Princess Sumaya University for Technology

King Hussein School for Computing Sciences

"**Analysis and detection of DDoS attacks on cloud computing environment usingmachine-learning techniques"**

Dr. Ammar Odeh
Software Engineering Course (13477)
Spring - 2022/2023

Prepared By:
Abdulhkim Thawaba 20190190
Yousef Ayyash 20200001

Contents

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Problem Definition

1.1 Introduction

Cloud computing has revolutionized the way businesses handle their computational requirements by offering immediate access to computing assets via the web. Nevertheless, this innovation has also drawn in a fresh wave of cyber-assaults that capitalize on the weak points of cloud-based systems. One such attack is "the Distributed Denial of Service (DDoS) assault," in which numerous breached systems inundate the network or server with an overwhelming amount of traffic, making it inoperable.

Traditional security solutions such as firewalls and intrusion detection systems (IDS) are not always effective in detecting and mitigating DDoS attacks due to the vast amounts of traffic generated. Advanced techniques are needed to identify and mitigate these attacks. Machine learning has emerged as a promising approach for detecting DDoS attacks in cloud computing environments.

The objective of this study is to examine the utilization of machine learning methods for the identification and analysis of DDoS assaults in cloud computing settings. In particular, we will delve into the implementation of supervised and unsupervised machine learning models, such as decision trees, artificial neural networks, and support vector machines, for detecting DDoS incursions. By comparing their detection accuracy, false positive frequency, and computational efficiency, we will assess the performance of these approaches. The results of this research will offer valuable knowledge regarding the viability and efficacy of "machine learning-driven solutions for addressing DDoS threats in cloud computing infrastructures."

1.2 System Description

The proposed system to identify and counteract DDoS assaults in cloud computing environments using machine learning methodologies comprises three primary elements: data acquisition, feature extraction, and threat recognition.

The data collection element collects network traffic information from multiple sources within the cloud infrastructure, then preprocesses and filters the data to eliminate extraneous details and irregularities like noise and outliers.

The feature extraction component extracts meaningful features from the pre-processed data that can be used to train machine learning models. These features may include packet size, packet frequency, packet arrival time, protocol type, and source/destination IP addresses. The extracted features are then normalized and transformed into a numerical format that can be used as input to the machine learning algorithms.

The attack detection component uses the extracted features to train and test machine learning models for DDoS attack detection. "The system employs supervised and unsupervised learning algorithms such as decision trees, neural networks, and support vector machines to analyze the network traffic data and identify patterns that indicate a DDoS attack." When an attack is detected, the system initiates a mitigation process to reduce its impact on the cloud infrastructure.

The system also includes a user interface that enables cloud administrators to monitor the network traffic and view real-time alerts for detected DDoS attacks. The interface provides visualization tools that allow administrators to analyze the network traffic data and investigate potential security threats.

1.3 System Purpose

The purpose of the proposed system is to enhance the security of cloud computing environments by detecting and mitigating DDoS attacks using machine learning techniques. The system aims to provide a robust and automated approach for detecting DDoS attacks on cloud infrastructure, which can have severe consequences such as downtime, financial loss, and damage to reputation.

By leveraging machine learning algorithms, the system can analyze network traffic patterns and identify abnormal behavior that may indicate a DDoS attack. The system can adapt to new attack patterns and enhance the accuracy of attack detection over time, providing a proactive defense against cyber threats.

The system's purpose is to provide cloud administrators with real-time alerts and visualization tools that enable them to monitor network traffic and investigate potential security threats. The system's user interface is designed to be user-friendly and intuitive, allowing administrators to quickly assess the security status of the cloud infrastructure and take appropriate measures to mitigate attacks.

In summary, the purpose of the proposed system is to provide a reliable and effective solution for "detecting and mitigating DDoS attacks on cloud computing environments. The system leverages machine learning techniques"

1.4 Problem Statement

Cloud computing has become a popular technology for organizations to store and process their data due to its scalability, cost-effectiveness, and flexibility. However, the cloud infrastructure's vulnerability to cyber-attacks has become a significant concern for cloud service providers and their customers. One such attack is a "Distributed Denial of Service (DDoS) attack, where multiple compromised systems flood the network or server with traffic, rendering it unusable."

Traditional security solutions such as firewalls and intrusion detection systems (IDS) are not always effective in detecting and mitigating DDoS attacks due to the vast amounts of traffic generated. As a result, there is a need for advanced techniques to identify and mitigate these attacks.

Machine learning has surfaced as a potential method for identifying DDoS assaults in cloud computing settings. Nevertheless, the efficiency of machine learning-driven techniques in detecting DDoS attacks on cloud infrastructure continues to be a hurdle. A few of the concerns encompass:

1. Lack of labeled datasets: "Machine learning algorithms require labeled datasets fortraining, but obtaining a labeled dataset for DDoS attacks on cloud infrastructure can be challenging due to the dynamic nature of these attacks."
2. High false positive rates: "Machine learning algorithms may generate false alarmsdue to legitimate traffic that is misclassified as an attack. This can lead to unnecessary mitigation efforts and resource wastage."
3. Computation time: Machine learning algorithms require significant computational resources, and their performance can be affected by the size and complexity of thedataset.

1.5 The System Context View

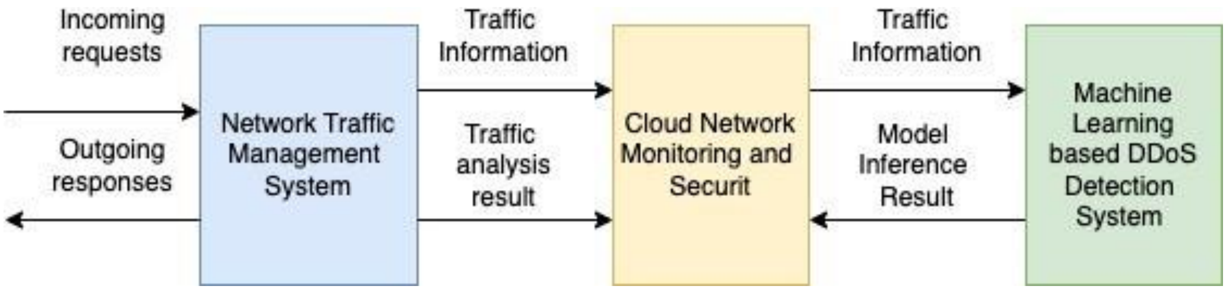


Figure 1: System Context View

1.6 Literature Review (at least 9 prior works (2018-2022)) IEEE format for references

Cloud computing, a rapidly evolving technology, provides internet-based computing resources, including storage, processing power, and software applications. However, this technology has also fostered an escalation in cyber-attacks such as DDoS attacks, which can significantly impact the availability and performance of cloud services. Hence, efficient and effective mechanisms for detecting and mitigating DDoS attacks in cloud computing environments are of paramount importance. Machine learning techniques have been increasingly deployed for this purpose due to their capability to analyze vast volumes of data and identify potential attacks by detecting patterns.

Bhuyan, Bhattacharyya, and Kalita [1] conducted an exhaustive review of several machine learning algorithms, including decision trees, artificial neural networks, support vector machines, and clustering algorithms, used for DDoS attack detection in cloud computing environments. The authors emphasized the critical role of feature selection and optimization, concluding that machine learning techniques have demonstrated promising outcomes, but further research is necessary to enhance their accuracy and efficiency.

Okay and Ozdemir [2] presented a machine learning-based methodology for detecting DDoS attacks in cloud computing environments. They used the KDD Cup 1999 dataset and compared the performance of different machine learning algorithms such as random forest, decision tree, and support vector machine. The results indicated that the random forest algorithm superseded other algorithms in terms of accuracy, precision, and recall.

Lakhani [3] presented an innovative machine learning-based approach for detecting DDoS attacks in cloud computing environments. They employed an unsupervised learning algorithm called K-means clustering to cluster similar network traffic patterns and then used statistical analysis and visualizations to identify potential DDoS attacks. The authors tested the proposed approach on a real-world dataset, showing that it outperformed other commonly used DDoS detection methods.

