



G.H. RAISONI COLLEGE OF ENGINEERING AND MANAGEMENT, WAGHOLI, PUNE.

Department of Artificial Intelligence



Topic:

Denial Of Service(DoS) Attacks Detection Using Machine Learning

Group ID:- AG6

Project Group:-

A40 ONKAR MANE

A51 RUTUJA OHAL

A56 SANKET CHAUDHARY

Exam Seat No:-

2020AAIT1101038

2020AAIT1101067

2020AAIT1101008

Internal Guide:- Barkha Kumari

Engineering

Management

Law

Schools

Other Courses

■ NAGPUR ■ PUNE ■ JALGAON ■ AMRAVATI ■ AHMEDNAGAR ■ CHHINDWARA



Contents

- ☛ Introduction
- ☛ Literature Survey
- ☛ Existing system and Challenges
- ☛ Motivation and Scope
- ☛ Problem Definition
- ☛ Mathematical Model
- ☛ Feasibility Analysis
- ☛ Software and Hardware requirement
- ☛ System Design
- ☛ System Feature
- ☛ Algorithms
- ☛ Dataset and Result Analysis
- ☛ Comparative analysis using graphs and tables
- ☛ Testing of Project:- Apply various testing strategy
- ☛ Constraints/Limitation
- ☛ Applications
- ☛ Result /Outcomes
- ☛ Conclusion
- ☛ References
- ☛ Paper Submission Deatils

Introduction

In today's interconnected world, cyber threats pose a significant risk to the stability and security of online services. Among these threats, **Denial of Service (DoS) attacks** stand out as one of the most prevalent and disruptive forms of cyberattacks. A DoS attack aims to overwhelm a target system or network with a flood of traffic, rendering it unavailable to legitimate users.

Traditional methods of DoS attack detection often struggle to keep up with the ever-evolving techniques employed by malicious actors. As a result, the need for more sophisticated and adaptive detection approaches arises. This is where machine learning algorithms come into play.

Machine learning, a subset of artificial intelligence, empowers systems to learn from data patterns and make predictions or decisions without explicit programming.

Literature Survey

- 1) Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-based Private Cloud. Authors:-Karan B. Virupakshar, Manjunath Asundi, Kishor Channal, Pooja Shettar, Somashekar Patil, Narayan D. G.
- 2) Cyber Attack Detection thanks to Machine Learning Algorithms. Authors:- Antoine Delplace, Sheryl Hermoso, Kristofer Anandita
- 3) Man-in-the-middle and denial of service attacks detection using Machine learning algorithms. Authors:-Sura Abdulmunem Mohammed Al-Juboori, Firas Hazzaa1, Zinah Sattar Jabbar, Sinan Salih, Hassan Muwafaq Gheni
- 4) Denial of Service Attacks: Tools and Categories. Authors:- Hadeel S. Obaid
- 5) Denial of Service Attack Classification Using Machine Learning with multi-features. Authors: Furqan Rustam, Muhammad Faheem Mushtaq, Ameer Hamza, Muhammad Shoaib Farooq, Anca Delia Jurcut Imran Ashraf

Existing system and Challenges

Existing System:

1. Traditional DoS Detection Methods: Existing systems often rely on rule-based and signature-based methods to detect denial of service (DoS) attacks. These methods are limited in their ability to adapt to evolving attack techniques.
2. Network Traffic Analysis: Existing systems typically analyze network traffic patterns to identify anomalies. This approach can be resource-intensive and may generate false positives, impacting network performance.
3. Limited Machine Learning Integration: Some existing systems incorporate machine learning techniques, but their models are often static and lack adaptability to emerging attack vectors.

Existing system and Challenges

Challenges:

1. **Evolving Attack Techniques:** Hackers are constantly developing new and sophisticated DoS attack methods, making it challenging to stay ahead with static detection systems.
2. **False Positives:** Existing systems can generate a significant number of false positives, leading to unnecessary alerts and straining network resources.
3. **Scalability:** As network traffic increases, scalability becomes a significant challenge in terms of real- time detection and response to DoS attacks.
4. **Anomaly Detection:** Machine learning models struggle with accurately identifying attack patterns, especially when dealing with complex and subtle anomalies.
5. **Training Data:** Collecting labeled training data for machine learning models in the context of DoS attacks can be difficult due to the sporadic nature of these attacks.

Motivation and Scope

Motivation:

The increasing frequency and complexity of denial of service (DoS) attacks pose a severe threat to network security. Traditional detection methods fall short in effectively countering these evolving threats. Employing machine learning in DoS attack detection offers a promising solution, as it can adapt to changing attack patterns and reduce false positives. This project is motivated by the urgent need to enhance the resilience of network infrastructures against DoS attacks, ultimately safeguarding critical data, services, and systems.

