# Distributed Denial of Service (DDOS) Attack Detection in Cloud Environments Using Machine Learning Algorithms

Sathish Polu[1], Dr.V.Bapuji[2]
[1]*Research Scholar, Bir Tikendrajit University*
[2]*Research Supervisor, Bir Tikendrajit University*

**Abstract: As cloud computing becomes an integral component of modern information technology infrastructures, the vulnerability of cloud environments to Distributed Denial of Service (DDoS) attacks poses a significant threat to the availability and performance of cloud-hosted services. This research addresses the imperative need for effective DDoS attack detection in cloud environments by leveraging machine learning algorithms. Through a comprehensive exploration of the existing literature, this study synthesizes knowledge on DDoS attack characteristics, the unique challenges posed by cloud environments, and the role of machine learning in enhancing cybersecurity. The research concludes by summarizing key findings and emphasizing the contributions to the field. Ultimately, this study underscores the critical role of machine learning in fortifying cloud security against DDoS attacks, offering a foundation for further advancements in cloud cybersecurity.**

Background:

In the contemporary landscape of information technology, the proliferation of cloud computing has revolutionized the delivery and accessibility of services, providing unparalleled scalability and efficiency. However, this paradigm shift towards cloud-based architectures brings with it an escalated risk, particularly in the form of Distributed Denial of Service (DDoS) attacks. DDoS attacks, characterized by their ability to inundate a target system with an overwhelming volume of traffic, pose a severe threat to the availability, performance, and reliability of cloud-hosted services.

The paramount challenge in safeguarding cloud environments against DDoS attacks lies in the dynamic and distributed nature of these assaults. Traditional security measures, while effective in conventional network settings, often prove inadequate when confronted with the complexities of cloud infrastructures. The need for sophisticated and adaptive detection mechanisms becomes evident as the cyber threat landscape evolves.

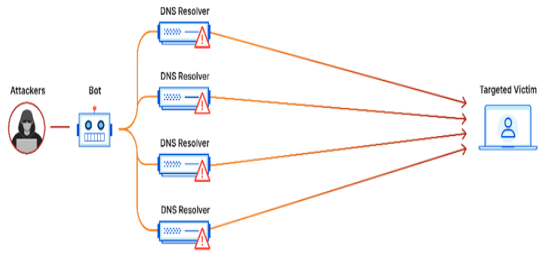
This research aims to address this imperative need by exploring the integration of machine learning algorithms for the detection of DDoS attacks in cloud environments. As DDoS attacks become more intricate and cloud services more prevalent, the traditional signature-based approaches prove insufficient. Machine learning, with its capacity to discern patterns and anomalies in large datasets, emerges as a promising avenue for enhancing the security posture of cloud-based systems.

The significance of this research lies not only in its response to the current challenges posed by DDoS attacks but also in its anticipation of future threats. By harnessing the power of machine learning algorithms, this study seeks to provide a proactive and adaptive defence mechanism against the evolving landscape of DDoS attacks in cloud environments.

In the subsequent sections, this paper will delve into the existing literature on DDoS attacks, elucidate the challenges posed by such attacks in cloud environments, and present a comprehensive overview of machine learning algorithms as applied to cybersecurity. The methodology, experimental results, and discussions will follow, offering insights into the effectiveness of machine learning in fortifying cloud security against DDoS attacks. Ultimately, this research contributes to the growing body of knowledge aimed at ensuring the resilience and reliability of cloud computing in the face of evolving cybersecurity threats.

Distributed Denial of Service (DDoS) attacks:

Distributed Denial of Service (DDoS) attacks manifest in various forms, each targeting different aspects of network infrastructure or applications. Understanding the common types of DDoS attacks and their potential impact on cloud services is crucial for implementing effective mitigation strategies.