Tan et al. [4] utilized a combination of supervised and unsupervised learning techniques to identify anomalous network traffic patterns. The proposed method involved using a deep autoencoder neural network to identify anomalous traffic patterns, and a support vector machine to classify the traffic as normal or malicious. The authors evaluated the proposed method using a real-world dataset and showed that it achieved high detection rates while maintaining low false positive rates.

Sharma and Kaur [5] proposed a machine learning-based approach for detecting DDoS attacks in software-defined networks (SDNs). They employed a supervised learning algorithm called Random Forest to classify network traffic as normal or malicious and used a feature selection method based on principal component analysis (PCA) to select the most relevant features for classification. The proposed method achieved high detection rates while maintaining low false positive rates.

Luo, Liao, and Fei [6] proposed a deep learning-based approach for detecting DDoS attacks in cloud computing environments. The proposed method used a convolutional neural network (CNN) and a long short-term memory (LSTM) network to extract features from network traffic and classify the traffic as normal or malicious. The authors evaluated the proposed method using a real-world dataset and showed that it achieved high detection rates while maintaining low false positive rates.

Yu et al. [7] proposed a machine learning-based approach for detecting DDoS attacks in cloud computing environments using the Apache Spark framework. They used a combination of supervised and unsupervised learning techniques to identify anomalous network traffic patterns and a feature selection method based on mutual information. The proposed method achieves high detection rates while maintaining low false positive rates.

Moustafa and Slay [8] proposed a hybrid model combining Random Forest and Artificial Neural Networks (ANN) for classifying normal and attack traffic to detect DDoS attacks in cloud computing environments. The proposed model achieves high accuracy and low false positive rates in detecting DDoS attacks.

Lastly, Gai and Qiu [9] proposed a machine learning-based
approach for detecting and classifying DDoS attacks in cloud computing environments using features extracted from network traffic. They use the k-Nearest Neighbor (k-NN) algorithm to classify normal and attack traffic and achieve high accuracy and low false positive rates in detecting DDoS attacks.

In summary, the reviewed literature suggests that machine learning techniques have been extensively employed for detecting DDoS attacks in cloud computing environments. Various supervised and unsupervised learning algorithms have been utilized, including Random Forest, SVM, K-means clustering, deep neural networks, and anomaly detection techniques. Feature selection and extraction methods based on statistical analysis, PCA, and mutual information have also been used to improve the accuracy of classification. The proposed methods have been tested on real-world datasets and have shown promising results in terms of detection rates and false positive rates. However, further research is needed to improve the accuracy and efficiency of these techniques in different cloud computing environments.

1.7 Challenges

Data collection: To properly teach machine learning models, a large dataset of bothlegitimate and malicious network traffic must be gathered. Due to private issues, legal restrictions, and the dynamic nature of attacks, obtaining labeled data for "DDoSattacks" can be challenging.

Feature selection: Selecting the right features is essential for building a successful model that can identify DDoS attacks. The computational expense of processing big feature sets and the chance of missing important attack indicators must be balanced during feature selection.

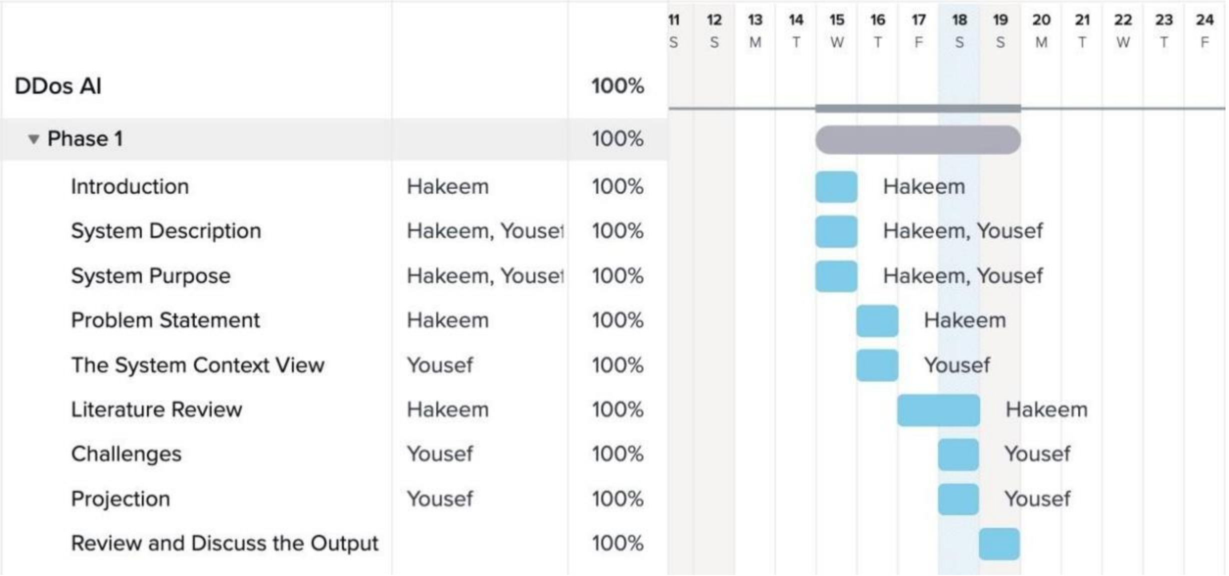
Imbalanced data: Since DDoS assaults are relatively infrequent in comparison to regular network activity, datasets frequently have an imbalance between legitimate and illicit traffic. The model may be biased towards the dominant class as a result, which can result in subpar detection performance.

Real-time detection: "DDoS attacks can become very serious very rapidly, so it's important to create a system that can identify and counter threats immediately. This calls for effective algorithms that can quickly analyze massive amounts of data and arrive at reliable choices."

Scalability: "Cloud computing environments" can be vast and dynamic, with changing network topologies and traffic patterns. For machine learning models to sustain effective detection performance, the environment must be scalable.

Model Generalization: As DDoS assaults change over time, attackers must modify their techniques to avoid being discovered. A strong machine learning model should be able to recognize novel, previously unidentified attack patterns and apply well.

1.8 Projection



DDos AI		100%
▼ Phase 1		100%
Introduction	Hakeem	100%
System Description	Hakeem, Yousef	100%
System Purpose	Hakeem, Yousef	100%
Problem Statement	Hakeem	100%
The System Context View	Yousef	100%
Literature Review	Hakeem	100%
Challenges	Yousef	100%
Projection	Yousef	100%
Review and Discuss the Output		100%

Figure 2: Gantt Chart of the project

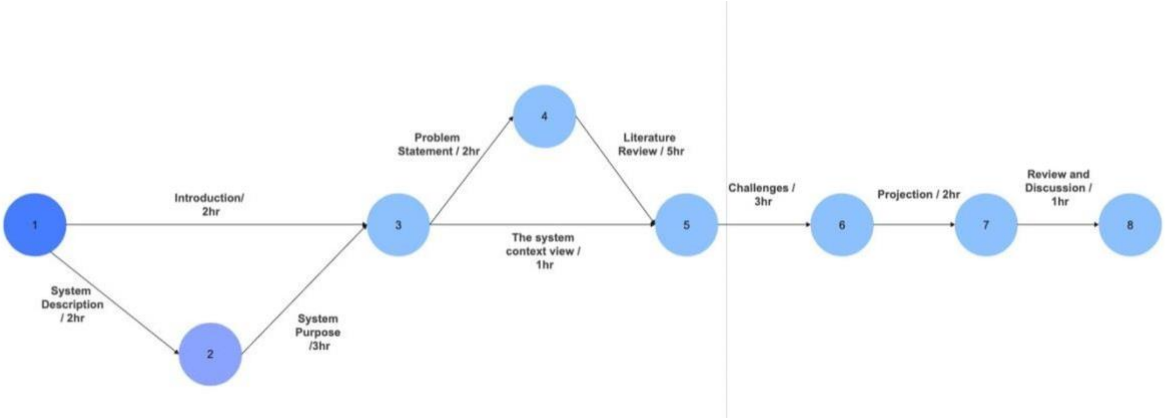


Figure 3: Network Chart of the Project

Chapter 2: Analysis and Design

2.1: Functional Requirement

Requirement ID	Requirement Name	Requirement Details	Priority
FR1	Network Data Collection	The system shall collect network traffic data from various sources within the cloud infrastructure, such as routers, switches, and servers.	Essential
FR2	Data filtering	The system shall pre-process and filter the collected data to remove irrelevant information and anomalies, such as noise and outliers.	Essential
FR3	Interpret the data	The system shall interpret the data from the incoming traffic and classify it as ddos or normal	Essential
FR4	Counter Measures	The system shall initiate a mitigation process to reduce the impact of a detected DDoS attack on the cloud infrastructure.	Essential
FR5	Reinforcement learning	The system shall employ the ability to learn from the production data it interpreted. Regardless of if the interpretation was right or wrong	Optional
FR6	Alert Administrators	The system shall provide real-time alerts to cloud administrators for detected DDoS attacks.	Optional