Scope:

This project aims to develop a robust DoS attack detection system using machine learning techniques. It will encompass data collection, feature engineering, model training, and real-time network traffic analysis. The scope also includes addressing challenges such as false positives, scalability, and real-time response. The project will contribute to the advancement of network security by providing a more adaptive and efficient means of detecting and mitigating DoS attacks, thereby enhancing the overall reliability and integrity of digital infrastructure.

Problem Definition

Problem Definition:

The problem addressed by this project is the inadequacy of existing denial of service (DoS) attack detection methods in effectively countering evolving and sophisticated attacks. Current systems primarily rely on rule-based and signature-based approaches, which struggle to adapt to new attack vectors. This project seeks to develop a machine learning-based DoS detection system to enhance the network's ability to identify and mitigate these attacks efficiently, reducing false positives and increasing the resilience of critical systems and services.

Mathematical Model

Mathematical Model:-

The mathematical model for the "Denial of Service Attack Detection Using Machine Learning" project involves creating a predictive framework that utilizes statistical and machine learning techniques to analyze network traffic patterns. It encompasses data preprocessing, feature extraction, and the application of supervised learning algorithms. The model aims to distinguish between normal and anomalous network behavior, effectively detecting DoS attacks. It incorporates metrics for precision, recall, and F1 score to evaluate its performance. The mathematical model will provide a quantitative basis for assessing the system's accuracy, efficiency, and effectiveness in mitigating DoS attacks in real-time network environments.

Feasibility Analysis

Feasibility Analysis:-

Feasibility Analysis for the "Denial of Service Attack Detection Using Machine Learning" project evaluates its practicality and viability. This analysis considers technical feasibility by assessing the availability of necessary resources, data, and machine learning tools. Economic feasibility examines the project's cost-effectiveness and potential return on investment. Operational feasibility assesses the adaptability of the system within the existing network infrastructure. Finally, the project's schedule feasibility ensures that it can be completed within a reasonable timeframe. This comprehensive analysis informs stakeholders about the practicality and potential success of implementing a machine learning-based DoS attack detection system.

