The Scientific Temper (2024) Vol. 15 (3): 2579-2587
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.25
https://scientifictemper.com/

RESEARCH ARTICLE

An optimized approach for detection and mitigation of DDoS attack cloud using an ensembled deep learning approach

P. S. Dheepika*, **V. Umadevi**

Abstract

As cloud computing gains in popularity, safety becomes an increasingly important consideration. One of the most challenging issues in cloud computing is the detection of distributed denial-of-service (DDoS) attacks (Gupta, B. B., *et al*., 2009). One of the most crucial aspects of cloud architecture is the ability to provide self-service whenever it is needed. Applications built on the cloud computing model are available on demand and at low cost. As cloud computing grows in popularity, so too is the amount of cyberattacks aimed against it. One such attack is a DDoS attack, which is designed to overload the cloud's hardware/software, resources, and services, making them difficult to use for everyone. The difficulty of this assault stems from the fact that it can overwhelm the victim's ability tocommunicate or compute in a short amount of time with little to no notice. It's gettingharder to spot and stop these assaults as they get more sophisticated and more numerous. Several machine learning methods, including logistic regression, K-nearest neighbors, support vector machine, decision tree, naive Bayes, multi-layer perceptron, XGBoost, and SGD, have been implemented for accurate DDoS flooding attackdetection. When compared to current methods, the suggested strategy of utilizing deep learning with quadratic discriminant appears to result in higher accuracy. There is also a thorough comparisonand evaluation of the abovementioned algorithms with respect to the accuracy measures used.

Keywords: DDoS attack, Cloud computing, Deep learning, SDN, Classifier, Quadratic discriminant.

Introduction

Cloud computing is an excellent rephrasing for "centralization," which refers to the practice of housing several computer services on a single server. The relocation of data and programs away from personal computers and desktop computers and onto the "cloud" Cloud computing is a highly formidable rival in the field of information technology since it provides "pay as you go" services at reduced prices. The majority of businesses and organizations of a significant size have already moved their data to the cloud. Cloud computing has helped to alleviate several challenges relating to time, effort, and cost by delivering services that need the least amount of money, the least amount of time, and the least amount of effort. SaaS, PaaS, and IaaS are the three cloud services that are proving to be the most useful to customers, despite the fact that cloud computing offers its consumers a wide variety of service options. Software as a service is what SaaS stands for, system as a Service, while PaaS and IaaS stand for framework as a service. DDoS attacks are the biggest danger to the World Wide Web, the Internet of Things (IoT), smart cities, medical care, technology in general, and the business parts of our lives. Denial-of-service attacks remain a risk to the integrity of connections in every business.

Despite their enormous size, they are becoming increasingly complicated, loud, and frequent. Two types of distributed denial of service attacks may be distinguished: (i) DDoS attacks that rely on reflections are discussed in the first section (Figure 1). In this stage of the attack, the attacker hides their true identity by using cyberspace devices to deliver traffic from the attack to the desired level, such as HTTP calls. These requests are sent out via the requesting host's IP address, utilizing the reflector servers' IP addresses as their ultimate destination. As a result, the

PG and Research Department of Computer Science, Nehru Memorial College, Puthanampatti (Affiliated to Bharathidasan University), Tiruchirapalli, India

***Corresponding Author:** P. S. Dheepika, PG and Research Department of Computer Science, Nehru Memorial College, Puthanampatti (Affiliated to Bharathidasan University), Tiruchirapalli, India, E-Mail: psdheepika@gmail.com

How to cite this article: Dheepika, P. S., Umadevi, V. (2024). An optimized approach for detection and mitigation of DDoS attack cloud using an ensembled deep learning approach. The Scientific Temper, **15**(3):2579-2587.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.25

Source of support: Nil

Conflict of interest: None.

Received: 12/07/2024 **Accepted:** 11/08/2024 **Published** : 10/09/2024

victim is made aware of all of these competing needs at the same time. In most cases, these attacks are used to misbehave in accordance with the application standards. Because it should be evident that everything that makes life easier for people may also have drawbacks. The usage of cloud computing is fraught with danger because of the vast quantities of data that are stored on the cloud; the prevalence of cloud computing is only expected to increase, bringing with it an escalation in the number of potential threats. There are many various types of assaults that are capable of causing serious harm to data that is stored on a cloud, but many writers have pointed out that a DDoS attack is one of the most damaging kinds of attack, and moreover, it may be regarded as an alert for cloud customers. There are a few various ways one may characterize a DDoS assault; however, regardless of the definition one uses, the attack still has the same significance. In order to launch a distributed denial of service attack, a hacker would first need to identify vulnerable network devices. This gives the attacker the ability to run his program on infected devices and take full control of any compromised machines. The DDoS attacker schedules his script to execute at a specified time, at the moment when all compromised devices begin sending massive amounts of traffic to the target server all at once, with the goal of exhausting the host’s bandwidth or resources (Kasinathan, P., *et al*., 2013). As an outcome, the compromised server failed to meet the needs of its typical clientele and had to refuse them service. The fundamental goal of a DDoS assault is to limit the access of the targeted system by making it inaccessible to legitimate users.