1. Volumetric Attacks:
- Description: Overwhelm a target's network bandwidth by flooding it with a massive volume of traffic.
- Impact on Cloud Services:
- Bandwidth Saturation: Exhausts available bandwidth, causing service disruption.
- Collateral Damage: Affects not only the targeted service but also other services sharing the same network infrastructure in a cloud environment.
2. TCP/IP Protocol Attacks:
- Description: Exploit vulnerabilities in the TCP/IP protocol stack to consume server resources.
- Impact on Cloud Services:
- Connection Exhaustion: Consumes server resources by exhausting available connections, making it difficult for legitimate users to connect.
- Server Resource Depletion: Strains servers by exploiting inefficiencies in the protocol stack.
3. Application Layer Attacks:
- Description: Target specific applications or services rather than network infrastructure.
- Impact on Cloud Services:
- Service Degradation: Impairs the functionality of specific applications or services hosted in the cloud.
- Resource Exhaustion: Exploits vulnerabilities in application layers, causing servers to exhaust resources.
4. DNS Amplification Attacks:
- Description: Exploit DNS servers to amplify the volume of attack traffic directed at the target.
- Impact on Cloud Services:
- DNS Infrastructure Overload: Overloads DNS servers, affecting the resolution of domain names for cloud services.
- Reflection Attack: Utilizes third-party DNS servers to amplify the attack, making attribution and mitigation challenging.
5. UDP Reflection/Amplification Attacks:
- Description: Exploit vulnerable UDP services to amplify attack traffic.
- Impact on Cloud Services:
- Amplified Traffic: Causes a significant increase in inbound traffic, overwhelming cloud infrastructure.
- Service Disruption: Disrupts services dependent on UDP, such as voice over IP (VoIP) or online gaming.
6. SSL/TLS Attacks:
- Description: Target the secure communication protocols (SSL/TLS) to exhaust server resources.
- Impact on Cloud Services:
- Resource Intensive: SSL/TLS handshakes can be resource-intensive, affecting the performance of secure cloud services.
- Service Unavailability: Disruption of secure communication channels within the cloud environment.

Each type of DDoS attack poses unique challenges for cloud services. Effective detection and mitigation strategies need to consider the specific characteristics and impact of each attack type, especially in the dynamic and shared nature of cloud environments. As cloud adoption continues to grow, the importance of robust DDoS protection mechanisms becomes increasingly critical for ensuring the availability and reliability of cloud-hosted services.

Deploying and maintaining secure cloud environments introduces a unique set of challenges due to the shared and dynamic nature of cloud infrastructure. Here are some key challenges associated with ensuring security in cloud environments:

1. Data Security and Privacy:
- Challenge: The shared nature of cloud storage and processing raises concerns about data security and privacy, especially when sensitive data is stored alongside that of other customers.
- Implications: Unauthorized access, data breaches, and compliance violations can occur if robust data security measures are not implemented.
2. Multi-Tenancy Risks:

- Challenge: Cloud services often operate on a multi-tenant model, where multiple customers share the same physical infrastructure.
- Implications: Security risks arise from the potential for one tenant's activities impacting the security and performance of other tenants' data and applications.

3. Network Security:
- Challenge: Cloud environments rely on virtualized networks, introducing challenges in securing data in transit and preventing unauthorized access.
- Implications: Vulnerabilities in network configurations may lead to data interception, manipulation, or unauthorized access.

4. Identity and Access Management:
- Challenge: Managing identities, access controls, and permissions in dynamic cloud environments can be complex.
- Implications: Inadequate access controls may result in unauthorized access, data leaks, or compromised accounts.

5. Compliance and Legal Concerns:
- Challenge: Meeting regulatory compliance requirements across different geographic regions and industries can be challenging in cloud environments.
- Implications: Non-compliance may result in legal consequences, fines, and damage to an organization's reputation.

6. Visibility and Monitoring:
- Challenge: Gaining comprehensive visibility into cloud resources and activities is challenging due to the scale and dynamic nature of cloud environments.
- Implications: Inadequate monitoring may lead to delayed detection of security incidents or vulnerabilities.