Software and Hardware requirement

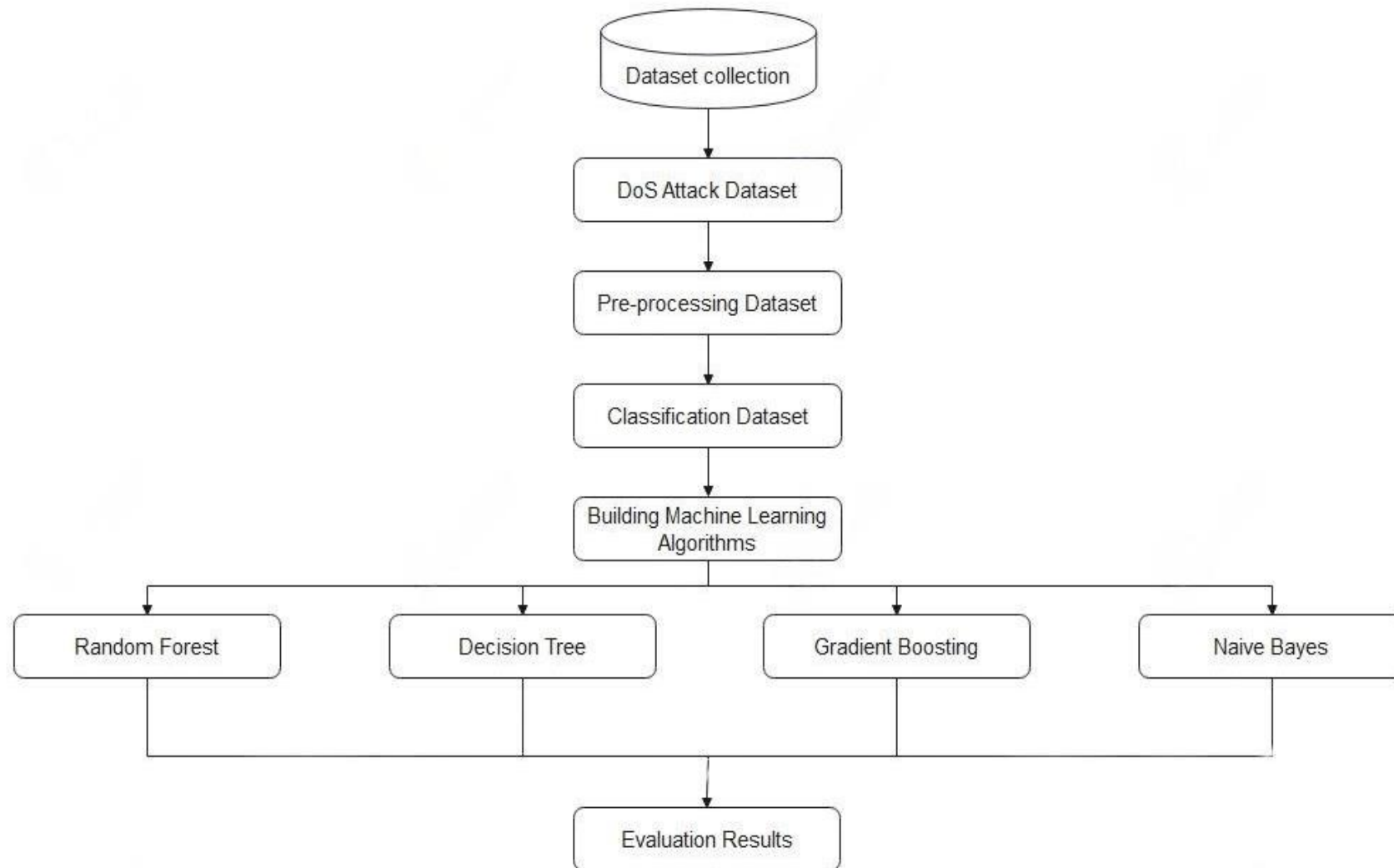
Software Requirements

- **OS:** Windows 10.
- **Framework:** Visual Studio, PyCharm
- **Server:** Localhost.

Hardware Requirements

- **Processor:** Intel Quad core 1.7 GHZ Processor or above.
- **RAM:** Minimum 4 GB of RAM.

System Design



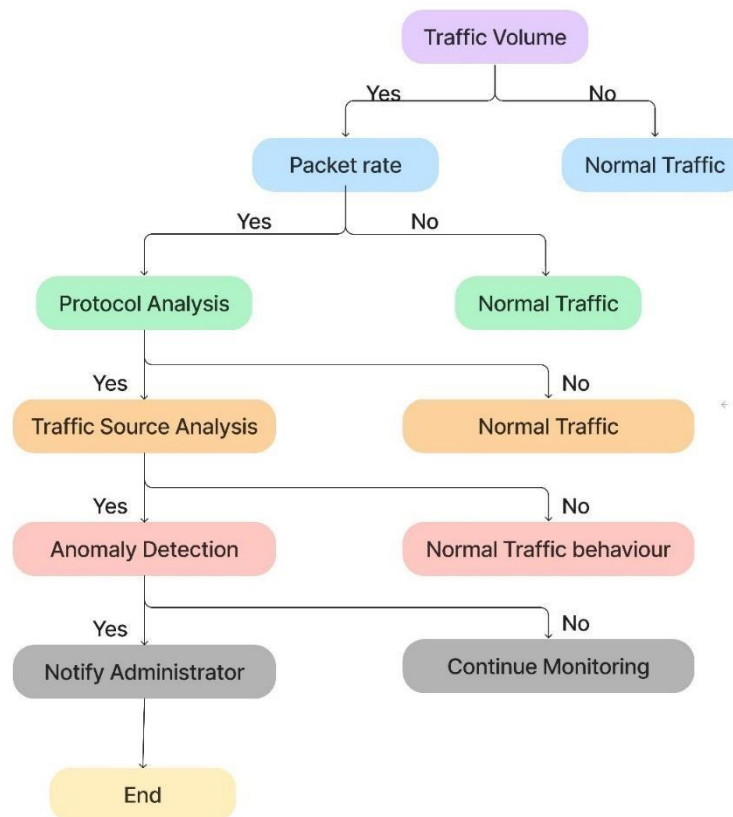
System Feature

System Features :

- 1. Real-time Monitoring:** Continuous and real-time analysis of network traffic patterns to promptly identify potential DoS attacks.
- 2. Machine Learning Algorithms:** Integration of adaptable machine learning models for accurate anomaly detection, improving system resilience.
- 3. Scalability:** Ability to handle growing network traffic loads and adapt to diverse network environments.
- 4. Low False Positives:** Minimization of false positive alerts to reduce the burden on network administrators and resources.