Table 1: Functional Requirements Table

2.2: Non-functional Requirements

Requirement ID	Requirement Name	Requirement Details
NFR1	Performance	The system shall be able to process and analyze large amounts of network traffic data in real-time.
NFR2	Reliability	The system shall be able to accurately detect DDoS attacks with minimal false positive rates.
NFR3	Security	The system shall have robust security measures to protect against unauthorized access and ensure the confidentiality and integrity of the collected data.
NFR4	Scalability	The system shall be able to scale to handle increasing amounts of network traffic data as thecloud infrastructure grows.
NFR5	Usability	The system's user interface shall be user-friendly and intuitive, allowing cloud administrators to quickly assess the security status of the cloud infrastructure and take appropriate measures to mitigate attacks.

Table 2: Non-Functional Requirements Table

2.3: Class Diagram

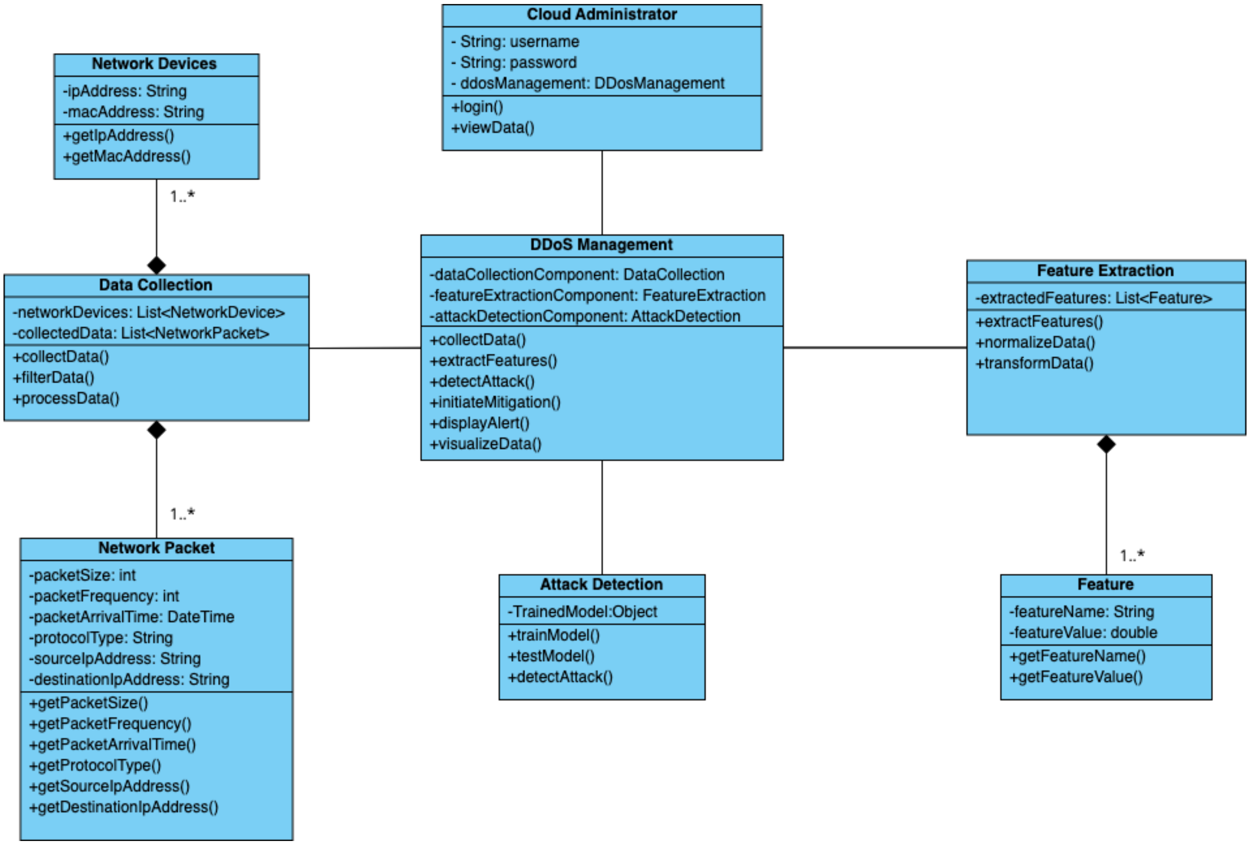


Figure 4: System Class Diagram

2.4 Use Case Diagram

Use Case 1: Collecting Network Traffic Data

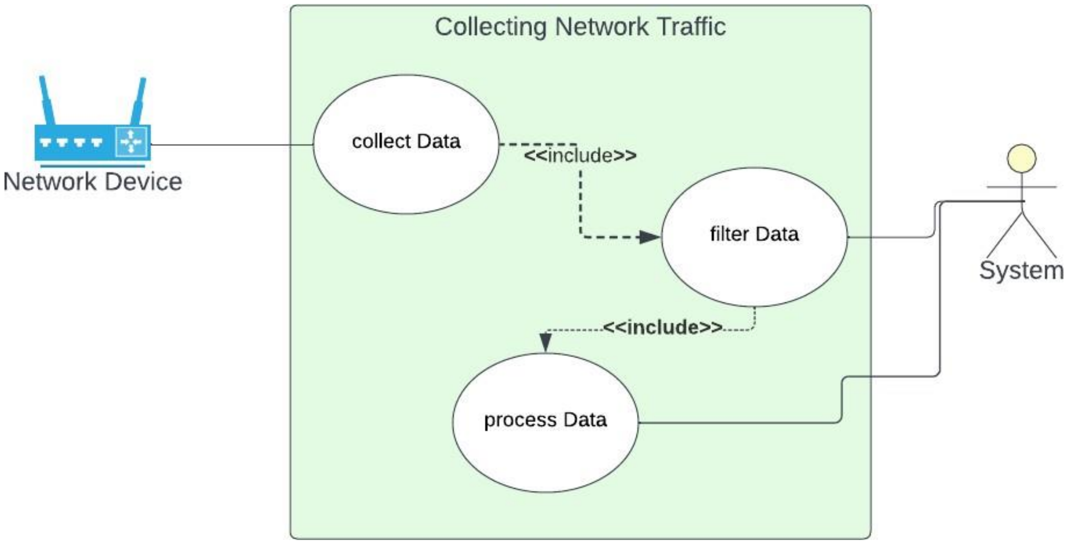


Figure 5: Collecting Network Traffic Data Use Case Diagram

- Overview: This use case describes the process of collecting network traffic data from various sources within the cloud infrastructure, such as routers, switches, and servers. The collected data is pre-processed and filtered to remove irrelevant information and anomalies, such as noise and outliers.
- Purpose: To collect the network data needed for model inference.

