TensorFlow, and Google Colab tools to train our model. We conduct perform manual attacks on DDoS and collect the data sets. We collect data sets of real-time attacks that have taken place previously from internet data sources.

B. Preprocessing

For the purpose of minimizing noise in the data and improving the efficacy of the previously mentioned technique, preprocessing should be done on the data. Feature extraction and data transfer to numerical values are the two phases of data preparation.

C. Feature Extraction

Feature extraction is the main step for data classification. In this project, three characteristics—packet length, delta time, and protocol—are considered in account.

D. Classification and Modification

With Data then, a numerical representation for these characteristics is created. The numerical values are fed to the ML models, with 30% utilized as training data and 70% for model testing. Each ML model is processed concurrently.

E. Decision Making

The voting majority among all algorithms determines the result.

Fig. 6 shows the training of the hyper model and Fig. 7, Fig. 8 shows the graph drawn between accuracy of the hyper model and epochs and loss vs. Epoch graphs.

```
Epoch 89/100
2272/2272 - 5s - loss: 0.0180 - accuracy: 0.9928 - val_loss: 0.0197 - val_accuracy: 0.9921 - 5s/epoch - 2ms/step
Epoch 90/100
2272/2272 - 5s - loss: 0.0218 - accuracy: 0.9916 - val_loss: 0.0206 - val_accuracy: 0.9917 - 5s/epoch - 2ms/step
Epoch 91/100
2272/2272 - 6s - loss: 0.0200 - accuracy: 0.9916 - val_loss: 0.0204 - val_accuracy: 0.9914 - 6s/epoch - 3ms/step
Epoch 92/100
2272/2272 - 4s - loss: 0.0202 - accuracy: 0.9917 - val_loss: 0.0235 - val_accuracy: 0.9897 - 4s/epoch - 2ms/step
Epoch 93/100
2272/2272 - 4s - loss: 0.0198 - accuracy: 0.9920 - val_loss: 0.0232 - val_accuracy: 0.9910 - 4s/epoch - 2ms/step
Epoch 94/100
2272/2272 - 6s - loss: 0.0191 - accuracy: 0.9923 - val_loss: 0.0240 - val_accuracy: 0.9914 - 6s/epoch - 3ms/step
Epoch 95/100
2272/2272 - 4s - loss: 0.0187 - accuracy: 0.9920 - val_loss: 0.0215 - val_accuracy: 0.9915 - 4s/epoch - 2ms/step
Epoch 96/100
2272/2272 - 6s - loss: 0.0204 - accuracy: 0.9922 - val_loss: 0.0231 - val_accuracy: 0.9908 - 6s/epoch - 3ms/step
Epoch 97/100
2272/2272 - 5s - loss: 0.0182 - accuracy: 0.9926 - val_loss: 0.0258 - val_accuracy: 0.9910 - 5s/epoch - 2ms/step
Epoch 98/100
2272/2272 - 4s - loss: 0.0186 - accuracy: 0.9924 - val_loss: 0.0252 - val_accuracy: 0.9904 - 4s/epoch - 2ms/step
Epoch 99/100
2272/2272 - 6s - loss: 0.0183 - accuracy: 0.9925 - val_loss: 0.0223 - val_accuracy: 0.9916 - 6s/epoch - 3ms/step
Epoch 100/100
2272/2272 - 4s - loss: 0.0201 - accuracy: 0.9921 - val_loss: 0.0313 - val_accuracy: 0.9907 - 4s/epoch - 2ms/step
```

Fig. 6. Training of proposed model.

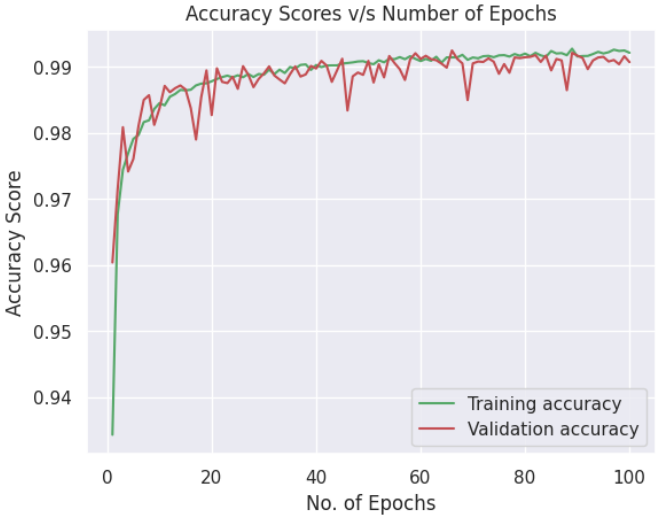

Fig. 7. Accuracy v/s Epochs in hyper model.

F. AUC and ROC

Evaluation of performance is an essential function in machine learning as shown in Fig. 9. So, we can depend on an AUC-ROC Curve when it comes to the classification their task. The AUC (Area Under the Curve) and ROC (Receiver Operating Characteristics) curves is employed to evaluate or show the performance of the multi-class classification trouble.

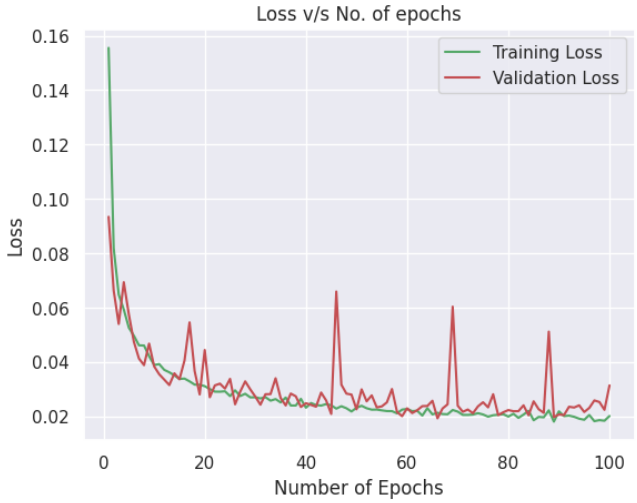

Fig. 8. Loss Vs Epochs.