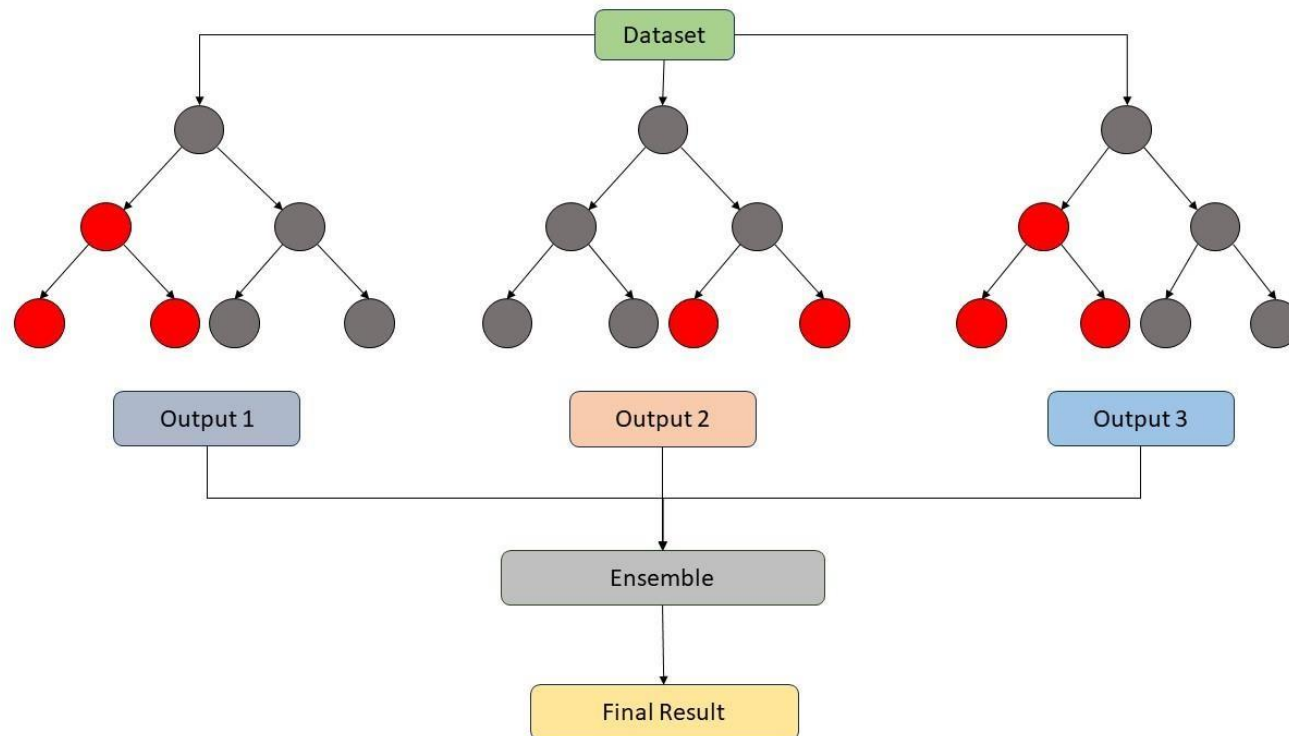
Algorithms

1. Decision Tree: A decision tree algorithm recursively partitions data into subsets based on feature values, creating a tree-like structure. It's used for classification and regression, making decisions by evaluating attribute conditions at each node.