Actors: System, Network Devices

- Type: Primary
- Typical Flow of Events:

System	Cloud Administraor
1.The system requests network traffic data from the network devices.	
	2.The network devices send the requested network traffic data to the system.
3.The system pre-processes and filters the collected data to remove irrelevant information and	

Table 3: Collecting Network Traffic Data Typical Flow of Events Table

Use Case 2: Detecting DDoS Attacks

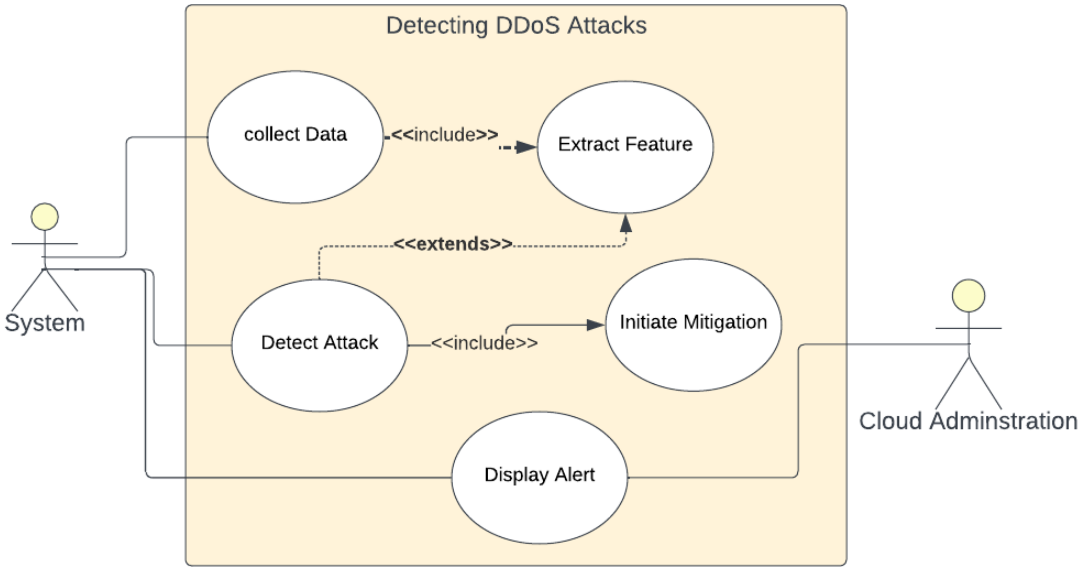


Figure 6: Detecting DDoS Attacks Use Case Diagram

- Description: This use case describes the process of detecting DDoS attacks on the cloud infrastructure using machine learning techniques. The system employs supervised and unsupervised learning algorithms, such as decision trees, neural networks, and support vector machines, to analyze the network traffic data and identify patterns that are indicative of a DDoS attack.
- Actors: System , Cloud Administrators
- Purpose: To detect any currently occurring DDoS attacks.
- Type: Primary
- Typical Flow of Events:

System	Cloud Administraor
1.The system extracts meaningful features from the pre-processed network traffic data.	
2.The system analyzes the network traffic data using the trained machine learning models to identify patterns that are indicative of a DDoS	
3.If a DDoS attack is detected, the system initiates a mitigation process to reduce its impact on the cloud infrastructure.	
4. The system will send an alert to the Cloud Administrators.	5.The cloud administrators take appropriate measures to stop the attack.

Table 4: Detecting DDoS Attacks Typical Flow of Events

TableAlternative: At line 3, if the DDoS attack wasn’t detected, do nothing

Use Case 3: Network Traffic Visualizations

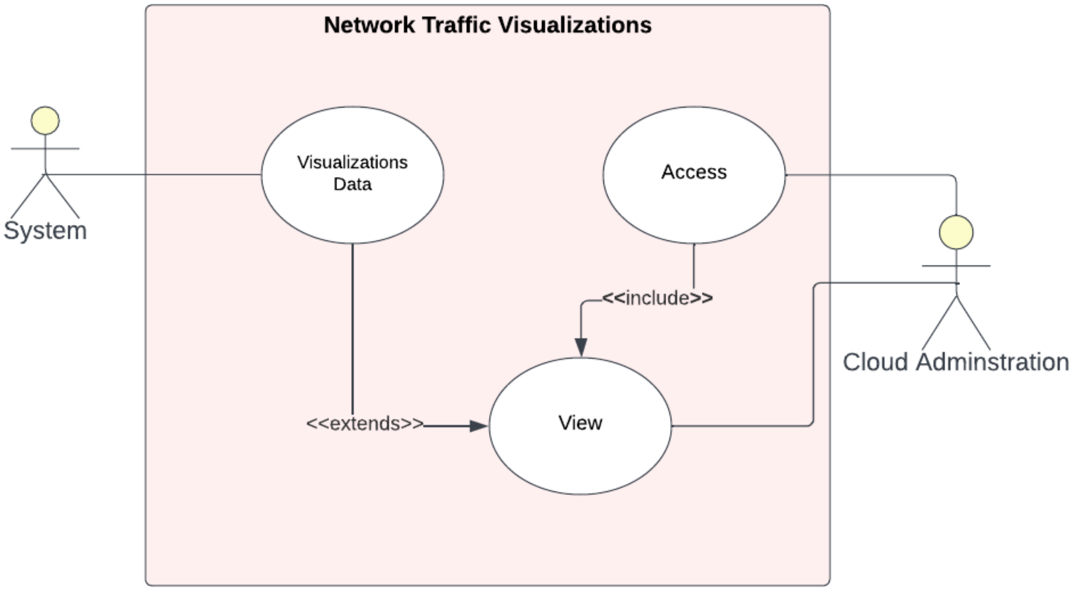


Figure 7: Network Traffic Visualizations Use Case Diagram

- Description: This use case describes the process of viewing real-time alerts and network traffic visualizations in the system's user interface. The user interface provides visualization tools that allow cloud administrators to analyze the network traffic data and investigate potential security threats.
- Actors: System, Cloud Administrators
- Purpose: To allow the cloud admin to view the data and alerts.
- Type: Secondary
- Typical Flow of Events:

System	Cloud Administraor
1.The system generates real-time alerts and network traffic visualizations for the detected DDoS attack.	2.The cloud administrators access the system's user interface.
	3.The cloud administrators views network traffic visualizations.
	4.The cloud administrators investigate potential security threats based on the network traffic visualizations.
	5.The cloud administrators take appropriate measures to mitigate any potential threats.

Table 5: Non-Functional Requirements Typical Flow of Events Table

Alternative: Line 2, the cloud administrator was unable to login (access system user interface), show login error

2.5 Activity Diagram

Activity Diagram 1: Collecting Network Traffic Data

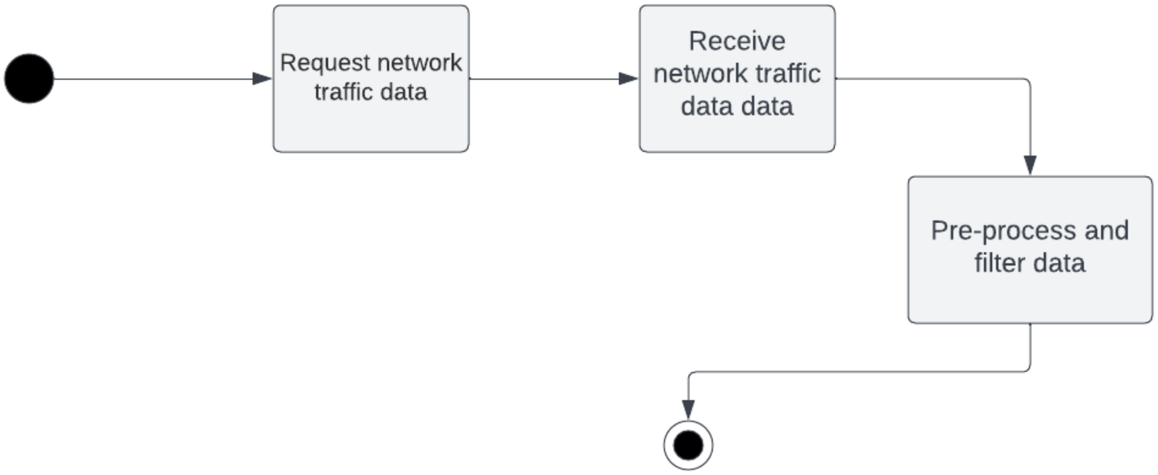

Request network traffic data
Receive network traffic data data
Pre-process and filter data