7. Distributed Denial of Service (DDoS) Attacks:
- Challenge: Cloud services are susceptible to DDoS attacks, which can overwhelm infrastructure and impact the availability of services.
- Implications: Downtime, service disruptions, and financial losses may result from successful DDoS attacks.

8. Incident Response and Forensics:
- Challenge: Cloud environments may lack standardized procedures for incident response and forensics.
- Implications: Delays in identifying, containing, and recovering from security incidents may occur without well-defined processes.

9. Dependency on Service Providers:
- Challenge: Organizations depend on cloud service providers for security measures, and the level of control varies.
- Implications: Limited control may lead to uncertainties regarding the effectiveness of security measures and incident response capabilities.

10. Dynamic Resource Provisioning:
- Challenge: The dynamic allocation and de-allocation of resources in cloud environments can complicate security configurations and policies.
- Implications: Misconfigurations or lapses in security policies during resource provisioning may expose vulnerabilities.

Addressing these challenges requires a holistic approach, involving a combination of robust security policies, encryption, access controls, regular audits, and ongoing security awareness and training for personnel. As the cloud landscape continues to evolve, organizations must stay proactive in adapting their security strategies to mitigate emerging threats and challenges.

Machine learning algorithms:

Several machine learning algorithms are commonly employed in Distributed Denial of Service (DDoS) detection. These algorithms leverage various techniques to analyse network traffic patterns, identify anomalies, and distinguish between normal and malicious activities. Here's an overview of some commonly used machine learning algorithms in the context of DDoS detection:

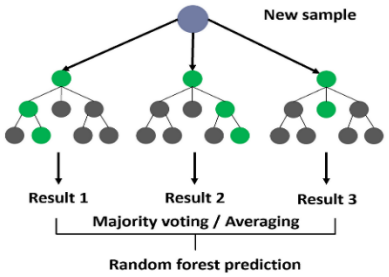
1. Random Forest:

An ensemble learning method that builds multiple decision trees during training and combines their outputs for classification.Robust against overfitting, handles large datasets, and provides insights into feature importance.Computationally expensive during training. Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. The fundamental idea behind Random Forest lies in combining the predictive power of multiple decision trees to achieve a more robust and accurate model.

Ensemble learning, of which Random Forest is a prime example, involves the combination of multiple models to improve overall performance. In the case of Random Forest, the base learners are decision trees.

Decision trees are simple models that recursively partition the feature space, making decisions based on feature values.

One key aspect of Random Forest is the use of bootstrap sampling, where each decision tree is trained on a random subset of the training data. This process introduces diversity among the trees, preventing the model from overfitting to the peculiarities of the training set. Additionally, at each split in a decision tree, only a random subset of features is considered, further enhancing the diversity.

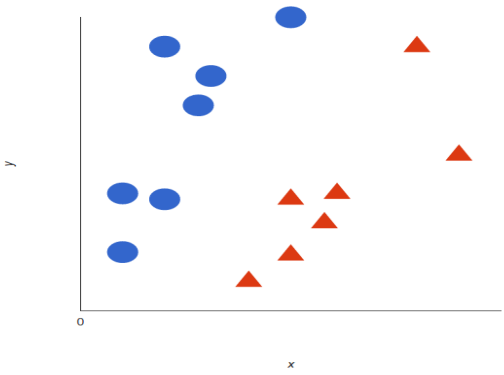



The final prediction of the Random Forest is determined through a combination of the predictions of individual trees. For classification tasks, a majority voting scheme is employed, while regression tasks involve averaging the predictions. This ensemble approach helps to mitigate the weaknesses of individual trees and improves the model's overall generalization to unseen data.