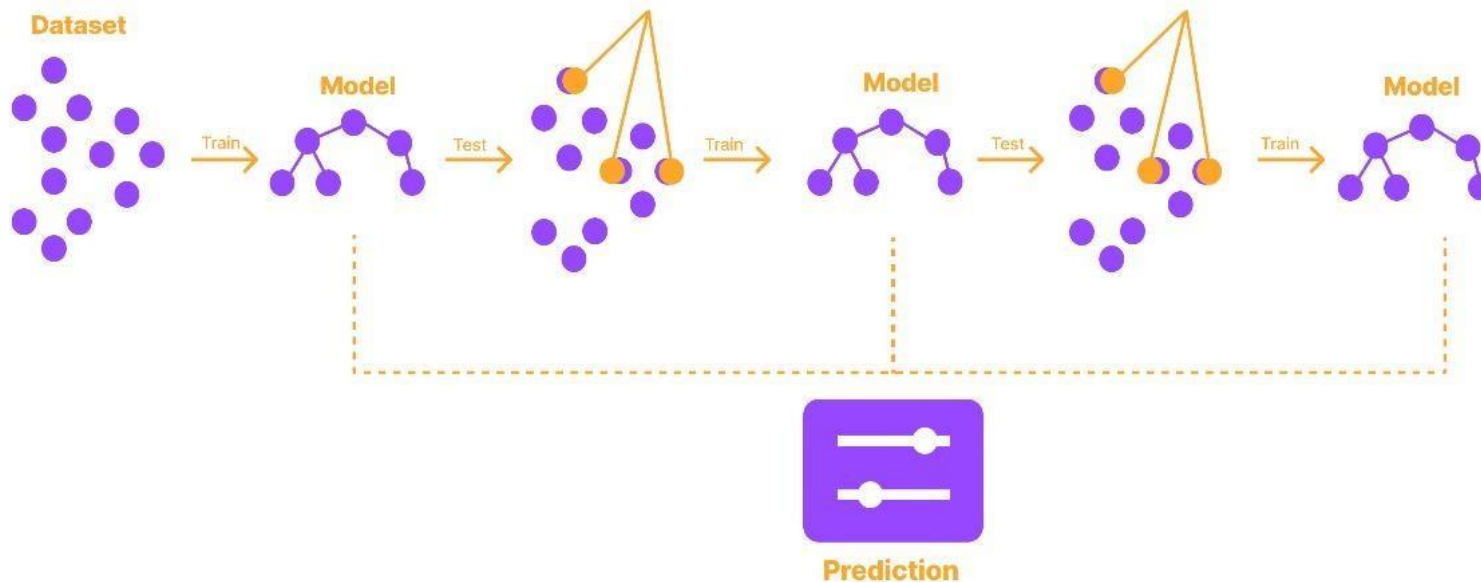
Algorithms

2. Random Forest: Random Forest is an ensemble learning technique that combines multiple decision trees to reduce overfitting. It aggregates their results to improve predictive accuracy and is widely used for classification and regression tasks.



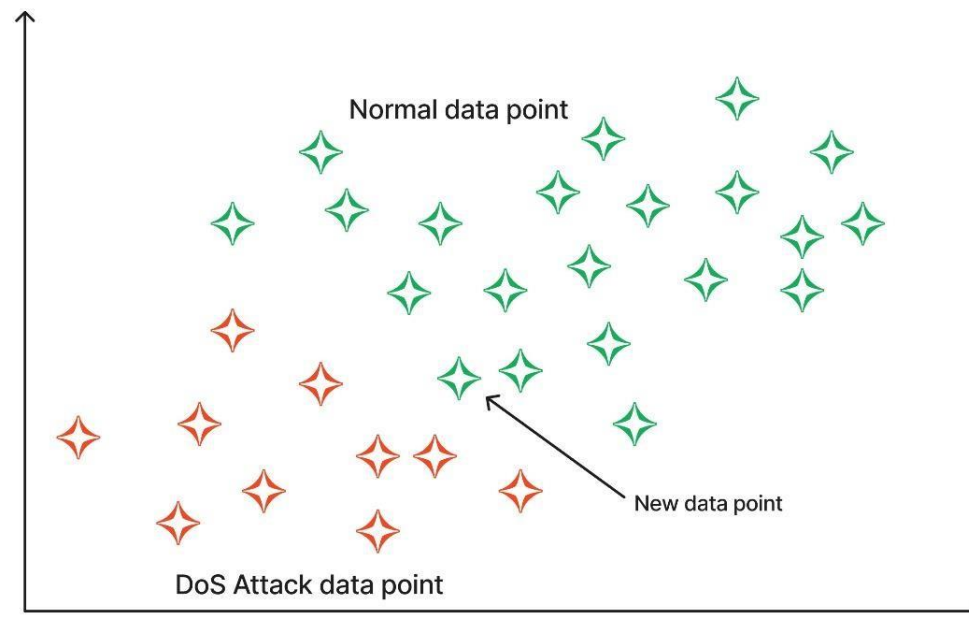
Algorithms

3. Gradient Boosting: Gradient Boosting is an ensemble method that builds decision trees sequentially, emphasizing correcting errors made by the previous trees. It improves predictive accuracy by combining the outputs of these trees.



Algorithms

4. Naive Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes feature independence and calculates probabilities to classify data, often used for text classification and spam filtering.



Dataset and Result Analysis

❑ Dataset Source:

<https://www.kaggle.com/>

❑ Result Analysis:-

Algorithm	Accuracy	Precision	Recall	F1 Score
RF	99.3	97	96.8	98.2
DT	98	98	98.7	98.2
GB	97	96.8	96.7	98
NB	92.7	78.9	100	92.4

Testing of Project:- Apply various testing strategy

Testing of Project:- Apply various testing strategy:

1. **Unit Testing:** Test individual components of the system, such as data preprocessing, feature extraction, and machine learning algorithms to ensure they function correctly.
2. **Integration Testing:** Verify that the different system modules work seamlessly together, checking data flow and interactions.
3. **Functional Testing:** Ensure that the system meets its functional requirements, including real-time monitoring, anomaly detection, and automated response.
4. **Performance Testing:** Assess the system's ability to handle high volumes of network traffic, measuring response times and resource usage.
5. **Security Testing:** Evaluate the system's resistance to known attack patterns, ensuring it can effectively detect and respond to DoS attacks.
6. **Scalability Testing:** Test how well the system scales with increased network traffic and data, ensuring it can handle growth.
7. **Usability Testing:** Collect feedback from users and administrators to assess the system's user-friendliness and the effectiveness of its reporting and alerting mechanisms.

Constraints/Limitation

Constraints/Limitations :

1.Data Quality: The effectiveness of machine learning models heavily relies on the quality and quantity of training data, and obtaining labeled data for DoS attacks can be challenging.

