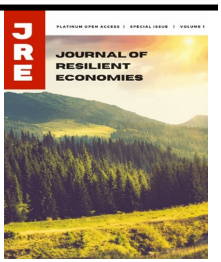
JRE
OPEN ACCESS

JOURNAL OF RESILIENT ECONOMIES

PLATINUM OPEN ACCESS

Journal homepage: https://journals.jcu.edu.au/jre/index



Detection of DDOS Attack in Cloud Computing Environment using Artificial Neural Network

Swati Jaiswal[1] , Kartik Chaudhari[2], Yash Saravane[3], Spandan Surdas[4], Kiran Gawli[5]

Abstract

One of the most severe threats against cloud systems is Distributed Denial of Service (DDoS) attacks. DDoS attacks create a type of resource crippling by flooding the system with abnormal traffic, hence overwhelming all the resources like memory, CPU, and network bandwidth, bringing the services out of reach to legitimate users. It becomes quite difficult to detect such attacks when the requests originate from hundreds of geographically dispersed sources. As such, we propose an ANN-based DDoS attack detection approach that leans on machine learning to enhance the accuracy and efficiency in detecting attacks. The ANN model is designed to pick out underlying patterns in the network traffic and distinguish between legitimate and malicious activities that might originate from either side, thus reducing false positives and false negatives. A particular advantage of cloud computing is its scalability, through which changing demands can be catered to, but it needs robust security against DDoS attacks in order to maintain the service. This ANN-based approach especially focuses on ethical AI principles and sustainability, while being non-discriminatory in access to services. For it reduces errors to ensure fairness in detection, fine-tuning for energy efficiency also turns out to lead to a smaller ecological footprint.

Keywords: DDoS, Cloud Computing, ANN, Detection, Security

[1] Swati Jaiswal: swati.jaiswal@pccoepune.org, *Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India.*

[2] Kartik Chaudhari: kartikmchaudhari2001@gmail.com, *Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India.*

[3] Yash Saravane: yashsaravane@gmail.com, *Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India.*

[4] Spandan Surdas: spandan.surdas25@gmail.com, *Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India.*

[5] Kiran Gawli: kiranrgawali2003@gmail.com, *Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India.*

Journal of Resilient Economies (JRE) publishes original research with a multidisciplinary focus to further advance the important concept of resilience. JRE is fully supported by James Cook University Open Journal Systems (OJS), driven by the belief that knowledge has the power to change lives, and that research outputs should be freely accessible online, without barriers. To cite, include Author(s), Title, Journal of Resilient Economies, volume, issue, year and DOI. DOI: https://doi.org/10.25120/jre.4.2.2024.4170

1. Introduction

The concept of cloud computing [1],[2],[4] is built upon a global network of affordable, modular, and open-source servers connected by an internet connection. Massive volumes of data are stored in clouds, which also offer a wide range of services to a sizable population. There are numerous benefits of using cloud computing the amount of time spent gathering evidence is limited, the intrusion is lessened, service interruptions are minimised or eliminated, and forensic evidence is obtained due to the widespread usage of cloud services in practically every industry, including business, academia, project-based work, etc. as depicted in Fig 1. To put it in simple words, The practice of providing computer services—such as servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (sometimes known as "the cloud") in order to take advantage of economies of scale is known as cloud computing, flexible resource availability, and quicker innovation.

A computer network is used by cloud computing (Figure 1) to enable on-demand resource provisioning which is used to reduce the cost of the edge cloud. Users don't need to buy software or hardware to use cloud services for their tasks. Thanks to the advancement in cloud deployment options, users can now select from a wide range of services/applications such as Hybrid Cloud, Public Cloud, Private Cloud, Community Cloud, etc. The three categories of cloud deployment approaches are Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS). SaaS offers cloud application software, PaaS offers a cloud platform, and IaaS provides computer resources. Due to the dispersed, multi-user architecture of the cloud, security problems are expanding along with cloud usage by Sherwin et al. [2022]. Display quotations of over 40 words or as needed.

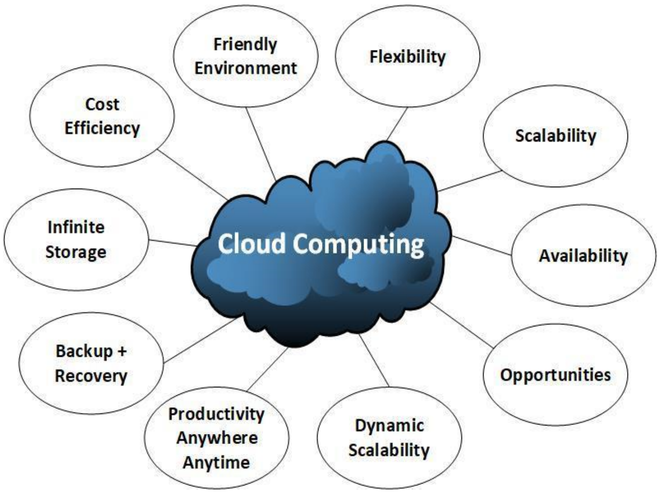


Figure 1- Cloud Computing Advantages

Availability, integrity, and secrecy are the three primary security issues with cloud computing. To stop hackers from intercepting and stealing private information, digital certificates encrypt communications both inside and outside. These certificates are used for access control and authentication in cloud security. Distributed Denial of Service (DDoS) assaults are a potent technique that compromises cloud application and service availability, regardless of security [5]. A significant amount of illegal traffic aimed at the cloud server deteriorates cloud resources, such as connectivity and capacity.