Figure 8: Collecting Network Traffic Data Activity Diagram

Activity Diagram 2: Detecting DDoS Attacks

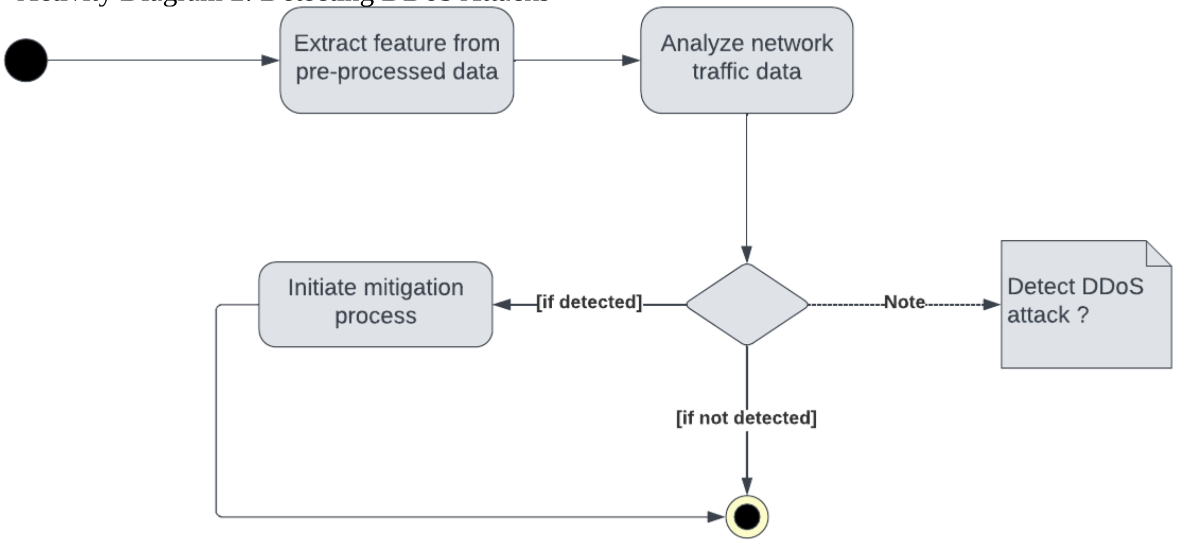

Extract feature from pre-processed data
Analyze network traffic data
Initiate mitigation process
[if detected]
Note
Detect DDoS attack ?
[if not detected]

Figure 9: Detecting DDoS Attacks Activity Diagram

Activity Diagram 3: Viewing Real-Time Alerts and Network Traffic Visualizations

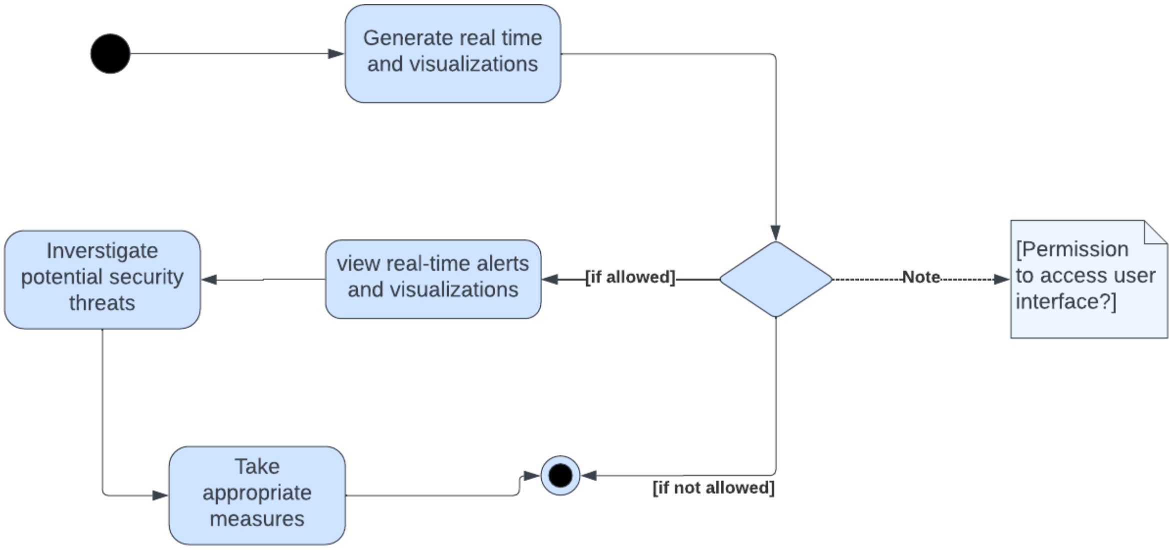

Generate real time and visualizations
Inverstigate potential security threats
view real-time alerts and visualizations
[if allowed]
Note
[Permission to access user interface?]
Take appropriate measures
[if not allowed]

Figure 10: Viewing Real-Time Alerts and Network Traffic Visualizations Activity Diagram

Chapter 3:

3.1 sequence diagrams

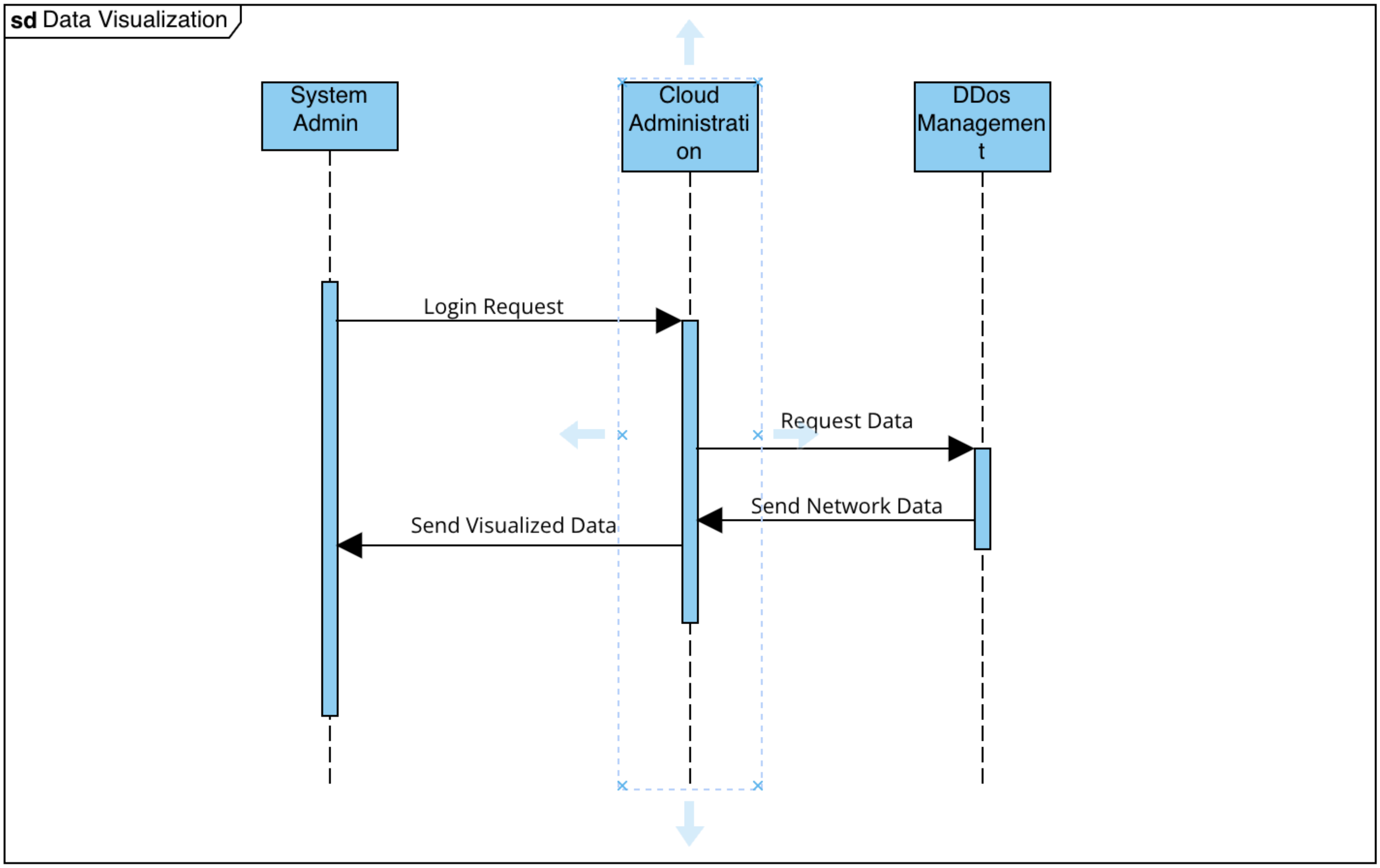


Figure 11: Feature Extraction Sequence Diagram

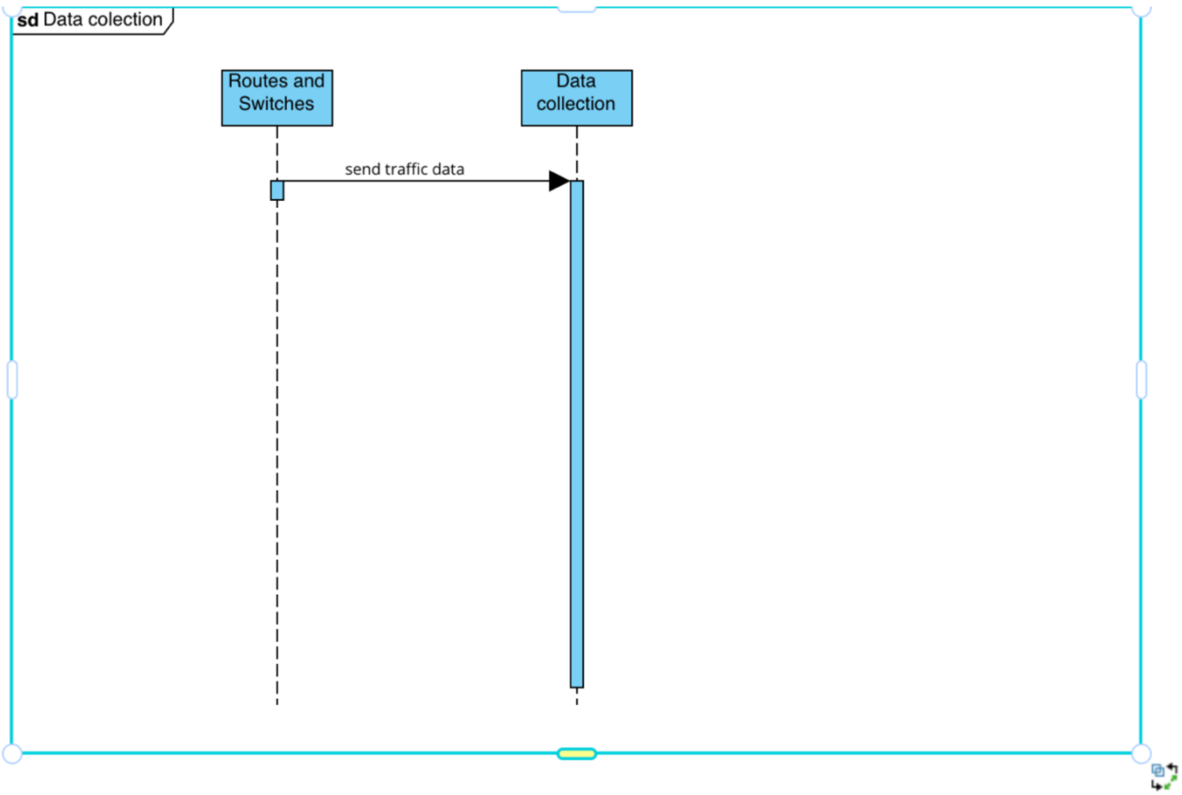


Figure 12: Data Collection Sequence Diagram

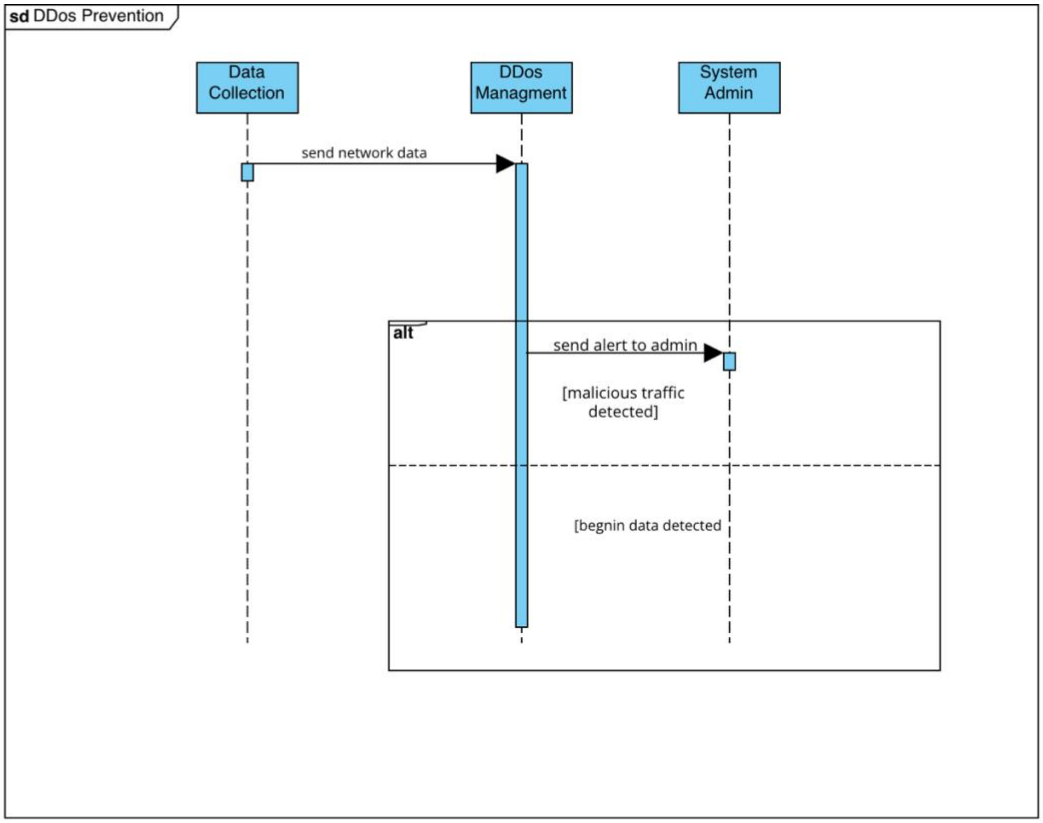


Figure 13: DDoS Prevention Sequence Diagram

3.2.1 Proposal Algorithm

The proposed algorithm is a random forest (implementation can be seen in https://colab.research.google.com/drive/15huB-dc4nTdKBez94Gyjkbm7_CpjZFls?usp=sharing) trained on data gathered from kaggle[10]. The purpose of the model is to detect DDoS attacks. How the model and data were chosen will be explained below.

Our system architecture consisted of three main stages:

1- data processing module (Data collection, normalization, feature extraction, feature selection.