2.Scalability: As network traffic grows, the computational resources required for real-time analysis can become a limiting factor.

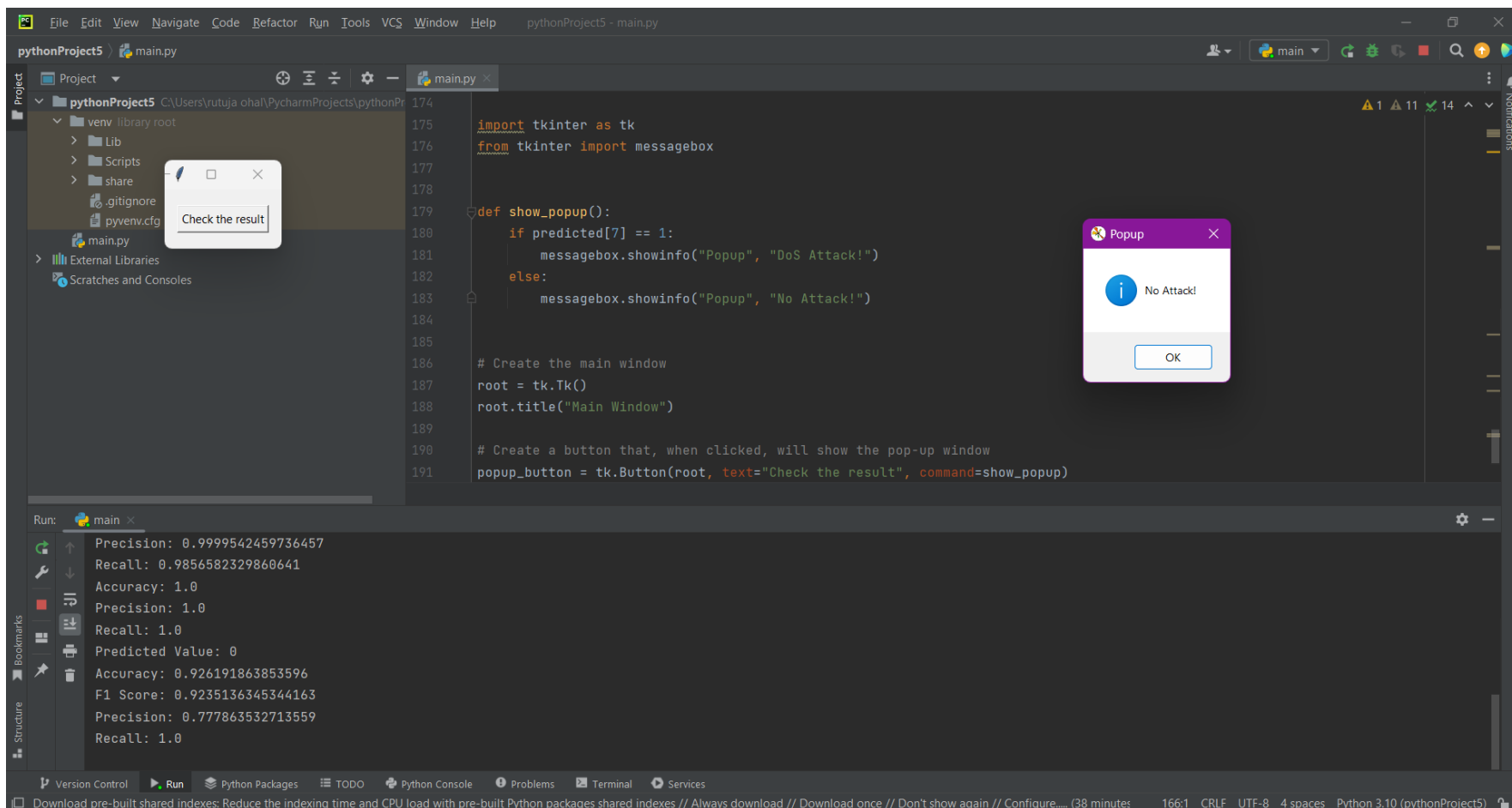
3.False Positives: Despite advancements, machine learning models may still produce false positives, potentially leading to unnecessary alerts and resource wastage.

Applications

1. **Network Security:** Detecting DoS attacks using machine learning helps safeguard networks and data from malicious intrusions, ensuring data confidentiality and integrity.
2. **E-commerce:** E-commerce platforms employ these algorithms to protect against DoS attacks, ensuring uninterrupted online shopping experiences for customers.
3. **Cloud Services:** Cloud providers use machine learning-based detection to secure their infrastructure against DoS threats, ensuring high availability for clients.
4. **IoT Security:** Machine learning algorithms help identify and mitigate DoS attacks on IoT devices, enhancing the security of interconnected smart devices and systems.