The out-of-bag (OOB) error is a unique feature of Random Forests. Since each tree is trained on a different subset of the data, the samples not included in the training set for a particular tree can be used to estimate the model's accuracy. This provides a built-in validation measure without the need for a separate validation set.Random Forest is highly parallelizable, making it computationally efficient. Each tree can be trained independently, allowing for parallel processing and scalability to large datasets.In terms of hyperparameters, Random Forest provides flexibility in tuning the number of trees in the forest, the depth of each tree, and the number of features considered at each split. Proper tuning of these hyperparameters is crucial to achieving optimal model performance.While individual decision trees are interpretable, the ensemble nature of Random Forests makes them less interpretable as a whole. However, feature importance scores can be extracted to understand the relative contribution of each feature to the model's predictions.In practice, Random Forest has proven to be a versatile and powerful tool for various machine learning tasks, excelling in scenarios where data is complex, noisy, or contains irrelevant features. Its robustness, ease of use, and ability to handle different types of data make it a popular choice in the machine learning community.

2. Support Vector Machines (SVM):

Classifies data points by finding the hyperplane that best separates different classes in a high-dimensional space.Effective in high-dimensional spaces, versatile due to various kernel options.Sensitive to noisy data, may suffer from a high computational cost.Support Vector Machines (SVM) is a powerful and versatile machine learning algorithm used for both classification and regression tasks. The theoretical foundations of SVM are rooted in the field of statistical learning theory and optimization.




SVM operates by finding the hyperplane that best separates different classes in the feature space. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class. These nearest points are called support vectors, and they play a crucial role in defining the optimal decision boundary.

In the case of a binary classification problem, where there are two classes, the hyperplane can be represented as a line in two-dimensional space or a plane in three-dimensional space. In higher-dimensional spaces, the hyperplane becomes a hyperplane.

SVM can handle both linearly separable and non-linearly separable data. For non-linearly separable data, SVM employs the kernel trick. The kernel function allows the algorithm to implicitly map the input data into a higher-dimensional space, where a hyperplane can effectively separate the classes. Common kernel functions include the linear kernel,

polynomial kernel, and radial basis function (RBF) kernel.The optimization problem associated with SVM involves finding the weights and biases of the hyperplane that minimize a cost function while satisfying certain constraints. The cost function penalizes misclassifications and is subject to the constraint that the margin should be maximized. This leads to a convex optimization problem that can be efficiently solved using various optimization techniques.

SVM is known for its ability to generalize well to unseen data and is less prone to overfitting, especially in high-dimensional spaces. The concept of the margin provides a form of regularization, and SVM tends to focus on the most critical data points (support vectors) when defining the decision boundary.In addition to binary classification, SVM can be extended to handle multi-class classification by combining multiple binary classifiers. Common strategies include one-vs-one and one-vs-all approaches.SVM has been widely used in various domains, including image classification, text classification, bioinformatics, and finance. Its effectiveness in high-dimensional spaces and its ability to handle complex relationships between features make it suitable for a range of applicationsDespite its strengths, SVM's performance can be sensitive to the choice of hyperparameters, such as the regularization parameter (C) and the choice of the kernel function. Proper tuning of these parameters is essential for achieving optimal performance. In summary, Support Vector Machines offer a robust and versatile framework for solving classification and regression problems. The emphasis on maximizing the margin and the ability to handle non-linear relationships through kernel functions contribute to SVM's popularity in the machine learning community.

3. Neural Networks:

A set of algorithms inspired by the human brain's structure, designed to recognize patterns.Suitable for complex, nonlinear relationships, effective in capturing intricate patterns. Requires substantial computational resources, may be prone to overfitting.