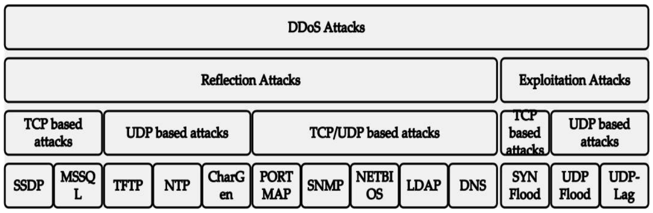


Figure 1: Taxonomy of DDoS attack

Cloud Characteristics

IaaS provides access to all of a company’s computer resources through the use of internet-based virtualization. Users of IaaS are able to acquire resources and finish services *via* WAN, such as the internet. Users are able to install components of an application by taking advantage of the capabilities offered by cloud computing. Customers of infrastructure as a service often pay for the services they use on an as-needed basis, typically by the hour, week, or month. Certain users are going to be compensated for the amount of virtual machine space that they have consumed. The pay-as-you-go model eliminates the need for upfront capital expenditures to implement software and hardware in-house. Platform as a service is a model of cloud computing in which service providers provide their customers access to the software platform on which they run their businesses. The PaaS provider is responsible for supplying all of the necessary hardware and software to construct such applications over a variety of channels, including dedicated networks and VPNs. Direct attack: By sending a torrent of packets straight to the victim server, the attacker was able to prohibit or limit access to the server that was the focus of their assault (Joosten, R., & Nieuwenhuis, L. J. (2017)). However, “directly” in this context does not suggest that the traffic came from the attacker themselves; rather, it came via compromised or compromised machines elsewhere in the network.

The TCP SYN flood attack technique is used in these situations, and it consists of sending a large number of SYN signals to the victim server. Spoofing is another method used by the attacker; it tricks the victim server into sending SYN-ACK packets to an erroneous address. This keeps happening until the affected server can no longer take any new requests. The field of machine learning, known as deep learning, has profound effects on how computers and data are processed. The method does this by using many layers of nonlinear processing to directly extract characteristics of interest from the input. This information might be text, photos, or network traffic. To create highly reliable models for dataclassification, deep learning has emerged as the state-of-the-art approach. As a large amount of data is readily available, there is a need for deep learning. All DNNs have input, hidden, and output layers. Labeling pictures using convolutional neural networks (CNNs) is common. Commercial applications, services, and networks make distributed denial of service assaults (DDoS) a key security threat (Samtani, S., *et al*., 2020). DDoS assaults are similar to legitimate concerns about availability, such as when IT staff do routine maintenance or when users report problems accessing the internet (Chen, Y. W., *et al*., 2020). Because of these challenges, identifying and countering the aforementioned modes of attack is much more challenging. The network’s speed may slow down while trying to detect a DDoS attack, making it harder to retrieve data or figure out why a certain website is inaccessible.

Related Work

The last ten years have seen a surge in the usage of deep learning methods and techniques in the networking industry. DDoS attacks are challenging to detect because it is difficult to tell malicious traffic apart from benign traffic. It is possible to tell good traffic from bad using deep learning classification techniques. Naive Bayes, Random Forest tree, and other machine learning algorithms have classified DDoS attack packets (Tahsien, S. M., *et al*., 2020). Several articles covered the topic of employing an ANN to spot distributed denial of service attacks.

However, Model (Jia, Y., *et al*., 2020) has demonstrated poorer accuracy in the categorization of UDP assaults, despite its ability to identify all three forms of DDoS attacks by analyzing collected data and extracting the utilized characteristics.

Analysis of network traffic patterns may be used to identify DDoS assaults, as was proved in an article (Galeano-Brajones, J., *et al*., 2020), which locates criteria such as time to live, protocol, source port number, and IP addresses. An old data set was utilized to inform a suggested model for identifying DDoS attacks based on these criteria.

In order to identify and mitigate DDoS assaults, the authors (Bhardwaj, K., *et al*., 2018) suggest an edge-centric approach in which IoT devices play an active role. Short-term memory classifiers are utilized, and their internal structure is affected by differences in network traffic; convolutional neural network (CNN) classifiers are also part of the strategy. The first has a 98.9% success rate for identifying objects, while the CNN achieves a perfect 99.9%. When used on edge servers that are more robust than a desktop computer, the suggested method provides reduced operating latency. The detection of DDoS assaults associated with the internet of things can be facilitated by an entropy-based detector that makes use of software defined networking (Galeano-Brajones, J., *et al*., 2020).

The states of SDN, evaluated across a data space representation, provide that form of detecting system overall confidence of discovering malicious actions anywhere from 68 to 99.7%. Cybercriminals have utilized distributed denial of service (DDoS) assaults to bring down target servers and break into enterprise networks with the capacity to overwhelm outcomes. Many companies today are having difficulty keeping up with DDoS attacks because they are getting bigger and more complicated. Hackers know about new systems and how they can be broken. With the latest technological breakthroughs, smart gadgets and the internet of things are especially vulnerable to DDoS attacks because they have limited memory and working power (Mouli, V. R., & Jevitha, K. P. (2016). In 2016, a hack on ISPs caused broad service problems for 9 hours. Netflix, CNN, and Twitter were among the companies that were affected. This tech problem caused a number of problems, which ranged from financial losses, less work getting done, damage to the brand, lower insurance ratings, shaky relationships between customers and vendors, and going across the IT budget (Kumar, R., *et al*., 2021). In this piece, we talk about the findings of many studies that used DL to find DDoS attacks. In this part, the results of a research study with datasets that included DDoS attacks were summarized, and multiple deep learning models were talked about. There are three main ways to find IDS (Sallam, A. A., *et al*., 2020): methods based on signatures, methods based on anomalies, and hybrid-based methods.