Result/Outcomes

Output for No Attack



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `main.py` with the following code:

```
174
175 import tkinter as tk
176 from tkinter import messagebox
177
178
179 def show_popup():
180     if predicted[7] == 1:
181         messagebox.showinfo("Popup", "DoS Attack!")
182     else:
183         messagebox.showinfo("Popup", "No Attack!")
184
185
186 # Create the main window
187 root = tk.Tk()
188 root.title("Main Window")
189
190 # Create a button that, when clicked, will show the pop-up window
191 popup_button = tk.Button(root, text="Check the result", command=show_popup)
```

On the left, the Project tool window shows the file structure of `pythonProject5`, including `venv`, `Lib`, `Scripts`, `share`, `.gitignore`, `pyvenv.cfg`, and `main.py`. A small window titled "Check the result" is visible over the file explorer.

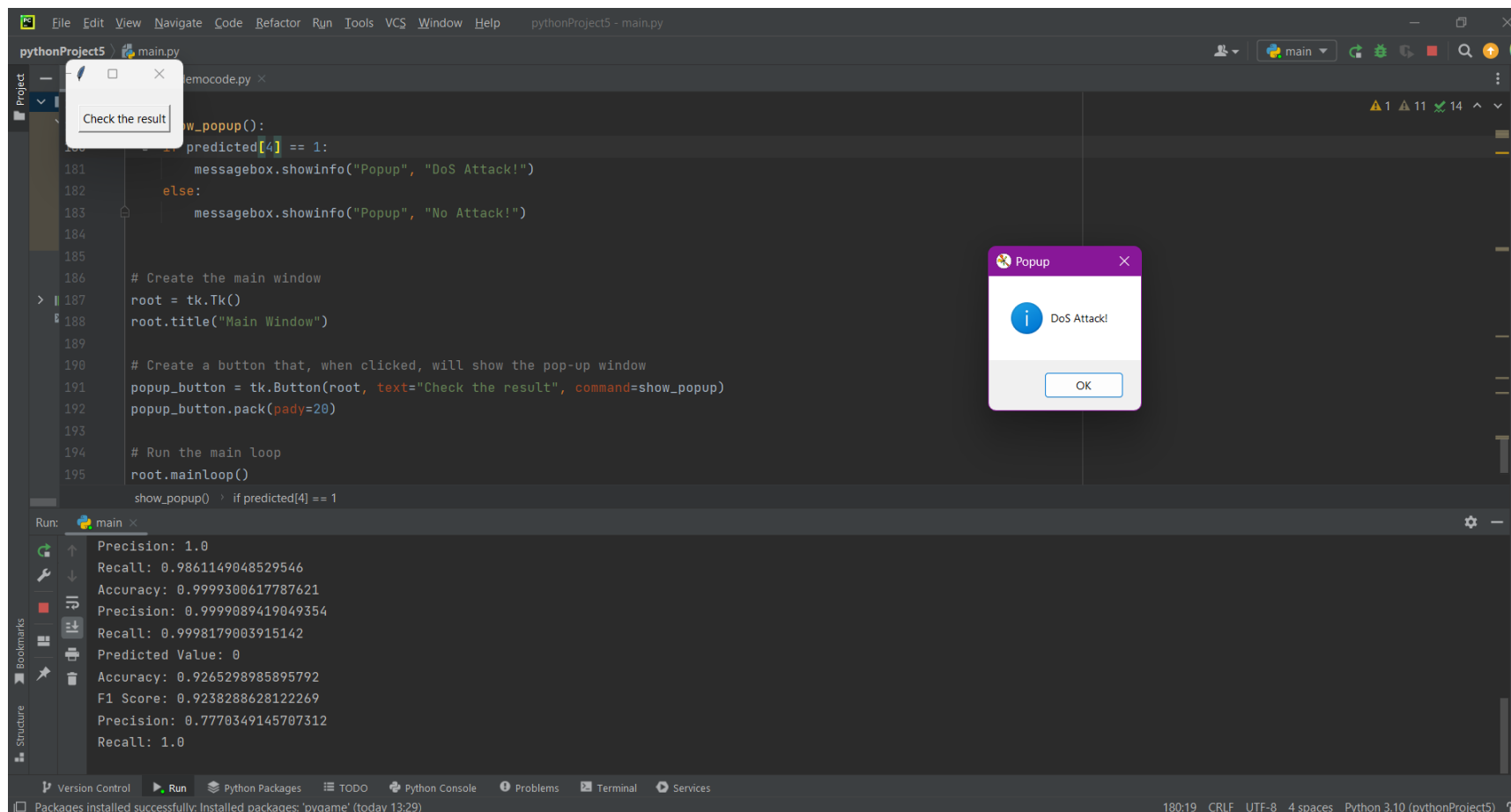
On the right, a "Popup" dialog box is displayed with the message "No Attack!" and an "OK" button.