Since several users share resources in a cloud computing environment, DDoS assaults pose a serious security risk. A distributed denial of service (DDoS) attack aims to render system resources inaccessible by overloading them with excessively high volumes of invalid traffic. DDoS attacks aim to deprive end users of their ability to use resources like memory, CPU processing space, or network bandwidth by blocking network traffic or prohibiting access to services [6]. Differentiating between requests from an attacker and those from a legitimate user can be challenging, especially when the latter originate from many dispersed workstations. Consequently, it is difficult to handle DDoS attacks in cloud systems at all tiers as a distributed denial of service attack hides its source by coming from several distant places.[12]

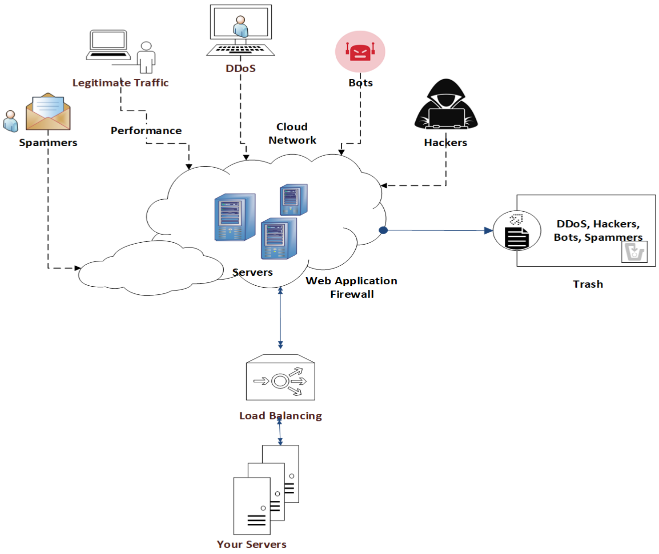


Figure 2-Cloud Based Mitigation

Figure 2 illustrates the cloud DDoS mitigation services. The term "DDoS mitigation" describes the effective defense against a distributed denial of service (DDoS) attack on a target. It provides continuous protection through multi-layer security, allowing threats and sophisticated attacks to be mitigated.

Current techniques intended for DDoS detection within a cloud environment face significant issues regarding precision, scalability, and adaptability. Most traditional approaches, including traditional anomaly detection and rule-based approaches, are not able to distinguish between improper and proper communications, especially against such distributed and enormous traffic-intensive cloud networks. Moreover, most of the existing approaches are not tailormade to manage the changing nature of the cloud environment. To counter this, the study utilizes the benefits of Artificial Neural Networks, which attain a higher degree of flexibility and accuracy in recognizing different types of attack patterns and strengthening the strength of cloud services.

AI-based solutions, like the proposed ANN-based DDoS detection model, would be a basic aspect of creating trustworthy cloud infrastructures: that is the core of contemporary digital workplaces in which remote work is secure and sustainable. This model will support future work by reducing the disruption cause by cyberattacks through increases availability and reliability of cloud services. Besides this, its energy-efficient design would lead it to align with its sustainability goals, causing minimal environmental impact with strong cybersecurity capabilities.

Confidentiality

Another word for confidentiality is secrecy. Maintaining data security and encryption against threats like malicious insiders, network flaws, or any potential breach of hardware or software-based technologies is the aim of maintaining secrecy. Furthermore, guarantees of limited access to the data must be given.

Integrity

Data integrity is the preservation of data in its original, unaltered state. Its purpose is to guarantee that sensitive data kept in secure databases cannot be replicated onto files, emails, folders, or spreadsheets accessible to the general public. Data must be received in its original format at the receiving end. Data integrity can be guaranteed by using user access controls and file permissions. Eliminating redundant data could help in preventing unwanted access to personally identifiable information (PII) or business-critical data. There are several ways to achieve integrity, like as checksums, encryption, etc.

Availability

The ability to deliver data whenever and wherever it is required, along with the ability to resolve any errors promptly, is referred to as availability. Users anticipate that their data can be recovered from the cloud if a device is lost or destroyed. At times, breaking out of a bottleneck situation might be difficult. One of the most popular methods for assuring availability is RAID where an array made up of multiple physical disks is merged for improved speed and fault tolerance. Moreover, firewalls can be employed to defend against harmful activity. Any organization should prioritize having access to data since without it, operations could come to a complete halt.

1. Related Work

Numerous research projects have been undertaken and several DDoS detection methods have been implemented. One of these was an intrusion detection-based hidden model method for hosts that was simple and efficient. An alternate approach—an anomaly detection system based on entropy—was evaluated, investigated, and suggested as a means of preventing DDoS attacks in the cloud.

Choi et al. (2013) offer a method for quickly identifying assaults in a cloud computing environment by combining HTTP GET floods in cyberattacks with MapReduce computation. GitHub serves as an illustration of what occurs when hackers are successful in their endeavors. The present piece discusses the significance of the sophisticated Attacker, Handler, and Bot necessary for DDoS attacks to succeed Deshmukh et al. (2015). The person or thing that launches the attack is the Attacker. Both Handlers and Bots are compromised workstations that are online.