However, writers in (Nagpal, B., *et al*., 2015) showed that the SVM model had superior accuracy when it came to detection. Five different classifier approaches were employed to identify DDoS assaults in IoT networks; all of them obtained an accuracy of 99.9% or above, which should encourage researchers to resume their work in this field (Abu-Mostafa, Y. S., *et al*., 2012). Although here the authors used a neural network approach and trained their model on a big dataset, they still found that the gathered packets sometimes exhibited traits that the system had not been learned before (Zhang, B., *et al*., 2017). In addition, ANN was employed by (Alsirhani, A., *et al*., 2019) to identify and categorize DDoS attacks. The authors only retrieved roughly five variables from the traffic on the network to be taught by the model; hence the accuracy with which it classified the three forms of assault was variable. It was found in a survey (Yudhana, A., *et al*., 2018) of ML techniques that were capable of identifying DDoS attacks that none of the tested techniques could outperform the others. However, the authors failed to illustrate the characteristics extracted from TCP headers and ICMP headers that were used by all of the techniques for learning (Sahi, A., *et al*., 2017).

The accuracy of the models used in (Peraković, D., *et al*., 2017) was greater than 98%, regardless of the model type employed, although the models were given outdated data sets that contained too many characteristics. Another study applied ML classification algorithms in a software-defined networking context (Thapngam, T., *et al*., 2014), where the authors made the critical insight that any classifier system gives superior accuracy on unrelated training data collection than on real-time traffic.

In a DDoS attack, the attacker generally uses innocent computers (zombies) by taking the pros of known or unknown bugs to send a large number of packets from these already-captured zombies to a server. This may occupy a larger portion of the network bandwidth of the cloud infrastructure and take much of the server's time. C.4.5 algorithm has been considered for designing a system to diminish the DDoS threat. This algorithm is generated with signature detection techniques that generate a decision tree to perform automatic and efficient detection of signatures attacks (Zekri, M., *et al*., 2017).

Here, they present a study of IDS research for IoT with the objective of identifying leading trends, open issues, and future research possibilities. They classified the IDSs based on detection method, IDS placement strategy, security threat and validation strategy. And also develop specific IDS schemes for IoT or attack detection strategies for IoT threats that might be embedded in IDSs (Zarpelão, B. B., *et al*., 2017).

In this article a study focused on enhancing IoT device and network security and privacy through experimental research and advanced machine learning techniques. This research contributes to the rapidly increasing field of IoT security by employing machine learning as a tool for fortifying IoT device and network security. They reveal a performance profile, flaking light on the model's potential to accurately categorize IoT devices as secure or vulnerable (Sreenivasulu, K., *et al*., 2023).

Authors in this work explain static IP addresses can develop cloud security by giving an additional level of protection (Aslam, J. M., & Kumar,K. M. (2024)). By importing stringent user authentication and access control measures, organizations can increase the security of their cloud resources and protect confidential information.

In this paper, they talk about the idea of malware and botnets working behind 'Distributed' DoS in IoT. The various DDoS defense techniques are described and compared to identify the security gaps present in them. And they list out the open research issues and challenges that need to be addressed for DDoS defense (Vishwakarma, R., & Jain, A. K. (2020)).

In this article, the authors have projected EAM and EHT has experimented within the cloud server, and the performance is analyzed based on the computation taken for each proposed procedure. The EHT performance is measured from the computation time for generating the data's hashcode. The EHT is tested with different data sizes, and time is calculated in milliseconds.

It also provides a method to verify the data authenticity when migrated. It gives a new technique to transfer the data in the virtual machine. The result shows that the EAM efficiently provides authentication and integrity of data migrated from on-premises to the cloud data center (Selvaraj, R., & Sundari, M. S. (2023)).

In this article, an article has been proposed that is categorized by efficient cryptographic operations gives the best solution in both efficiency and security. It also gives the best in its reduced computation costs on both the semi-trusted server and the IoT-cloud side. This helps in giving efficient, secure, and privacy-preserving data management. This research provides a strong foundation for the development of advanced data integrity solutions that prioritize efficiency, security, and scalability in equal measure (Gokulkannan, K., *et al*., 2024).

Methodology

Naive Bayes Classifier

Naïve Bayes classifiers are simple probabilistic machine learning models. It begins by computing the likelihood of every category in a dataset and then applying discriminative learning in order to forecast the outcome of a new class. Suppose the characteristics are adequate on their own. In that case, we may use Bayes' theorem to determine the likelihood of event A (the hypothesis) occurring given that event B (the evidence) has already taken place. This is a naive assumption, given that the outcome of one estimate does not impact the outcome of another.