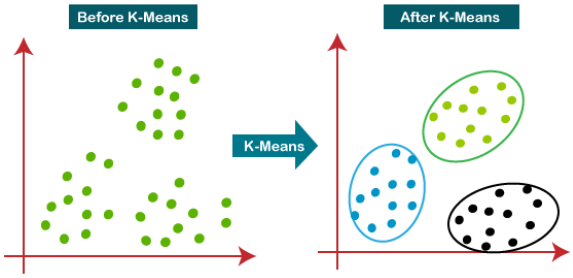
Neural networks, a fundamental concept in machine learning and artificial intelligence, are computational models inspired by the structure and function of the human brain. Here's an overview of the theory behind neural networks without breaking it down into points:Neural networks consist of interconnected nodes organized into layers. The three main types of layers are the input layer, hidden layers, and output layer. Information flows through the network from the input layer, through the hidden layers, to the output layer.Each node, or neuron, in a neural network receives input, processes it through an activation function, and produces an output. The activation function introduces non-linearity, enabling the network to learn complex relationships in the data. Connections between neurons are assigned weights, which determine the strength of the connection. During training, these weights are adjusted to minimize the difference between the predicted output and the actual output.The process of passing input data through the network to generate an output is called feedforward. Backpropagation is the learning algorithm used to train the network. It involves calculating the error between the predicted and actual outputs, propagating this error backward through the network, and adjusting the weights to minimize the error.The loss function quantifies the difference between the predicted and actual outputs. During training, the goal is to minimize this loss. Different tasks (classification, regression) and architectures may require different loss functions.Neural networks "learn" by adjusting their weights based on the training data. The learning process involves iteratively presenting examples to the network, computing errors, and updating weights. This process continues until the network achieves satisfactory performance.

Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). These functions introduce non-linearity, allowing the network to model complex patterns and relationships.Neural networks with multiple hidden layers are referred to as deep neural networks. Deep learning has gained prominence due to its ability to automatically learn hierarchical representations from data.Neural networks can be prone to overfitting, where they perform well on training data but poorly on unseen data. Regularization techniques, such as dropout and weight regularization, help mitigate overfitting.Convolutional Neural Networks (CNNs) are specialized for image and grid-like data, while Recurrent Neural Networks (RNNs) are designed for sequence data. Both architectures have played key roles in the success of deep learning.Neural networks find applications in a wide range of fields, including image and speech recognition, natural language processing, robotics, healthcare, and more. Their

versatility and ability to automatically learn features make them powerful tools in various domains.In summary, neural networks serve as the foundation for modern machine learning and have demonstrated remarkable success in solving complex problems across diverse domains. The interplay of interconnected nodes, weights, and activation functions enables neural networks to learn and represent intricate patterns in data.

4. K-Means Clustering:

Divides data into 'k' clusters based on similarity.Simple and computationally efficient.Requires the number of clusters ('k') to be specified, sensitive to initial cluster centres.K-Means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K distinct, non-overlapping subsets (clusters). The primary goal of K-Means is to group data points into clusters in such a way that the sum of squared distances (Euclidean distances) between data points and the centroid of their assigned cluster is minimized.The algorithm starts by randomly initializing K centroids, where K is the pre-defined number of clusters. Centroids are the centre points of clusters.For each data point in the dataset, the algorithm assigns it to the cluster whose centroid is closest. The closeness is typically measured using Euclidean distance, but other distance metrics can also be used.After assigning all data points to clusters, the algorithm updates the centroids by computing the mean of all data points in each cluster. The new centroids become the centre points for the next iteration.