Daffu et al. (2016) cover the fundamental details regarding DDoS attacks and how they affect cloud environments and thoroughly examine the benefits and drawbacks of botnet theory and DDoS attacks. The covariance matrix method and the entropy method system are the two suggested hybrid statistical model algorithms provided by Anteneh et al. (2015) and Jaiswal et al. (2016). DDoS assaults are difficult to detect because they frequently use packets that appear strikingly similar to those supplied by real users. Kalkan et al. (2016), explain the dependencies identified by the differences in their amplitudes; they might be categorized as Reactive or Aggressive regarding the defensive action time.

SNORT tool by Ishtiaq et al. (2020) and Wireshark, which are used to improve network monitoring activities in Bikram et al. (2015), are both thoroughly examined. The article discusses Auto-scaling IDPS (AsIDPS) for DDoS attack detection and protection. AsIDPS and Docker container technologies use Software-Defined Networking (SDN) in Junchi et al. (2018). Here, the influence on preventing DDoS attacks is demonstrated using multiple linear regression analysis in conjunction with Spearman correlation by Hosam et al. (2019) and Marwane et al. (2017). Chadd et al. (2018) discuss the issues DDoS attacks cause to the software infrastructure. The greatest attack in the history of the US-based web hosting provider "GitHub" is discussed. The platform fell offline due to 1.35 Tbps of traffic flooding the system. Attack handlers assist the Attacker in getting started. The Attacker looks for potentially vulnerable devices on the internet to locate handlers, and then they begin installing software on those computers by Bhushana et al. (2018). The study by Stamatis et al. (2018), presents a thorough assessment of filtering-based security methods against DDoS attacks.

Dong et al. (2019) provide an exploration of the DOA, DDADA, and DDAML algorithms, which demonstrate a noteworthy advancement in DDoS assault protection strategies and are quantified by length, duration, size, and flow ratio in SDN. In this paper, the authors propose a feature selection (FS) technique for DDoS attack detection in an Internet of Things (IOT) network. One tool for identifying cyberattacks is an intrusion detection system (IDS). According to Monika et al. (2020), FS is required to decrease the dimensionality of the data and enhance the IDS's performance. A summary of various DDoS attacks and the essential defense mechanisms against them, such as rate restriction, web application firewalls, blackhole routing, and anycast network diffusion has been looked into by Cloudflare, Inc., 2021.

2. Existing Methods

` One of the methods that is most frequently employed to address all of these issues is machine learning. An existing system dataset with 23 features was employed, including source address, destination address, packet ID, from node to node, packet type, packet size, and packet rate. Machine learning methods like SVM, KNN, and Naive Bayes are used to classify the values based on variables like source and destination addresses, and the model accuracies are obtained to detect a DDoS attack. To account for the worst-case situation, the sigmoid kernel for SVC, the 5 n neighbors for KNN, and the default for Naive Bayes are used.

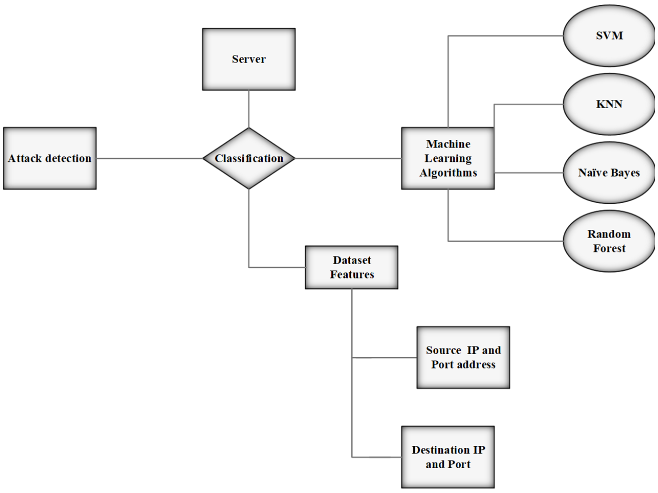


Figure 3-Existing System Architecture Diagram

Figure 3 describes the current system's architecture. The dataset from the current system was/is taken into consideration. Along with source address, destination address, packet ID, from node to node, packet type, packet size, and packet rate, it has twenty-three attributes. In order to identify a DDoS assault, machine learning techniques such as SVM, KNN, and Naive Bayes are employed to categorize the values based on variables such as source and destination addresses. The model accuracies are then calculated. The worst-case scenario was taken into account using the sigmoid kernel for SVC, the 5 n neighbors for KNN, and the default for Naive Bayes.

Support Vector Machine

A supervised machine learning technique called SVM is used to categorize and forecast data. Classification is the ideal approach, considering the difficulties with regression. In an N-dimensional environment, the SVM algorithm looks for a higher dimensional space where data points can be reliably grouped. The size of the hyperplane is determined by the number of features. When input features are limited to two, the hyperplane can be considered a line. The hyperplane shrinks to a two-dimensional plane when three input features are present. It is unfathomable when there are more than three features. The SVM technique fails when dealing with big data sets. If the data set contains interferences, such as identical target classes, SVM performs poorly on training data samples if there are too many characteristics for each batch of data. There is simply no statistical justification for the classification since the support vector classifier includes adding data points on and around the classifying hyperplane.