Decision Tree

A non-parametric supervised learning approach that belongs to the class of decision trees is called a decision tree. Its primary function is that of assist with the resolution of difficulties involving regression and classification. The primary objective is to construct a model that is capable of making accurate forecasts regarding the value of a variable of interest. This may be accomplished by studying the straightforward rules of a decision tree, which are deduced based on the characteristics of the data. One may think of the tree as an approximation of a piecewise constant. In order to find a solution to the classification issue, the class DecisionTreeClassifier is utilized. It shouldn't be too hard for the class to classify the information in more than one way. As input, the classifier can take either a dense collection X of shape comprising the training data from the dataset or an array of integers Y of shape containing the labels for the training samples. After the algorithm has been fit, it is used to make predictions about the category of the test samples. The classifier has a tendency to make its prediction based on the category that has the lowest index out of all of the classes when there are numerous classes that have the same exact probability (Sharma, K., & Mukhopadhyay, A. (2020)). The probabilities of each class are defined as the fraction of the sample used for training that matches the class in a leaf might be predicted in addition to being output to a specific class. The classifiers can both separate things into two groups and separate things into more than two groups.

K-Nearest Neighbours

When it comes to supervised machine learning, K-nearest neighbour is one of the easiest approaches. The classifiers can both separate things into two groups and separate things into more than two groups. With the K- NN technique, new data may be quickly placed in an appropriate category as it becomes available. This is so because it establishes the boundaries of the new knowledge by comparing it to previously available facts. The KNN classifier has been shown to be effective in detecting invasive attacks while also having a low false-positive rate. It is able to differentiate between normal and aberrant patterns of activity and classifies the condition of networks at various phases of a distributed denial of service assault.

Support Vector Machine

Support Vector Machines are increasingly useful for pattern recognition, filtering out scams, and intrusion detection (Liu, C., *et al*., 2011). Regression, categorization, and distribution estimation SVM formulations exist. Linearly separated data and the best classification hyperplane provide it. A training set is denoted by the notation D = (X1, y1),... (Xn, yn). In this context, yi is the class label and Xi is a characteristic vector of the training samples. It supports the values 1 and -1 to indicate whether or not the vector's value is within the 1 or -1 range. The two groups are said to be linearly separable if and only if an exponential relationship can fully disentangle them from one another. Since detecting a DDoS attack is analogous to solving a problem involving binary classification, we can employ the SVM algorithm to collect data, identify characteristic details for modeling, locate the most efficient categorizing hyperplane between benign and malicious traffic, and finally, test our model and obtain classification results (Subbulakshmi, T., *et al*., 2011).

Random Forest

A classification estimate may be obtained from each and every tree that appears in a random forest. The result of this process is that the class that received the most votes ultimately serves as the basis for the model's overall forecast. The primary goal of the classifiers is to have a sizeable number of trees that, when combined and worked as a system, are superior to the performance of each of its constituent models. The importance of having a small correlation across the models cannot be overstated. The ability of uncorrelated models to provide models with greater precision than any of each of the forecasts gives these models an advantage. The primary reason for this is that the trees cover each other's mistakes made by individuals. If the majority of the trees in the group had the correct information, then the collection as a whole would be able to go in the proper path even while certain of the trees in the group had incorrect information. The classifier uses attribute randomization and bagging in order to produce each unique tree and to build a forest of trees that have no connection with one another in any way. This is accomplished by generating a forest of trees.

Deep Neural Network

Deep neural networks are among the most popular and cutting- edge models now in use. One way to conceptualize this model is as a multi-layered network, similar to a layer of neural networks. DNN has been successfully used in a wide variety of contexts, most notably those involving auto-regressive models for the prediction of time series. At a minimum, there are three tiers of nodes in this design, which are referred to as the "input," "hidden," and "output" levels. Each of these stages is linked to the next, and information flows unidirectionally from the data input terminals to the output ones. The next version of the DNN uses a function of activation for classification and a training method called reverse propagation. To train a deep neural network to differentiate between normal and DDoS attack states, we pick a subset of data from the network to use as an input signal.

Stochastic Gradient Descent Classifier

During the training period, this classification uses the random gradient descent optimization method to create standardized linear models like SVM and logistic regression, among others. This method is used as part of the training. Each sample at a time, this algorithm figures out the gradient of the loss, and it changes the model by predicting the minimal cost function. This is done by slowing down the acquisition rate or the strength plan. The SGD encoder is able to accurately anticipate outcomes for complex situations at scale because it uses minibatch learning along with the partial fit approach. When data exceeds RAM limits, simple linear classifiers fail. However, the SGD classifier continues to function normally. This framework is sensitive to how big the features are, and it needs exact changes to a lot of hyperparameters, like the total number of rounds and the parameter for regularization, in order to work well.

Proposed Algorithm

Loop

- Confidence values are computed.
- Anyone valued with the most confident unlabeled class is selected.
- The predicted one value with a label is moved from unlabeled class to labeled data L [Based on features]
- The existing feature vector is computed.
- After computation, most prompt and flexible labels are selected in every class and labeled as class C.
- The feature values are updated every time.
- The above steps are repeated. [Based on unlabeled data]

Loop ends

The classifier is trained based on the features chosen and it includes {x1,x2} labeled to data L result set {x1,x2},C is returned.

Discriminant Analysis

In the last step, a quadratic discriminant analysis is incorporated into the deep learning algorithm. Within this analysis, each class is generative and follows a Gaussian distribution. It is very close to linear discriminant analysis, with the main change being that all of the classes have the same correlation and mean. The fraction of the information points that belong to a certain class is what is meant when we talk about the "class-specific prior" (Figure 2). The term "covariance of the vectors that belong to that class" refers to the information included in the "class-specific covariance matrix." The term "class-specific mean vector" denotes the arithmetic mean of the input variables that are associated with that class.