At the bottom, the Run tool window shows the output of the program:

```
Precision: 0.9999542459736457
Recall: 0.9856582329860641
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
Predicted Value: 0
Accuracy: 0.926191863853596
F1 Score: 0.9235136345344163
Precision: 0.777863532713559
Recall: 1.0
```

Result/Outcomes

Output for DoS Attack



```
pythonProject5 \ main.py
pythonProject5 \ memocode.py x
180 def show_popup():
181     predicted[4] == 1:
182         messagebox.showinfo("Popup", "DoS Attack!")
183     else:
184         messagebox.showinfo("Popup", "No Attack!")
185
186 # Create the main window
187 root = tk.Tk()
188 root.title("Main Window")
189
190 # Create a button that, when clicked, will show the pop-up window
191 popup_button = tk.Button(root, text="Check the result", command=show_popup)
192 popup_button.pack(pady=20)
193
194 # Run the main loop
195 root.mainloop()
show_popup() if predicted[4] == 1

Run: main x
Precision: 1.0
Recall: 0.9861149048529546
Accuracy: 0.9999300617787621
Precision: 0.9999089419049354
Recall: 0.9998179003915142
Predicted Value: 0
Accuracy: 0.9265298985895792
F1 Score: 0.9238288628122269
Precision: 0.7770349145707312
Recall: 1.0
Packages installed successfully: Installed packages: 'pygame' (today 13:29)
```

Check the result

Popup

DoS Attack!

OK

Conclusion

Conclusion:-

Security threats are evolving and getting more hidden and complicated. Detecting malicious security threats and attacks have become a huge burden to cyberspace. We should apply proactive prevention and early detections of security vulnerabilities and threats rather than patching security holes afterwards. To analyze large amount of data to find out suspicious behaviors, threat patterns, and vulnerabilities and to predict and prevent future cybersecurity threats are a challenge. Machine Learning (ML) is a powerful instrument to take up such challenge.

In this report, we used dataset to classify denial of service attack by using Gradient Boosting, Support Vector Machine, Naive Bayes, K-Nearest Neighbors and compare their performance. The experimental results show that the Gradient Boosting, Support Vector Machine performed better than Naive Bayes and K-Nearest Neighbors algorithm in the dataset with slightly imbalanced distribution.

References

1. S. Banerjee and P. S. Chakraborty, "Proposed approach to detect distributed denial of service attacks in software defined network using machine learning algorithms," International Journal of Engineering Technology, vol. 7, no. 2.8, p. 472, Mar. 2018, doi:10.14419/ijet.v7i2.8.10488.
2. Christina, S. Karpagavalli, G. Suganya, "Email spam filtering using supervised machine learning techniques," Int. J. Comput. Sci. Eng., 02 (09) (2010), pp. 3126-3129.
3. Y. Mirsky, N. Kalbo, Y. Elovici, and A. Shabtai, "Vesper: Using echo analysis to detect man-in-the-middle attacks in LANs," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1638–1653, Jun. 2019, doi: 10.1109/TIFS.2018.2883177.
4. P. S. Saini, S. Behal, and S. Bhatia, "Detection of DDoS attacks using Machine learning algorithms," in 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), Mar. 2020, pp. 16–21, doi: 10.23919/INDIACom49435.2020.9083716.
5. Chunhui Bao, Yifei Pu, and Yi Zhang, "Fractional-Order Deep Backpropagation Neural Network," Computational Intelligence and Neuroscience, Vol. 2018, Article ID 7361628, 2018.
6. S. Banerjee and P. S. Chakraborty, "Proposed approach to detect distributed denial of service attacks in software defined network using machine learning algorithms," International Journal of Engineering Technology, vol. 7, no. 2.8, p. 472, Mar. 2018, doi:10.14419/ijet.v7i2.8.10488.

Paper Submission Deatils

Paper Submission Details:-

Paper Submission Status:- Submitted

**EAI ICISML 2024 - EAI 3rd International
Conference on Intelligent Systems and Machine
Learning**

Journal Name:- EAI (Publishing Partner:- Springer)

Link :- <https://icisml.eai-conferences.org/2024/>

Engineering

Management

Law

Schools

Other Courses

■ NAGPUR ■ PUNE ■ JALGAON ■ AMRAVATI ■ AHMEDNAGAR ■ CHHINDWARA



Thank you !

Engineering

Management

Law

Schools

Other Courses

■ NAGPUR ■ PUNE ■ JALGAON ■ AMRAVATI ■ AHMEDNAGAR ■ CHHINDWARA



RAISONI
GROUP OF INSTITUTIONS