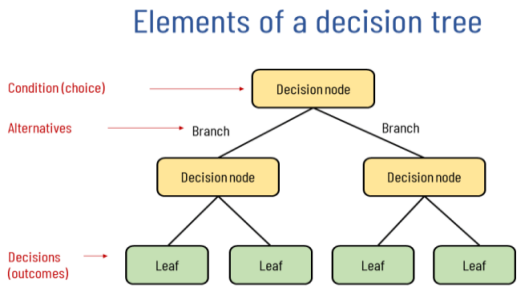

The assignment and update steps are repeated iteratively until convergence. Convergence occurs when the centroids no longer change significantly between iterations or when a specified number of iterations is reached.Choosing the appropriate value for K is a critical aspect of using K-Means. It is often determined based on domain knowledge, exploration of the data, or using methods like the elbow method, silhouette analysis, or cross-validation.The final clustering outcome can be sensitive to the initial placement of centroids. To mitigate this, the algorithm is often run multiple times with different initializations, and the best result is chosen.K-Means relies on the Euclidean distance metric to measure the dissimilarity between data points and centroids. It assumes that clusters are spherical and equally sized.K-Means has limitations, including sensitivity to outliers, the requirement to specify the number of clusters in advance, and the assumption that clusters are isotropic (circular/spherical).K-Means is widely used in various fields such as customer segmentation, image compression, anomaly detection, and document clustering.K-Means can scale well with large datasets, but its performance may deteriorate with high-dimensional data.In summary, K-Means clustering is a straightforward and efficient algorithm for partitioning data into distinct clusters based on similarities. Despite its simplicity, it has proven to be effective in various applications and serves as a foundational method in unsupervised learning.

5. Decision Trees:

A tree-like model of decisions, with each node representing a decision based on features.Intuitive to understand, computationally efficient, and handles both numerical and categorical data.Prone to overfitting, may not capture complex relationships.Decision Trees are a fundamental machine learning algorithm used for both classification and regression tasks. A Decision Tree is a hierarchical tree-like structure consisting of nodes, where each node represents a decision or a test on a feature, each branch represents an outcome of the decision, and each leaf node represents the final predicted outcome.Decision nodes in a Decision Tree correspond to tests on a particular feature. These tests split the data into subsets based on the feature's values.Leaf nodes are the terminal nodes of the tree and represent the final predicted outcomes. In a classification tree, each leaf node is associated with a class label, while in a regression tree, the leaf nodes contain a numerical value.The decision on which feature to split on at each decision node is determined by a splitting criterion. Common criteria include Gini impurity for classification tasks and mean squared error for regression tasks. The goal is to make splits that result in more homogeneous subsets.The process

of building a Decision Tree involves recursively partitioning the data based on the selected features until a stopping criterion is met. This could be a maximum depth, a minimum number of samples in a leaf node, or other criteria.Pruning is a technique used to prevent overfitting. It involves removing branches or nodes from the tree that do not contribute significantly to improving its performance on unseen data.To make a prediction for a new data point, it traverses the tree from the root to a leaf node based on the feature values of the data point. The predicted outcome is the majority class (classification) or the mean value (regression) of the instances in the leaf node.Decision Trees are inherently interpretable. The decision-making process is transparent, making it easy to understand and explain how the model arrived at a particular prediction.Decision Trees can handle both categorical and numerical features. For categorical features, the tree performs a categorical split, while for numerical features, it performs a binary split.Decision Trees can be used as building blocks for ensemble methods like Random Forests and Gradient Boosting, which combine multiple trees to improve overall predictive performance.

Decision Trees find applications in various domains, including healthcare (diagnosis), finance (credit scoring), and natural language processing. Their versatility and interpretability make them suitable for a wide range of tasks.Decision Trees can be sensitive to noisy data and outliers. They may also create overly complex trees that overfit the training dataIn summary, Decision Trees provide a simple yet powerful framework for decision-making in machine learning. Their hierarchical structure and interpretability make them valuable tools for understanding and solving a variety of problems.The choice of machine learning algorithm depends on factors such as the nature of the data, the type of DDoS attacks expected, and the available computational resources. Hybrid approaches that combine multiple algorithms or techniques are also gaining popularity to enhance the robustness and adaptability of DDoS detection systems. Additionally, ongoing research continues to explore new and improved machine learning approaches to stay ahead of the evolving landscape of DDoS threats.