Implementation Results

An SDN-specific dataset (Figure 3) known as the distributed denial of service dataset was created by using the mininet emulator throughout the data collection process. Its principal function is to provide assistance to a variety of machine learning and deep learning algorithms so that they may more accurately classify incoming traffic. The

```
Out[12]: Text(0, 0.5, 'Frequency %')
```

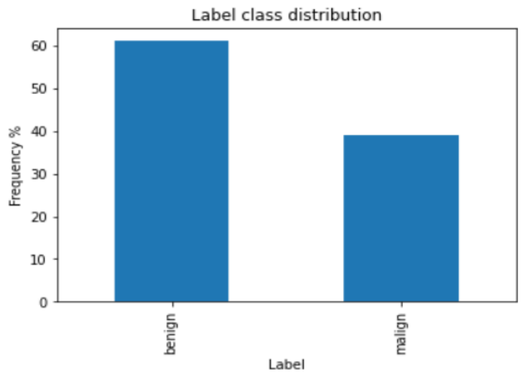


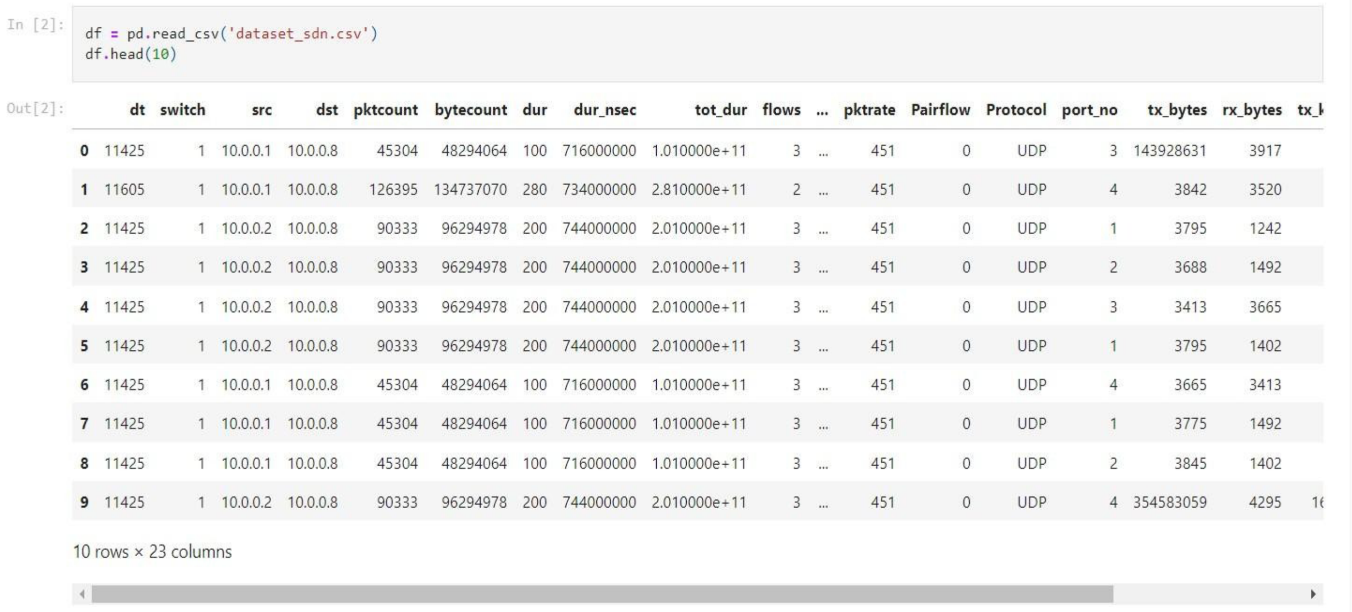
Figure 2: Class distribution of DDos attacks

development of the dataset necessitates a number of stages, one of which entails the building of ten distinct topologies in mininet. Each of these topologies includes only one Ryu controller that acts as the hub between all of the switches in the topology. The network simulation is done for both of the benign TCP, UDP, and ICMP traffic as well as the collecting of damaging data to simulate TCP Syn attack, ICMP attack, and UDP flood assault. In addition, the simulation runs for both types of traffic simultaneously. The simulation includes modeling for all three of these different kinds of assaults. The dataset consists of a total of 23 characteristics, certain of which get the information directly from the switches, while other characteristics are the outcome of mathematical calculations based on those switches' values (Figure 4). The following items may be found on the set of extracted features that have been taken from the dataset:

- Packet_count is the number of packets that have been counted.
- byte_count is the number of bytes that are contained within the packet.
- ID of the switch is referred to as the switch-id.
- duration_sec is the amount of time a packet is sent in seconds.
- duration_nsec is the amount of time a packet was sent in nanoseconds.
- Source IP is the IP address of the machine that is sending the data.
- Destination IP is the Internet protocol address of the computer that will receive the data.
- Number of Ports – The number of ports used by the program
- tx_bytes - represents the total amount of bytes that havebeen sent from the switch.
- port rx_bytes - represents the total amount of bytes that the switch has received.
- port dt field displays the time and the date as a number once it has been converted, and the flow is checked at a monitoring period of 30 seconds.