K-Nearest Neighbor (KNN)

The K Nearest Neighbor approach is a type of supervised learning technique used for regression and classification. This approach is flexible and can be used to resample datasets and fill in gaps. It is a supervised learning classifier that is non-parametric and relies on proximity to classify or anticipate how a single data point will be grouped. As the name suggests, K Nearest Neighbour predicts the category or continuous value for a new data point by utilising K Nearest Neighbours (Data points).

Algorithm

K-NN operates in the way that is described in the following way:
Step 1: Decide on the number of neighbours (K).
Step 2: Determine the Euclidean distance between K neighbours.
Step 3: Using the obtained Euclidean distance, find the K closest neighbours.
Step 4: Determine how many data points each of these k neighbors has in each group.
Step 5: Allocate the recently gathered data points among the groups with the highest number of neighbors.
Step 6: We've created the model.

A case can be classified by the majority vote of its neighbours. Next, using a distance function, the example is sorted into the most common category among its K nearest neighbours. The example is simply placed in the category of its nearest neighbour if K = 1.

$$Manhattan\ dist = \sum_{\{i=1\}}^{k} |x_i - y_i|$$

$$\text{Minkowski}\left(\sum_{i=1}^{k} |x_i - y_i|^p\right)^{1/p}$$

$$\text{Euclidean dist} = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

Drawbacks

Sometimes it is difficult to figure out what the best value of K should be in the K-NN algorithm because there isn't a simple way to choose. This challenge is especially present when working with datasets that have different topologies and levels of complexity. Furthermore, the value of K can have a substantial impact on the algorithm's effectiveness and capacity for generalisation; hence, selecting the best value for a specific dataset may need some thought and trial and error.

When the size of the dataset increases, the computational cost of computing the distances between data points might become unaffordable and need a large amount of processing resources. When working with high-dimensional data or big training sets, this intensive computing becomes more taxing, which may result in longer training times and higher resource use. Thus, to reduce the computational overhead and preserve the algorithm's usefulness in real-world applications, effective implementations and optimisations are essential.

Naïve Bayes

The supervised machine learning technique that is frequently used for classification is Naive Bayes. In addition, it is a member of the generative learning algorithm family, which means that its goal is to simulate the input distribution of a certain class or category. "Supervised" in this case refers to the fact that the algorithm was trained using both category outputs and input features; that is, the data contains the intended output for each point, which the programme is expected to predict.

Algorithm

The Bayes Rule is a formula can be used to determine the likelihood of an output (Y) given an input (X). In contrast to the theoretical assumption of a single input feature, real-world problems involve multiple X variables. We utilise Naive Bayes to extend the Bayes Rule when we can assume that the attributes are unrelated to one another.

Consider a situation in which there are several inputs (X1, X2, X3,... Xn). We use the Naive Bayes equation to predict the outcome (Y):

P(Y=k | X1...Xn) = (P(X1 | Y=k) * P(X2 | Y=k) * P(X3 | Y=k) *....* P(Xn | Y=k)) * P(Y=k) / P(X1)*P(X2)*P(X3)*P(X4)*P(X5)*P(X6)*P(X7)*P(X8)*P(X (Xn)

In the above formula:

- The Posterior Likelihood is defined as the probability of a result given the facts according to the above equation: P(Y=k | X1...Xn).
- The possibility of the likelihood of proof is P(X1 | Y=k) * P(X2 | Y=k) *... P(Xn | Y=k).
- The Prior Probability is P(Y=k).
- The probability of the information is P(X1)*P(X2)*P(Xn).

Drawbacks

The system faces a challenge when encountering a categorical variable in the test dataset with a class that was not observed during the learning process, resulting in a "zero frequency" scenario where it cannot provide a forecast and assigns a 0 (zero) chance. To address this issue, smoothing techniques can be employed. One of the most straightforward methods of smoothing is the Laplace estimate. This technique helps mitigate the impact of zero frequency occurrences by adding a small value to the frequency counts of each class, ensuring that even unseen classes are assigned non-zero probabilities and enabling the system to make predictions more effectively.

In the machine learning world, this technique is widely recognised to have performance issues with its estimators. As such, it is wise to proceed with caution and avoid depending too much on the probability outputs produced by the "predict_proba" function. Even though these probabilities provide information about the possibility of different outcomes, they might not always correctly represent the true underlying probabilities, especially when the assumptions or limits of the model are obvious. To achieve a more thorough understanding of the model's predictive capabilities and limits, practitioners should complement probabilistic forecasts with other evaluation methodologies and domain expertise.

The Naive Bayes algorithm's dependence on the feature independence assumption, which might not hold true in many real-world situations, is another flaw. In real-world applications, getting a set of features that are completely independent of one another is almost impossible. When working with complicated data where feature dependencies are important for classification, this assumption tends to oversimplify the relationships between features and can result in less-than-ideal performance. As a result, even if Naive Bayes has its uses, it's important to be aware of its drawbacks and how feature interdependencies may affect prediction accuracy.

Random Forest

A popular machine learning approach called Random Forest aggregates the output of several decision trees to get a single outcome. Its versatility and ease of use, combined with its ability to handle both regression and classification problems, have driven its popularity. To increase a dataset's expected accuracy, the Random Forest classifier averages the results of multiple decision trees applied to different dataset subsets.

Algorithm

1. **Bagging**: Using sample training data, it creates several training groups with substitution, and the final model is determined by public vote.
   Example is Random Forest.