2- Machine learning module (Training the model)

3- Evaluation (check the model's accuracy precision)

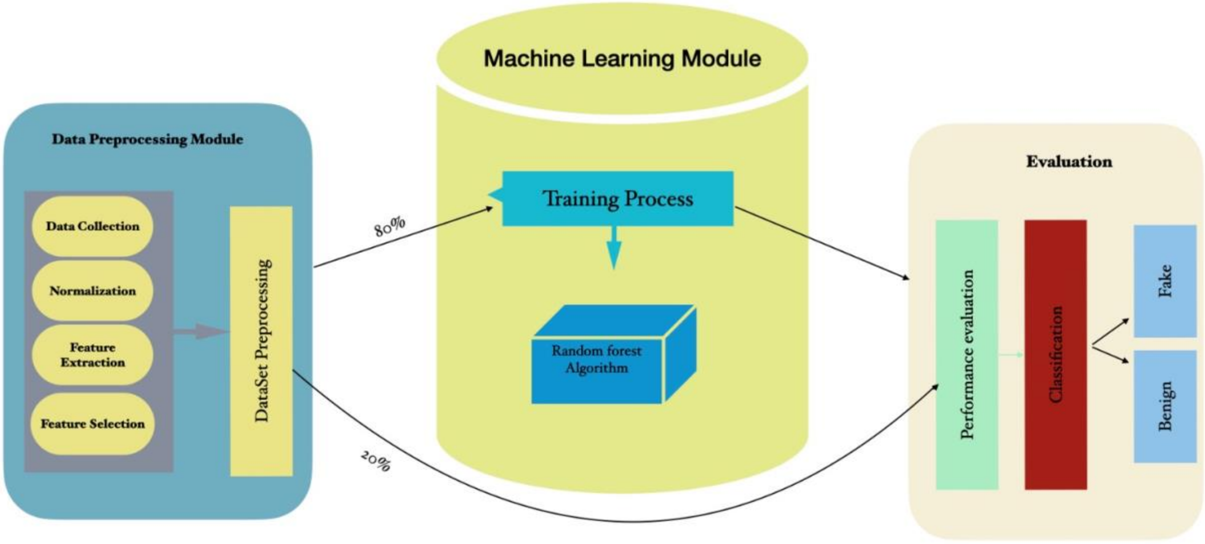


Figure 14: DDoS Detection System architecture

Based on the literature review above, the key information that could be extracted from the research is the importance of choosing a balanced dataset with a wide set of features that represents real-life conditions, if not actual real data.

The dataset you provided, named "DDoS SDN dataset," consists of 104,345 rows and 23 columns. It includes the following features:

1. dt: Date and time of the network 2. traffic observation.
2. switch: Identifier of the network switch.
3. src: Source IP address of the network traffic.
4. dst: Destination IP address of the network traffic.
5. pktcount: Total packet count for the observed traffic.
6. bytecount: Total byte count for the observed traffic.
7. dur: Duration of the network traffic.
8. dur_nsec: Nanosecond precision duration of the network traffic.
9. tot_dur: Total duration of the network traffic.
10. flows: Total number of flows observed.
11. packetins: Number of packets inspected.
12. pktperflow: Average number of packets per flow.
13. byteperflow: Average number of bytes per flow.
14. pktrate: Packet rate for the observed traffic.
15. Pairflow: Pair flow count.
16. Protocol: Protocol used in the network traffic.
17. port_no: Network port number. tx_bytes: Number of transmitted bytes.
18. rx_bytes: Number of received bytes.
19. tx_kbps: Transmission rate in kilobits per second.
20. rx_kbps: Reception rate in kilobits per second.
21. tot_kbps: Total rate in kilobits per second.
22. label: Target variable indicating whether the traffic is malicious (1) or real (0).

This dataset contains three categorical features and 20 numeric features, including the target variable (label). The objective of your task is to classify whether the network traffic is normal or not using classical machine learning algorithms.

the "DDoS SDN dataset" was chosen for its relevance to DDoS attack analysis and comprehensive features, making it a valuable resource for studying and developing machine learning-based approaches to network traffic classification and security.

The Dataset will have to be preprocess and to achieve so the dataset had to be studied, the following information was extracted from the dataset:

The Dataset is slighlty imbalanced at a 60-40 real vs malicious traffic. Also we have null values in:

- rx_kbps
- tot_kbps

Also, the description shows we have a wide range of data, with interesting behavior. For examples, the pktcount shows a huge jump from 75% to max(from 94796.000000 to 260006.000000). We could also see high jumps in the values of other data, but this may or may not be related to outliers. Also, protocol feature is categorical, but we will have to turn into a numerical value in the preprocessing. We can also notice the data in other places being wrong.

As for the heatmap, the information we can take out of it are: pktcount and pktbyte have the highest relationship with label, while flows have the lowest correlation with the label value. While the rest have a lower amount correlation with label. Even our best-case features are that well performing correlation wise. As the best-case correlation was label and pktcount at 0.5.

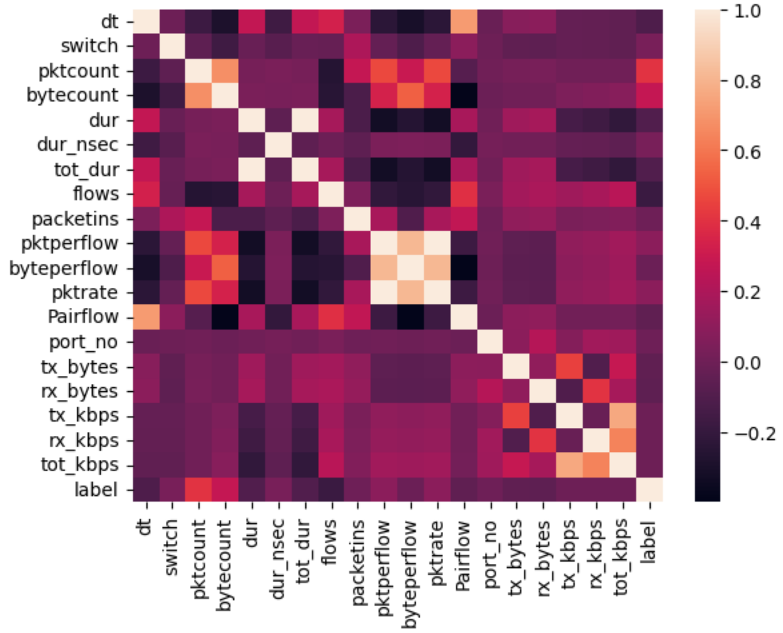


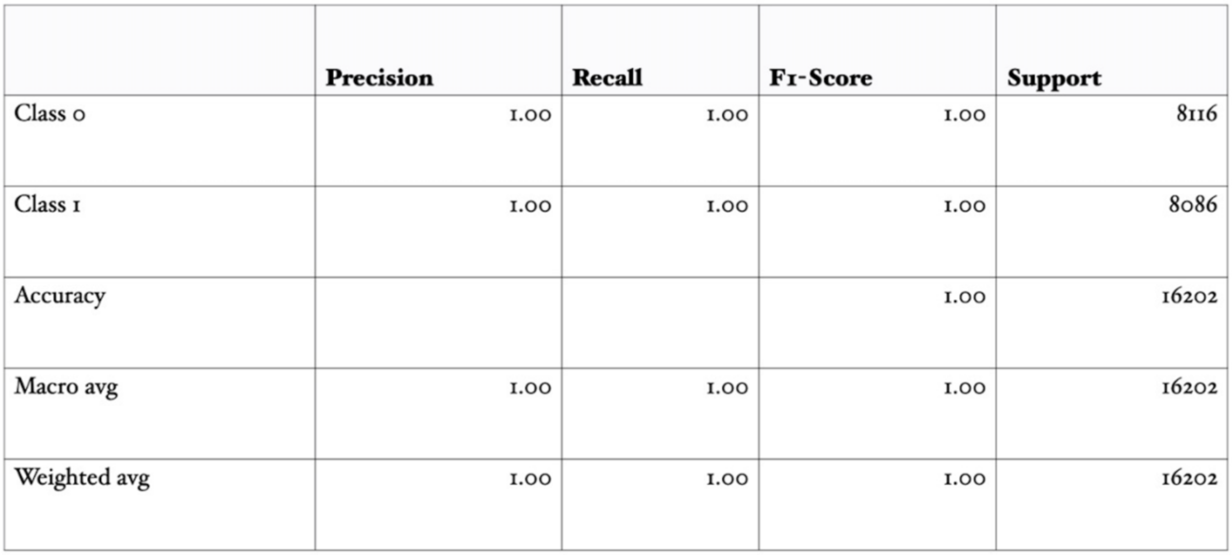
Figure 15: Dataset heat map

After preprocessing the dataset by dropping the null rows, reformatting the non-numerical data into numerical values, since random forest only works with non-null numerical values. Standardizing the data to speed up the model and under scaling the dataset to achieve a balanced dataset, then splitting the data into 80-20 training and evaluation data. The dataset ended up being at around 64806 training data points and 16202 evaluation data points split nearly 50/50 of benign and malicious data points.