The following are examples of computed characteristics that are included in the dataset:

- Byte per flow refers to the total number of bytes transmitted in a particular flow.
- Packets transmitted in a single flow are referred to as the "packet per flow."
- Packet rate is the number of packets that are communicated in one second. This value is derived by dividing the number ofpackets that are sent for each flow by the number of monitoring intervals.
- "packet_ins" are communications that are initiated by the switching device andsent to the controller.



```
In [2]: df = pd.read_csv('dataset_sdn.csv')
        df.head(10)
```

Out[2]:

	dt	switch	src	dst	pktcount	bytecount	dur	dur_nsec	tot_dur	flows	...	pktrate	Pairflow	Protocol	port_no	tx_bytes	rx_bytes	tx_k
0	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3	...	451	0	UDP	3	143928631	3917	
1	11605	1	10.0.0.1	10.0.0.8	126395	134737070	280	734000000	2.810000e+11	2	...	451	0	UDP	4	3842	3520	
2	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	1	3795	1242	
3	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	2	3688	1492	
4	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	3	3413	3665	
5	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	1	3795	1402	
6	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3	...	451	0	UDP	4	3665	3413	
7	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3	...	451	0	UDP	1	3775	1492	
8	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3	...	451	0	UDP	2	3845	1402	
9	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	4	354583059	4295	16

10 rows × 23 columns

Figure 3: Loading dataset

- Transfer entries of switch are records in the flow database of a switch that are used to coordinate and process packets.
- Flow tables are used to organize the flow of data via a switch.

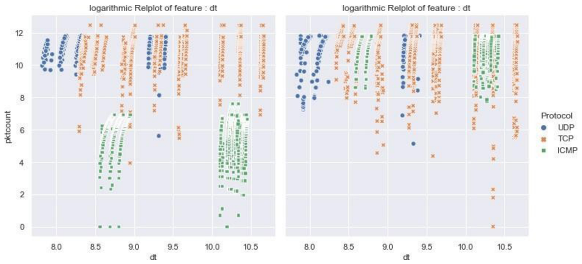
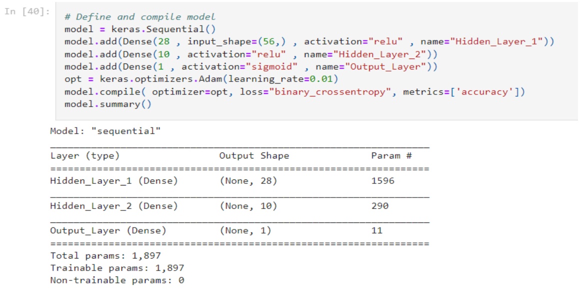


Figure 4: Visualize the distribution of continuous features wrt. packet count, protocol and type of attack



```
In [40]:  # Define and compile model
          model = keras.Sequential()
          model.add(Dense(28 , input_shape=(56,) , activation="relu" , name="Hidden_Layer_1"))
          model.add(Dense(10 , activation="relu" , name="Hidden_Layer_2"))
          model.add(Dense(1 , activation="sigmoid" , name="Output_Layer"))
          opt = keras.optimizers.Adam(learning_rate=0.01)
          model.compile( optimizer=opt, loss="binary_crossentropy", metrics=['accuracy'])
          model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Hidden_Layer_1 (Dense)       (None, 28)                1596
_________________________________________________________________
Hidden_Layer_2 (Dense)       (None, 10)                290
_________________________________________________________________
Output_Layer (Dense)         (None, 1)                 11
=================================================================
Total params: 1,897
Trainable params: 1,897
Non-trainable params: 0
```

Figure 5: Defining deep neural network

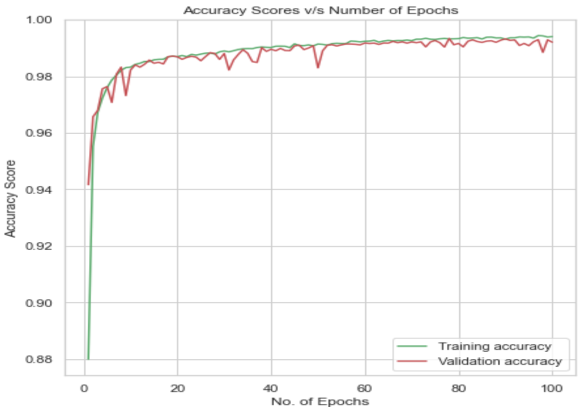
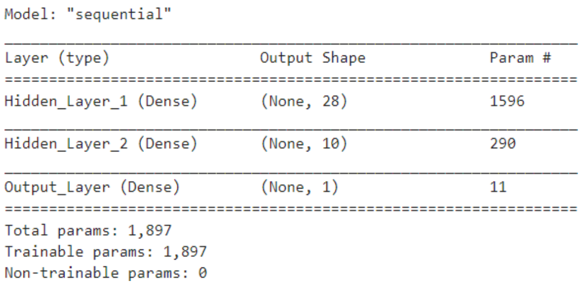