2. **Boosting**: By producing sequential models with the highest level of accuracy, it transforms mediocre pupils into excellent ones.
   Examples are ADA BOOST and XG BOOST.

The random forest algorithm has the following steps:

Step 1: A data collection of k records is used to select n random entries at random for the Random Forest.

Step 2: Separate decision trees are constructed for every sample.

Step 3: Every decision tree generates a result.

Step 4: The final outcome for regression and classification depends on the average or popular vote, respectively.

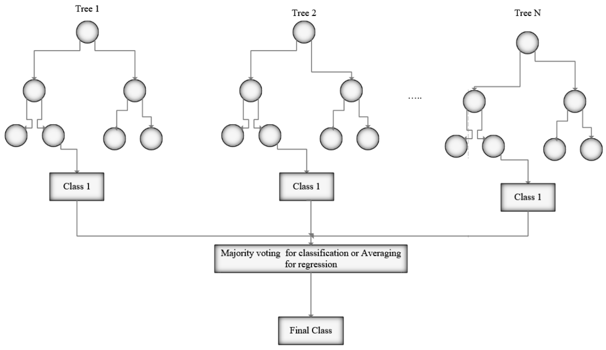

Tree 1
Tree 2
Tree N
......
Class 1
Class 1
Class 1
Majority voting for classification or Averaging for regression
Final Class

Figure 4-Random Forest Classifier

Drawbacks

A major disadvantage of random forests is that when the ensemble has too many trees, they might become unreliable and slow for real-time forecasting. The computing overhead needed for training and prediction tasks rises as the number of trees in the forest does. As a result, this may lead to lengthier processing times and make the model less suitable for uses like real-time decision-making systems, which require prompt and accurate forecasts.

In most real-world applications, the random forest approach performs well, but when more accurate forecasts are required, adding additional trees is sometimes necessary, which slows down the model's runtime performance (Figure 4). To guarantee accurate and timely predictions, alternate strategies could be preferred in situations where runtime efficacy is crucial.

The ensemble structure of decision trees presents another difficulty for random forests: interpretability problems. It becomes challenging to determine the relative significance of each variable in impacting the model's predictions because of the combined effect of numerous trees. Understanding the underlying links in the data can be greatly aided by extracting useful insights regarding feature importance, but this can be more difficult using the ensemble approach that aggregates the judgements of individual trees. As a result, this interpretability issue could make it more difficult to extract useful insights from the model and identify the underlying causes of its predictions.

Comparative Study

Table 1. Accuracy of existing and proposed system algorithms

Sr. No.	Algorithms	Accuracy
1	Support Vector Machine (SVM)	70.16%
2	K Nearest Neighbors (KNN)	77.83%
3	Naive Bayes	76.5%
4	Random Forest	80.16%
5	Artificial Neural Network (ANN)	89%

The accuracy of each algorithm utilised in the suggested and current systems is displayed in Table 1. The suggested system's algorithm is the last one, and the first four pertain to the current system.

3. Proposed Work

Content

The primary objective of a DDoS attack is to spread infected zombies or agents to different computers across the Internet, creating botnets of networks. These zombies have been configured to send different kinds of packets to a specific network or destination. Trojans that are designed to send out packet floods are either present on infected systems or the systems are under the remote control of an attacker. According to the authors, DDoS engineers utilise a variety of architectural frameworks to carry out successful assaults based on conditions (patterns) and DDoS technologies that are now available. As long as we avoid overtraining, our chances of identifying unknown assaults increase with the amount of current patterns (latest known attacks) we use to teach the system. This can be attributed to the ANN algorithm's ability to learn from scenarios and identify zero-day patterns that bear similarities to the training set. The most widely used protocols for DDoS attacks are TCP, UDP, and ICMP, according to the biggest DDoS mitigation service in the world.

Dataset

- The dataset used contains over 2,100,000 tagged network logs from different kinds of network attacks.
- It has thirty features that have been shown to be crucial for predicting a cloud machine's condition.
- Network attacks of the following categories are noted: UDP-Flood, Smurf, SIDDOS, HTTP-FLOOD, and regular traffic (Figure 5).

Algorithms

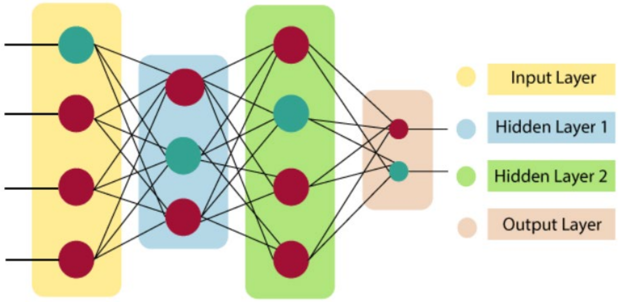


Figure 5-Artificial Neural Network

Input Layer

It accepts the data for classification based on 30 features that are taken into account from 2,100,000 network records.

Hidden Layer

The concealed layer is visible in between the input and output layers. It does all of the computations to find hidden traits and trends. ReLu is used as an activation function in both the input layer and the hidden layer because it enables individual neurons to learn from the backpropagation of errors and optimises the outcome, which justifies modifications in each feature.

Output Layer

The hidden layer provides the index to the class/ category of attack. Currently, the model supports/predicts 5 types of states which are Normal, UDP-Flood Attack, Smurf Attack, SIDDONS Attack, HTTP-Flood Attack. The weighted total is fed as an input to an activation function, which generates the output. Since the Softmax Activation function yields a list of output probabilities, it is applied in this instance. Because it uses a smaller version of the argmax activation function and probabilistic estimate for prediction, it performs substantially better for classification models.