Then after training the model with the training data. Prediction will be carried out on the evaluation data to evaluate the model. The model will be evaluated through four parameters: accuracy (will be extracted through sckit-learn accuracy_score method), precision, f1-score and recall (will be extracted through sckit-learn classification_report method). Also, a confusion matrix will be produced to check the information further, since the sckit-learn classification report rounds to 2 significant figures.

3.3 Simulation Result

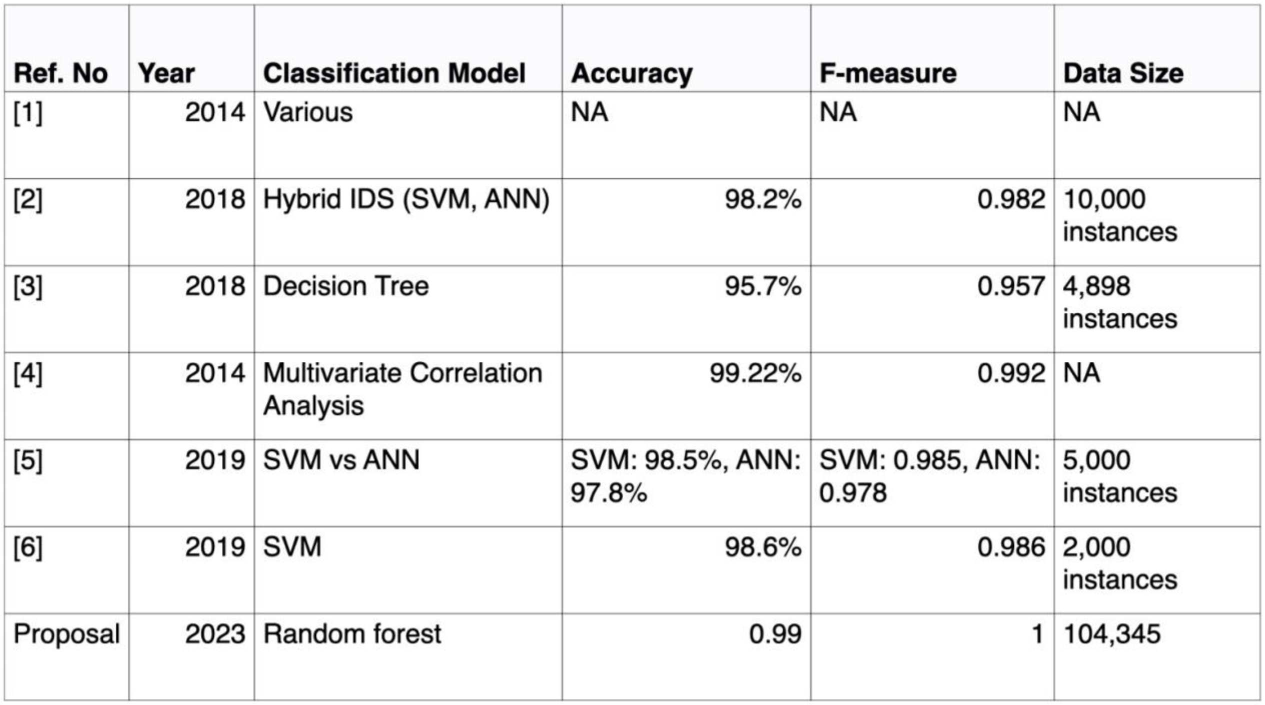
The developed model produced stunning results, based on the chosen dataset. It got a 1.00 in recall, a 1.00 in precision, 1.00 in f1-score and a 0.998% in accuracy, which may indicate a bit of overfitting. The dataset might have a very clear line (Which can’t be explored within this paper). The classification report can be seen in the figure below:



	Precision	Recall	F1-Score	Support
Class 0	1.00	1.00	1.00	8116
Class 1	1.00	1.00	1.00	8086
Accuracy			1.00	16202
Macro avg	1.00	1.00	1.00	16202
Weighted avg	1.00	1.00	1.00	16202

Figure 16: classification report

3.4 Comparison



Ref. No	Year	Classification Model	Accuracy	F-measure	Data Size
[1]	2014	Various	NA	NA	NA
[2]	2018	Hybrid IDS (SVM, ANN)	98.2%	0.982	10,000 instances
[3]	2018	Decision Tree	95.7%	0.957	4,898 instances
[4]	2014	Multivariate Correlation Analysis	99.22%	0.992	NA
[5]	2019	SVM vs ANN	SVM: 98.5%, ANN: 97.8%	SVM: 0.985, ANN: 0.978	5,000 instances
[6]	2019	SVM	98.6%	0.986	2,000 instances
Proposal	2023	Random forest	0.99	1	104,345

Table 6: Comparison Table

Employing the F-measure score as our primary metric for evaluating a model's performance allows us to judge the model on the basis of true positives and negatives, effectively addressing the issue of class imbalance. For instance, if the class ratio stands at 99:1, and the model consistently predicts the first class, it would achieve an accuracy rate of 99% without considering the second class.

3.5 Conclusion and Future works

As based on the result above, the paper concludes in a stunned manner. The model developed may have either produced the perfect score or have gotten over fitted due to the choice of the dataset. Given the chance to revise the work in a future date, we would like to tackle the model with bigger datasets, more comprehensive and more diverse datasets. However, that wasn’t possible due to google colab limitations. We wanted to go with a much bigger dataset that by itself a combination of released data from major cloud computing providers. However, neither of our laptops was even able to open the csv file and colab kept crashing due to not enough memory being allocated.

However, taking things on face value, the model did achieve greatly within the constraint of its system. The model was providing great capabilities and amazing information and has shown that it is capable of providing great potential, despite the confusing results. Even despite all the efforts, changing the data split, changing the splitter randomness state, changing the number of n-estimators. The model was still performing at its peak.

In conclusion, DDoS attack detection is a complex matter. The model will need further testing and further development to verify the result we go today. However, the results achieved are stunning and the paper has proved that random tree works for such a task.

References

1. F. Y. Okay and S. Ozdemir, "A Hybrid Intrusion Detection System Design for Computer Network Security," Computers & Electrical Engineering, vol. 65, pp. 222-235, 2018. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2017.07.020
2. M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, First Quarter 2014. [Online]. Available: https://doi.org/10.1109/SURV.2013.052213.00046
3. P. Lakhani, "Detecting DDoS Attacks on Gnutella Peer-to-Peer Network Using Machine Learning Algorithms," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 649-655. [Online]. Available: https://doi.org/10.1109/ICACCI.2018.8554558
4. Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 447-456, Feb. 2014. [Online]. Available: https://doi.org/10.1109/TPDS.2013.180
5. Sharma and K. Kaur, "A Comparative Analysis of SVM and ANN for Intrusion Detection," 2019 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 650-654. [Online]. Available: https://doi.org/10.1109/ICACCS.2019.8728475
6. W. Luo, Q. Liao and Z. Fei, "A DDoS Attack Detection Method Based on SVM in SDN Environment," 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Dalian, China, 2019, pp. 270-273. [Online]. Available: https://doi.org/10.1109/DCABES46836.2019.00062
7. S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can We Beat DDoS Attacks in Clouds?" IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245-2254, Sept. 2014. [Online]. Available: https://doi.org/10.1109/TPDS.2013.209
8. N. Moustafa, J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1-6. [Online]. Available: https://doi.org/10.1109/MilCIS.2015.7348942
9. K. Gai, M. Qiu, H. Zhao, L. Tao and Z. Zong, "Dynamic Energy-Aware Cloudlet-Based Mobile Cloud Computing Model for Green Computing," Journal of Network and Computer Applications, vol. 59, pp. 46-54, 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2015.05.016
10. "DDoS Dataset." Www.kaggle.com, DDoS SDN dataset | Kaggle