Figure 6: Plotting accuracy Vs epochs



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Hidden_Layer_1 (Dense)       (None, 28)                1596
_________________________________________________________________
Hidden_Layer_2 (Dense)       (None, 10)                290
_________________________________________________________________
Output_Layer (Dense)         (None, 1)                 11
=================================================================
Total params: 1,897
Trainable params: 1,897
Non-trainable params: 0
```

Figure 7: Final fitting of hypermodel layers

- tx_kbps is the speed of the packet transfer measured in kilobits per second.
- rx_kbps is the average rate of packet reception expressed in kilobits per second.

Bandwidth of a Port = Addition of transmit and receive rates in kbps

The class label is the output characteristic that appears in the last column of the data set. This column indicates whether the given traffic type is beneficial or detrimental. The malicious activity is represented by a value of 1, whereas legitimate data is represented by the value 0. Approximately 250 minutes were spent simulating the network, during which time 1, 04, 345 separate data instances were gathered and stored (Figures 5, 6). In addition to this, the simulation was run for a predetermined amount of time so that additional specific samples of data could be collected (Figure 7).

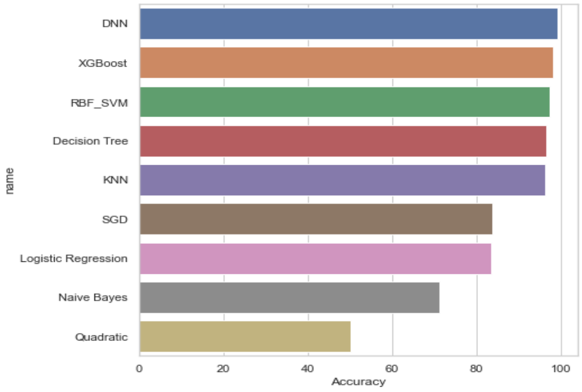
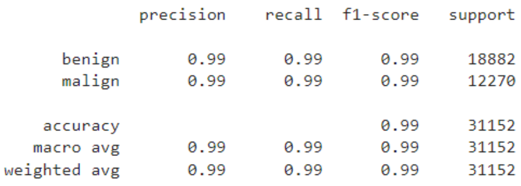


Figure 8: Visualize accuracies of the models



```
              precision    recall  f1-score   support

      benign       0.99      0.99      0.99     18882
      malign       0.99      0.99      0.99     12270