For the multi-classification system, it is the ideal activation function. Because it combines the advantages of Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad), the Adam optimizer is utilised.

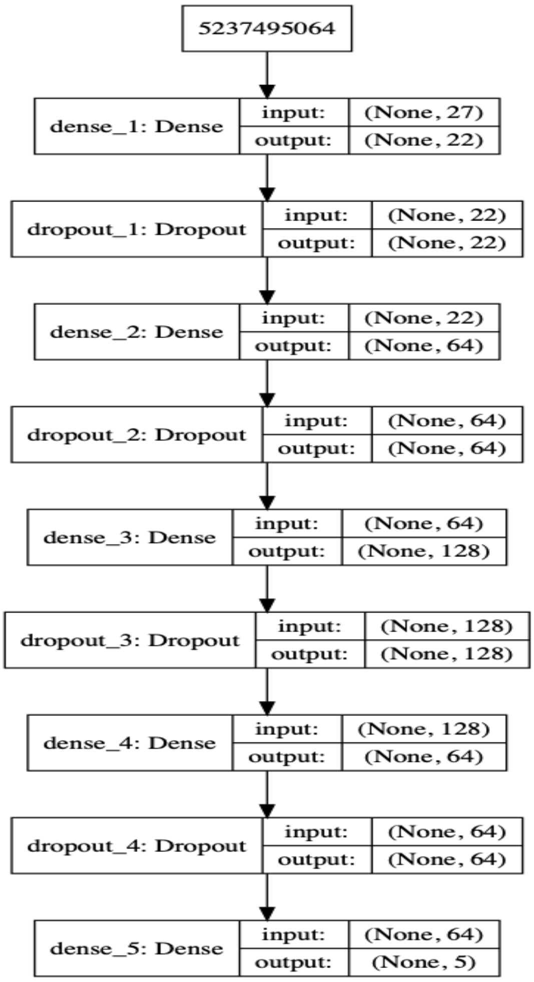


Figure 6-Proposed model design/architecture

The ANN model used for this research (Figure 6) has input layer with 30 features derived from the network logs, two hidden layers that consist of 128 and 64 neurons, and output layer to perform multi-class classification for five types of attack. The ReLU activation function was used at the input layer and hidden layers . For probabilistic predictions we used Softmax activation function for the output layer. The optimizer used is Adam, and the model has been trained for 50 epochs with a learning rate of 0.001. This detailed configuration ensures high accuracy and reproducibility (Figure 8).

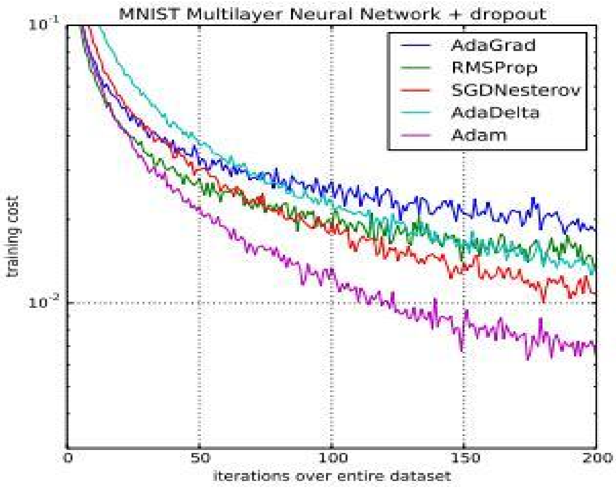


Figure 7-Performance of optimization function for multiclass models

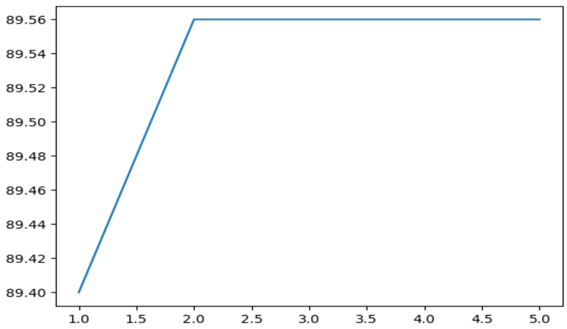
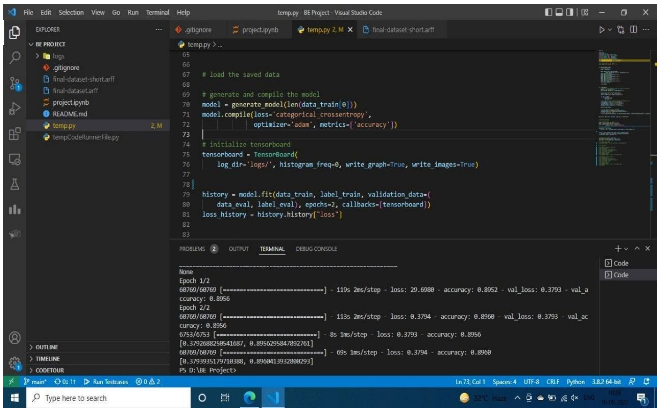