## CONCLUSION

In conclusion, this research has addressed the critical issue of DDoS attacks in cloud environments, proposing and evaluating machine learning-based detection algorithms. The experiments conducted on real-world datasets have demonstrated the effectiveness of the proposed models in accurately identifying and mitigating DDoS attacks. The utilization of machine learning techniques, such as [specific algorithms used], has shown promising results in enhancing the robustness of DDoS detection systems. Our findings underscore the importance of leveraging the capabilities of machine learning for proactive DDoS defence in cloud environments. The ability of these algorithms to adapt and learn from evolving attack patterns provides a scalable and efficient solution for detecting and responding to DDoS threats. Furthermore, the integration of these models into cloud security frameworks contributes to the overall resilience of cloud infrastructures.In conclusion, the outcomes of this study not only contribute to the academic discourse on cloud security but also offer tangible insights for cybersecurity practitioners and cloud service providers seeking effective measures against DDoS attacks. As the landscape of cyber threats evolves, the integration of advanced machine learning algorithms remains a pivotal component in the arsenal of tools to safeguard cloud infrastructures against DDoS attacks.

## REFERENCE

[1] Gaurav Somani, Manoj Singh Gaur, Dheeraj Sanghi, Mauro Conti and Rajkumar Buyya, "DDoS Attacks in Cloud Computing: Issues Taxonomy and Future Directions", *Computer Communications*, vol. 107, pp. 30-48, March 2017.

[2] Gaurav Somani, Manoj Singh Gaur, Dheeraj Sanghi and Mauro Conti, "DDoS attacks in cloud computing: Collateral damage to non-targets", *Computer Networks*, vol. 109, pp. 157-171, November 2016.

[3] Sherwin Kati, Abhishek Ove, Bhavana Gotipamul, Mayur Kodche and Swati Jaiswal, "Comprehensive Overview of DDOS Attack in Cloud Computing Environment using different Machine Learning Techniques", *SSRN*, vol. 16, pp. 1-14, May 2022.

[4] Rashmi V. Deshmukha and Kailas K. Devadkar, "Understanding DDoS Attack & its Effect in Cloud Environment", *Procedia Computer Science*, vol. 49, pp. 202-210, June 2015.

[5] Jiangtao Pei, Yunli Chen and Wei Ji, "A DDoS Attack Detection Method Based on Machine Learning", *Journal of Physics: Conference Series*, vol. 1237, pp. 32-40, July 2019.

[6] Z. Masetic, K. Hajdarevic and N. Dogru, "Cloud Computing Threats Classification Model Based on the Detection Feasibility of Machine Learning Algorithms", *2017 40th International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO)*, pp. 1314-1318, May 2017.

[7] Zhuo Chen, Fu Jiang, Yijun Cheng, Xin Gu, Weirong Liu and Jun Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud", *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 251-256, May 2018.

[8] Sirisha Potluri, Monika Mangla, Suneeta Satpathy and Sachi Nandan Mohanty, "Detection and Prevention Mechanisms for DDoS Attack in Cloud Computing Environment", *2020 11th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6, October 2020.

[9] Aroosh Amjad, Tahir Alyas, Umer Farooq and Muhammad Arslan Tariq, "Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm", *ICST Transactions on Scalable Information Systems*, vol. 6, pp. 159-834, August 2019.

[10] Matthias Gander, Basel Katt, Michael Felderer, Adrian Tolbaru, Ruth Breu and Alessandro Moschitti, "Anomaly Detection in the Cloud: Detecting Security Incidents via Machine Learning", *Communications in Computer and Information Science*, vol. 379, pp. 103-116, January 2013.

[11] Gopal Singh Kushwah and Virender Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing", *Journal of Information Security and Applications*, vol. 53, pp. 102-532, August 2020.

[12] Gopal Singh KUSHWAH and Virender RANGA, *Distributed denial of service attack detection in cloud computing using hybrid extreme learning machine*, vol. 29, pp. 1852-1870, March 2021.

[13] Zecheng He, Tianwei Zhang and Ruby B. Lee, "Machine Learning Based DDoS Attack Detection from Source Side in Cloud", *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing*, pp. 114-120, July 2017.

[14] Marwane Zekri, Said El Kafhali Noureddine Aboutabit and Youssef Saadi, "DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments", *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pp. 1-7, October 2017.