    accuracy                           0.99     31152
   macro avg       0.99      0.99      0.99     31152
weighted avg       0.99      0.99      0.99     31152
```

Figure 9: Classification report of accuracy metrics

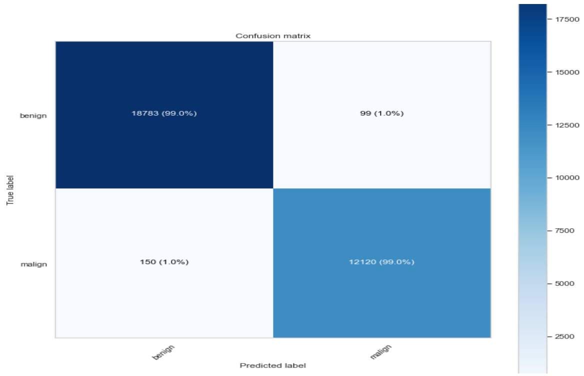


Figure 10: Confusion matrix

With the use of discriminant analysis, the variable settings of DNN have now been fine-tuned. Additionally, the particular criteria are brought into focus (Figure 8). After selecting the optimal parameters, the model is then examined for its overall performance (Figure 9). With a precision of 99.18785095214844, DNN is the strongest baseline classifier currently available (Figure 10).

Conclusion

The use of machine learning was employed to analyze and detect DDoS attacks. The present study employs a set of data that is specifically dedicated to software-defined networks (SDNs). Initially, the dataset comprised 23 distinct features. The concluding column of the information set is commonly referred to as the category label, which serves as the output feature. This column classifies the traffic as benign or malicious. Fraudulent traffic is represented by the numerical value of 1, while benign traffic is represented by the numerical value of 0. There area total of 104345 instances of it. The dataset was cleansed by removing null values from the rx_kbps and tot_kbps variables to enhance the model-building process. All procedures related to data processing, such as preparing the data and the extraction process, one hot encoding, and standardization, have been completed. After undergoing a single cycle of hot encoding, the aforementioned data frame contained 103839 instances and 57 attributes prior to being fed into the model. A deep neural network was employed as the foundational model. It has been determined that the efficacy of the proposed model surpasses that of the benchmark classifiers that were employed. The proposed model demonstrated an accuracy of 99.38%, surpassing the next best-performing model, XGBoost, by approximately 1.21%, with an accuracy of 98.17%.

Acknowledgments

The authors would like to thank Nehru Memorial College, Puthanampatti - 621007, which is affiliated with Bharathidasan University, Tiruchirappalli, Tamilnadu, for helpful supporting this work in an effective manner.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). *Learning from data* (Vol. 4, p. 4). New York: AMLBook.

Alsirhani, A., Sampalli, S., & Bodorik, P. (2019). DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in apache spark. *IEEE Transactions on Network and Service Management*, *16*(3), 936-949.

Aslam, J. M., & Kumar, K. M. (2024). Enhancing security of cloud using static IP techniques. *The Scientific Temper*, *15*(01), 1790-1798.

Bhardwaj, K., Miranda, J. C., & Gavrilovska, A. (2018). Towards {IoT-DDoS} Prevention Using Edge Computing. In *USENIX workshop on hot topics in edge computing (HotEdge 18)*.

Chen, Y. W., Sheu, J. P., Kuo, Y. C., & Van Cuong, N. (2020, June). Design and implementation of IoT DDoS attacks detection system based on machine learning. In *2020 European Conference on Networks and Communications (EuCNC)* (pp. 122-127). IEEE.

Galeano-Brajones, J., Carmona-Murillo, J., Valenzuela-Valdés, J. F., & Luna-Valero, F. (2020). Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: An experimental approach. *Sensors*, *20*(3), 816.

Gokulkannan, K., Parthiban, M., Jayanthi, S., & Kumar, M. (2024). Cost effective cloud-based data storage scheme with enhanced privacy preserving principles. *The Scientific Temper*, *15*(02), 2104-2115.

Gupta, B. B., Joshi, R. C., & Misra, M. (2009). Defending against distributed denial of service attacks: issues and challenges. *Information Security Journal: A Global Perspective*, *18*(5), 224-247.

Jia, Y., Zhong, F., Alrawais, A., Gong, B., & Cheng, X. (2020). Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet of Things Journal*, *7*(10), 9552-9562.

Joosten, R., & Nieuwenhuis, L. J. (2017, March). Analysing the impact of a DDoS attack announcement on victim stock prices. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (pp. 354-362). IEEE.

Kasinathan, P., Pastrone, C., Spirito, M. A., & Vinkovits, M. (2013, October). Denial-of-Service detection in 6LoWPAN based Internet of Things. In *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)* (pp. 600-607). IEEE.

Kumar, R., Kumar, P., Tripathi, R., Gupta, G. P., Kumar, N., & Hassan, M. M. (2021). A privacy-preserving-based secure framework using blockchain-enabled deep-learning in cooperative intelligent transport system. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 16492-16503.

Liu, C., Yang, J., Chen, R., Zhang, Y., & Zeng, J. (2011, July). Research on immunity-based intrusion detection technology for the Internet of Things. In *2011 Seventh International conference on natural computation* (Vol. 1, pp. 212-216). IEEE.

Mouli, V. R., & Jevitha, K. P. (2016). Web services attacks and security-a systematic literature review. *Procedia Computer Science*, *93*, 870-877.

Nagpal, B., Sharma, P., Chauhan, N., & Panesar, A. (2015, March). DDoS tools: Classification, analysis and comparison. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 342-346). IEEE.

Oliveira, R. A., Laranjeiro, N., & Vieira, M. (2015). Assessing the security of web service frameworks against Denial of Service attacks. *Journal of Systems and Software*, *109*, 18-31.

Peraković, D., Periša, M., Cvitić, I., & Husnjak, S. (2017). Model for detection and classification of DDoS traffic based on artificial neural network. *Telfor Journal*, *9*(1), 26-31.

Sahi, A., Lai, D., Li, Y., & Diykh, M. (2017). An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. *IEEE Access*, *5*, 6036-6048.

Sallam, A. A., Kabir, M. N., Alginahi, Y. M., Jamal, A., & Esmeel, T. K. (2020, February). IDS for improving DDoS attack recognition based on attack profiles and network traffic features. In *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 255-260). IEEE.

Samtani, S., Kantarcioglu, M., & Chen, H. (2020). Trailblazing the artificial intelligence for cybersecurity discipline: A multi-disciplinary research roadmap. *ACM Transactions on Management Information Systems (TMIS)*, *11*(4), 1-19.

Selvaraj, R., & Sundari, M. S. (2023). EAM: Enhanced authentication method to ensure the authenticity and integrity of the data in VM migration to the cloud environment. *The Scientific Temper*, *14*(01), 227-232.

Sharma, K., & Mukhopadhyay, A. (2020, August). Cyber Risk Assessment and Mitigation Using Logit and Probit Models for DDoS attacks. In *AMCIS*.

Sreenivasulu, K., Sampath, S., Gopi, A., Kartikey, D., Bharathidasan, S., & Kumar, N. L. (2023). Advancing device and network security for enhanced privacy. *The Scientific Temper*, *14*(04), 1271-1276.

Subbulakshmi, T., BalaKrishnan, K., Shalinie, S. M., AnandKumar, D., GanapathiSubramanian, V., & Kannathal, K. (2011, December). Detection of DDoS attacks using Enhanced Support Vector Machines with real time generated dataset. In *2011 Third International Conference on Advanced Computing* (pp. 17-22). IEEE.

Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. *Journal of Network and Computer Applications*, *161*, 102630.

Thapngam, T., Yu, S., Zhou, W., & Makki, S. K. (2014). Distributed Denial of Service (DDoS) detection by traffic pattern analysis. *Peer-to-peer networking and applications*, *7*, 346-358.

Vishwakarma, R., & Jain, A. K. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication systems*, *73*(1), 3-25.

Yudhana, A., Riadi, I., & Ridho, F. (2018). DDoS classification using neural network and naïve bayes methods for network forensics. *International Journal of Advanced Computer Science and Applications*, *9*(11).

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & De Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, *84*, 25-37.

Zekri, M., El Kafhali, S., Aboutabit, N., & Saadi, Y. (2017, October). DDoS attack detection using machine learning techniques in cloud computing environments. In *2017 3rd international conference of cloud computing technologies and applications (CloudTech)* (pp. 1-7). IEEE.

Zhang, B., Zhang, T., & Yu, Z. (2017, December). DDoS detection and prevention based on artificial intelligence techniques. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)* (pp. 1276-1280). IEEE.