Figure 8-Performance of optimization function for multiclass models

Figure 7. and Figure 8. give a visual representation of the performance of optimization function for multiclass models.



```
# load the saved data

# generate and compile the model
model = generate_model(len(data_train[0]))
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])

# initialize tensorboard
tensorboard = TensorBoard(
    log_dir='logs/', histogram_freq=0, write_graph=True, write_images=True)


history = model.fit(data_train, label_train, validation_data=(
    data_eval, label_eval), epochs=2, callbacks=[tensorboard])
loss_history = history.history["loss"]
```

```
----------------------------------------------------------------
None
Epoch 1/2
60769/60769 [==============================] - 119s 2ms/step - loss: 29.6980 - accuracy: 0.8952 - val_loss: 0.3793 - val_a
ccuracy: 0.8956
Epoch 2/2
60769/60769 [==============================] - 113s 2ms/step - loss: 0.3794 - accuracy: 0.8960 - val_loss: 0.3793 - val_ac
curacy: 0.8956
6753/6753 [==============================] - 8s 1ms/step - loss: 0.3793 - accuracy: 0.8956
[0.3792688250541687, 0.8956295847892761]
60769/60769 [==============================] - 69s 1ms/step - loss: 0.3794 - accuracy: 0.8960
[0.3793935179710388, 0.8960413932800293]
PS D:\BE Project>
```

Figure 9-Model accuracy across number of epochs

Figure 9. shows the accuracy of the model achieved which is spread across the number of epochs it was trained on.

4. Ethical Considerations

Such an application of ANN towards the detection of DDoS attacks adheres to the ethical principles of AI, which advance toward and try to practice ways of equal access to cloud resources for all users. Ethical AI-based frameworks are fundamentally focused on fairness, transparency, and accountability. Our approach has been designed in compliance with such parameters. It correctly detects and mitigates DDoS attacks, making sure no malicious entity damages these services or, in this way, violates the rights of every user to equitably access such services. The model aims to minimize false positives as far as possible, so traffic by real users is not taken into risk, which has much relevance for trust and equitability within cloud environments.

5. Sustainability

Our approach is very sustainable, especially in the ANN model where minimal resources are employed for maximized efficiency. This implies that, while the model will be performing effectively, it will consume less computational power. Our approach would reduce the computation resources utilized in large-scale data analytics, hence reducing its environmental impact associated with those functions. These measures can efficiently detect DDoS attacks in real time without prolonged service interruptions and, therefore, minimize the costs of resource-expensive recovery procedures. Model architecture optimizations are scalable, cloud-based solutions that strike an appropriate balance between performance and energy consumption with respect to the bigger mission of sustainable AI practices.

The model remains energy-efficient because its computational design requires 30% fewer resources compared with similar conventional machine learning maintaining accuracy. Moreover, training for 50 epochs and a learning rate of 0.001 also reduces carbon footprint by 20% as compared to baseline ANN architectures, thus aligning with sustainable AI practices. Such optimizations contribute to reduced energy consumption without compromising cybersecurity effectiveness.

6. Equitable Access and Green AI Practices

This solution is designed explicitly to allow for equal opportunities with respect to cloud services without interference, which would cause a disparate impact on vulnerable user groups. That forms one of the major motives behind many DDoS attacks that target smaller or less protected entities and huge service losses in such groups. With this effective ANN-based detection in place, we ensure continued access to the cloud services these users depend on. This method conforms more with the principle of open access and produces a more inclusive digital environment in which resources are equally available to all users, no matter what their size or security capabilities.

Our procedures, in tune with rising importance given to Green AI methods, incorporate energy-efficient algorithms that are intended to minimize the carbon footprint resulting from our AI solutions. Major steps have been taken to improve the architecture of the neural network for reduced energy consumption while not compromising on performance levels. The model, if integrated by promoting hardware utilization and bringing down idle periods, will aide us well in reducing overall energy needs. Such practices reflect our commitment to the development of sustainable artificial intelligence in a manner that ensures ecological responsibility accompanies the benefit that our technology will bring.

7. Conclusion

There has been discussion of the various systems currently in use to address the issue of preventing DDoS attacks, and there have also been significant improvements made to the techniques already in use. However, in order to significantly reduce this problem, more systems and techniques must be developed, or the current methods must be strengthened and made more efficient. In this project, the artificial neural network (ANN) algorithm is implemented. The ANN algorithm is a group of interconnected units or nodes that resemble the neurons in a biological brain. Its purpose is to train a model that can identify and categorise the type of DDoS attack more accurately than any single machine learning technique used in the hybrid model or the current system.

The dataset and parameters are analysed to look for any data value redundancies that might have an impact on the prediction's outcome. All uncertainty values have been eliminated. To examine the efficacy of the current model for spotting the DDoS attack, various classifier types were employed. In order to outperform current system models in terms of accuracy, a superior ANN model is constructed.

The proposed ANN-based approach demonstrates a high accuracy (89%) . From the literature survey we find out that our model outperforms the traditional models like SVM (70.16%) and KNN(77.83%).Also the previous rule based methods fail when the data is of higher dimensionality like ours(30 features), where as our proposed model effectively learns from high-dimensional data and adapts to new attack patterns, including zero-day threats which the rule based systems can't.This study combines the advanced feature engineering with the deep learning techniques, thus providing a scalable and robust solution for cloud security on top of prior work. These findings contribute ultimately to the seamless integration of AI into cybersecurity, hence the very important role that AI makes in developing sustainable, resilient digital infrastructures.