It is one of the most essential criteria for assessing the efficiency of any classification model. AUROC (Area Under the Receiver Operating Characteristics) is a different method of expressing its contents.

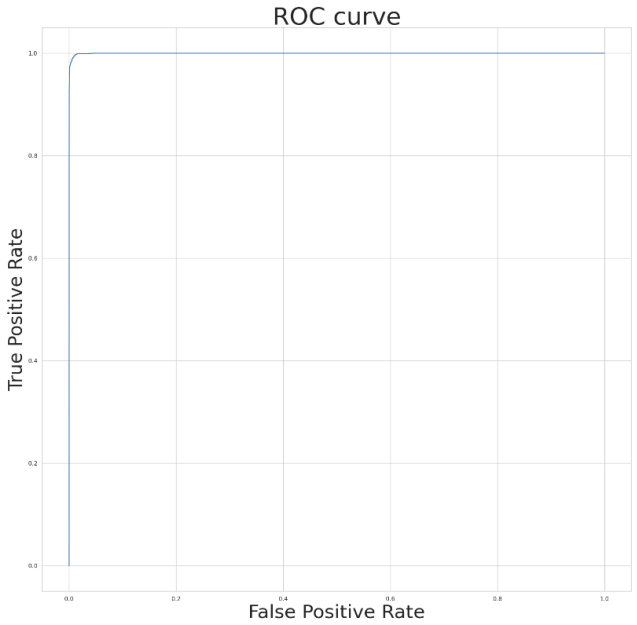

Fig. 9. ROC curve of Hyper model.

The accuracy of the proposed algorithm is compared with other standard machine learning algorithms in Table I.

TABLE I. COMPARISON OF ACCURACY OF DIFFERENT MODEL

S.no	Model	Accuracy
1	XGBoost	98.179892
2	KNN	96.809194
3	Decision Tree	96.619800
4	Proposed Hypermodel	99.072289

VI. CONCLUSION

Nearly all the work needs to be done manually while utilizing traditional methods, and the results are often inaccurate and difficult to identify risks. However, by employing a Machine Learning strategy, we can spot risks much more quickly than with traditional methods. Three neural network (ML) algorithms—Naive Bayesian, KNN, and Random Forest were employed in our proposal. The suggested hypermodel produced highly excellent precision and other output metrics like AUC and ROC by utilizing the hyper-tuning method in machine learning. We will be mounting it on all networks such routers, firewalls, and server since it has a limited resource consumption and can run on inexpensive components. The current system simply detects DDoS attacks, but in the future, we will improve it to a DDoS-avoiding model to improve server security. As technology advances, we will boost this model's efficiency.

REFERENCES

[1] G. Asad M, Asim M, Javed T, Beg MO, Mujtaba H, Abbas S (2020) Deep Detect: detection of Distributed Denial of Service attacks using deep learning. Computer J 63:983–994

[2] V. S, B. A. N.G. and N. -K. Baik, "Detection of DoS Attacks in Smart City Networks with Feature Distance Maps: A Statistical Approach," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2023.3264670.

[3] Mishra, A.; Gupta, N.; Gupta, B.B. Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms. Telecommun. Syst. 2023, 82, 229–244

[4] A. Haddaji, S. Ayed and L. C. Fourati, "A Transfer Learning Based Intrusion Detection System for Internet of Vehicles," 2023 15th International Conference on Developments in eSystems Engineering (DeSE), Baghdad & Anbar, Iraq, 2023, pp. 533-539, doi: 10.1109/DeSE58274.2023.10099623.

[5] Farhan Ullah, Shamsher Ullah, Gautam Srivastava, Jerry Chun-Wei Lin, IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic, Digital Communications and Networks, 2023,ISSN 2352-8648, https://doi.org/10.1016/j.dcan.2023.03.008.

[6] Bhardwaj A, Mangat V, Vig R (2020) Hyperband tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in Cloud. IEEE Access 8:181916–181929

[7] Catak FO, Mustacoglu AF (2019) Distributed denial of service attack detection using autoencoder and deep neural networks. J Intell Fuzzy Syst 37:3969–3979

[8] Chen J, tao Yang Y, ke Hu K, bin Zheng H, Wang Z (2019) DADMCNN: DDoS attack detection via multi-channel CNN. In: ACM international conference proceeding series, vol Part F1481. Association for Computing Machinery, New York, pp 484–488

[9] Morgan Kaufmann, pp 1–38 Hasan MZ, Hasan KMZ, Sattar A (2018) Burst header packet flood detection in optical burst switching network using deep learning model. Procedia Comput Sci 143:970–977

[10] He J, Tan Y, Guo W, Xian M (2020) A small sample DDoS attack detection method based on deep transfer learning. In: Proceedings—2020 International Conference on Computer Communication and Network Security, CCNS 2020. Institute of Electrical and Electronics Engineers Inc., pp 47–50

[11] Nugraha B, Murthy RN (2020) Deep learning-based slow DDoS attack detection in SDN-based networks. In: 2020 IEEE conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020—Proceedings. Institute of Electrical and Electronics Engineers Inc., pp 51–56

[12] Panigrahi R, Panigrahi R, Borah S (2018) A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. Int J Eng Technol 7:479–482 Premkumar M, Sundararajan TV (2020) DLDM: deep learning-based defense mechanism for denial-of-service attacks in wireless sensor networks. Microprocess Microsyst 79:103278