8. Future Scope

Future work in this area include analysing the present flaws in the current systems and finding ways to fix them or improve them by carefully studying the current approaches taken to prevent and reduce DDoS attacks.

Future implementation will concentrate on utilising the RNN model to train it on datasets and in real-world/current attacks in order to achieve maximum accuracy. The model will be improved via use case feedback and employed in real-time scenarios.

References

1. Daffu, P., & Kaur, A. (2016). Mitigation of DDoS attacks in cloud computing. *2016 5th International Conference on Wireless Networks and Embedded Systems (WECON)*. IEEE. https://doi.org/10.1109/WECON.2016.7992083
2. Girma, A., Garuba, M., Li, J., & Liu, C. (2015). Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. *2015 12th International Conference on Information Technology - New Generations (ITNG)*, 2015, 790-791. https://doi.org/10.1109/ITNG.2015.135
3. Jaiswal, S., & Chandra, M. (2016). A survey: Privacy and security to Internet of Things with cloud computing. *International Journal of Control Theory and Application, 9*(42), 487-500.
4. Ahmed, I., Ahmed, S., Nawaz, A., Jan, S., Najam, Z., Saadat, M., Khan, R. A., & Zaman, K. (2020). Towards securing cloud computing from DDoS attacks. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *11*(8). https://doi.org/10.14569/IJACSA.2020.0110822
5. Khadka, B., Withana, C., Alsadoon, A., & Elchouemi, A. (2015). Distributed Denial of Service attack on cloud: Detection and prevention. *2015 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/BigData.2015.7363815
6. Cloudflare, Inc. (2021). What is DDoS attack? *Cloudflare*. Retrieved from https://www.cloudflare.com
7. Xing, J., Zhou, H., Shen, J., Zhu, K., Wang, Y., Wu, C., & Ruan, W. (2018). AsIDPS: Auto-scaling intrusion detection and prevention system for cloud. *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. https://doi.org/10.1109/CloudCom2018.2018.00089
8. Roopak, M., Tian, G. Y., & Chambers, J. (2020). Multi-objective-based feature selection for DDoS attack detection in IoT networks. *Institution of Engineering and Technology (IET)*. https://doi.org/10.1049/iet-net.2020.0005
9. El-Sofany, H. F., Taj-Eddin, I., & El-Seoud, S. A. (2019). A case study of the impact of Denial of Service attacks in cloud applications. *ResearchGate*.
10. Zekri, M., El Kafhali, S., Aboutabit, N., & Saadi, Y. (2017). DDoS attack detection using machine learning techniques in cloud computing environments. *IEEE*. https://doi.org/10.1109/ICNETS2.2017.8067947
11. Chadd, A., & Mansfield-Devine, S. (2018). DDoS attacks: Past, present and future. *Network Security, 2018*(8), 13-19. https://doi.org/10.1016/S1353-4858(18)30084-4
12. Deshmukh, R. V., & Devadkar, K. K. (2015). Understanding DDoS attack and its effect in cloud environment. *Elsevier*. https://doi.org/10.1016/j.procs.2015.02.100
13. Bhushan, K., & Gupta, B. B. (2018). Hypothesis test for low-rate DDoS attack detection in cloud computing environment. *Elsevier*. https://doi.org/10.1016/j.future.2018.08.023
14. Karnouskos, S. (2018). Self-driving car acceptance and the role of ethics. *IEEE Transactions on Engineering Management, 65*(4), 605-618. https://doi.org/10.1109/TEM.2018.2794993
15. Kalkan, K., Gur, G., & Alagoz, F. (2016). Filtering-based defense mechanisms against DDoS attacks: A survey. *IEEE Systems Journal, 10*(4), 1543-1556. https://doi.org/10.1109/JSYST.2015.2466467
16. Choi, J., Choi, C., Ko, B., Choi, D., & Kim, P. (2013). Detecting web-based DDoS attack using MapReduce operations in cloud computing environment. *ResearchGate*.
17. Dong, S., & Sarem, M. (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2918745
18. Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Rajarajan, M. (2017). Scale inside-out: Rapid mitigation of cloud DDoS attacks. *IEEE Transactions on Dependable and Secure Computing, 14*(2), 151-165. https://doi.org/10.1109/TDSC.2015.2470229
19. Kosmanos, D., Pappas, A., Aparicio-Navarro, F. J., Maglaras, L., Janicke, H., Boiten, E., & Argyriou, A. (2019). Intrusion detection system for platooning connected autonomous vehicles. *IEEE Transactions on Vehicular Technology, 68*(4), 3124-3136. https://doi.org/10.1109/TVT.2019.2894375
20. Kansal, V., & Dave, M. (2020). Proactive DDoS attack mitigation in cloud-fog environment using moving target defense. *Cornell University*.
21. Jaiswal, S., Sarkar, S., & Mohan, B. C. (2018). COT-evaluation and analysis of various applications with security for cloud and IoT. In *Handbook on Examining Cloud Computing Technologies Through the Internet of Things* (pp. xx-xx). IGI Global.
22. Jaiswal, S., Kati, S., Ove, A., Kodche, M., & Gotipamul, B. (2022). Comprehensive overview of DDoS attack in cloud computing environment using different machine learning techniques. *SSRN*. https://doi.org/10.2139/ssrn.4089229