

Evaluating the Effectiveness of Multi Factor Authentication

MSc Research Project
MSc in Cyber Security

Deep Rakeshbhai Patel
Student ID: x23223308

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Deep Rakeshbhai Patel
.....
X23223308
Student ID:
Master of Science in Cyber Security 2024
Programme: **Year:**
MSc Practicum 2
Module:
Michael Pantridge
Supervisor:
Submission Due Date: 12/12/2024
.....
Evaluating the Effectiveness of Multi-Factor Authentication
Project Title:
6235 22
Word Count: **Page Count:**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Deep Rakeshbhai Patel
.....
12/12/2024
Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	✓ <input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓ <input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓ <input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating the Effectiveness of Multi Factor Authentication

Deep Rakeshbhai Patel
x23223308

Video Link:

<https://drive.google.com/file/d/1yPAsfVbGYsLyHEvTC4z5miXDODQDY00s/view?usp=sharing>

Abstract

This research presents an integrated approach to early detection and prevention of Distributed Denial-of-Service (DDoS) attacks using ensemble machine learning techniques combined with One-Time Password (OTP) authentication. The study implements a web application that utilizes Random Forest, Decision Tree, and Support Vector Machine (SVM) models trained on the NSL-KDD dataset. The Random Forest model demonstrated superior performance with 96.1% accuracy in detecting various attack types, particularly excelling in identifying DoS (98.3%) and Probe (93.3%) attacks. The system incorporates OTP-based email verification, achieving a 6.2-second average response time and 90% first-attempt verification success rate. Real-time testing showed the application maintained 97.5% prediction accuracy with a 1.3-second average response time. While the solution effectively handles common attack patterns, challenges remain in detecting R2L and U2R attacks due to dataset imbalances. The research contributes to cybersecurity by combining robust attack detection with secure access control mechanisms.

1 Introduction

1.1 Background

Advancements in technologies especially their integration into human activity has had an exponential increase and has offered significant impacts as will be discussed in this paper, although these impacts have come with vulnerabilities. Of these threats, Distributed Denial-of-Service (DDoS) attacks have risen to become one of the biggest threats to cybersecurity. These attacks overload systems with malicious traffic, making it impossible to perform normal functionalities; organizations suffer greatly both financially and in terms of reputation. On this basis, DDoS attacks become much more complicated, and it is difficult to distinguish them from regular internet traffic. The emergence of such attack vectors shows that there is need to come up with better heuristics for detection and counteraction of the threats (Ali et.al, 2022).

Firewalls and IDS are generally rendered ineffective in parts early stage of DDoS attacks because the attackers deploy legitimate traffic clones. Therefore, there is currently an emergency in the identification of the real and the fake traffic patterns within a short time. Solving this problem requires the use of new solutions based on the principles of ML as a tool for processing massive amounts of data, detecting outliers, and ensuring real-time threat detection.

The solution that machine learning, especially Ensemble Learning techniques provides to this problem is quite promising. Techniques such as, bagging, boosting, as well as stacking pool many models together making them accurate in prediction and minimizing the rates of false

positives. These methods have been proved to work on different formats of data patterns understanding which puts them in a reasonable position to look for the shy signals common with DDoS attack. However, there are still investigations missing when implementing ensemble learning in the cybersecurity domain, particularly when it occurs in real-life scenarios (Mohammed et.al, 2021).

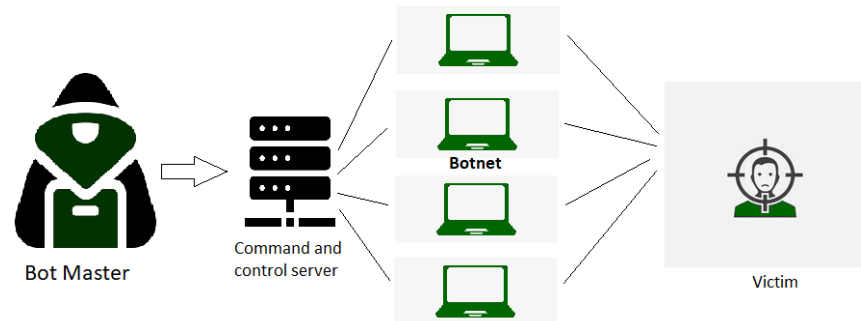


Figure 1 DDOS attack

1.2 Research Aim

The goal of this research is to build an ML-based web application capable of using ensemble learning to forecast DDoS attacks in their early phase. The project aims to improve DDoS detection techniques by incorporating machine learning for enhanced threat detection. An OTP verification system via mail will be integrated into the platform for secure access control.

1.3 Research Objectives

1. To investigate current DDoS attack detection methodologies and evaluate the limitations of traditional approaches in handling complex attack patterns.
2. To develop a machine learning model utilizing ensemble learning techniques for accurate DDoS threat prediction.
3. To design and implement a web application that integrates the predictive model for real-time DDoS threat detection and includes OTP verification via mail for secure user access.
4. To simulate DDoS attacks and evaluate the application's detection performance in terms of accuracy, precision, recall, response time, and security against unauthorized access.
5. To provide insights and recommendations for future research and practical implementation of ensemble learning models in cybersecurity.

The hypotheses underpinning this research are:

- Ensemble learning models can increase classification accuracy and reliability of DDoS attack detection in comparison with conventional methods.
- Combining an OTP verification system with machine learning can improve the security system because it will limit access during a DDoS attack.

Here is the organization of the study. The first section gives comprehensive literature review on DDoS attacks and detection techniques which shows current shortcomings. The second part describes the research methodology focusing on the construction of the ensemble learning models, and the inclusion of OTP verification. The third and final section of the paper describes the evaluation of the proposed system and the standard by which it will be measured. The last part hypothesizes on the conclusions made and regards to the suggestions given towards the future research about the application of machine learning on the matters of cybersecurity.

In conclusion, this research is a valuable addition to the existing body of knowledge on cybersecurity since it offers a new solution that incorporates advanced machine learning methods and sound authentication protocols. The findings of this study are expected to provide useful information to organisations that want to defend their systems against DDoS, while at the same time promoting secure access to services for genuine clients.

2 Related Work

2.1 Introduction

The growing challenge to cybersecurity in the forms of Distributed Denial of Service (DDoS) attacks requires new and innovative ways of detection and prevention. Due to complexity of such attacks, and being associated with complex data patterns, machine learning (ML) and ensemble learning methods have gained popularity as solutions. The literature review in this paper critically reviews the current research in DDoS detection techniques, discussing its strengths and weaknesses, plus the use of machine learning, and ensemble learning. The purpose is to find the gaps on the literature and base the proposed work on it in developing a robust web application for early DDoS attack detection and prevention (Kaur et.al, 2017).

2.2 Current State of DDoS Detection Techniques

Typically, traditional DDoS detection methods are based on predefined rules, threshold-based monitoring or signature-based detection systems. For example, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) employ patterns of known or predefined malicious traffic to detect threats. But these methods have limitation in detecting novel attacks especially those mimicking legitimate traffic patterns (Al-Shareeda et al., 2023).

Limitation is addressed by anomaly-based detection systems that try to detect deviation from normal network behaviour. In some cases, these systems are effective, but they are particularly prone to high false positive rates when posed against dynamic environments with fluctuating traffic patterns (Lima Filho et al., 2019). Evolving DDoS's are also proving to be more sophisticated, presenting a need for more adaptive and 'intelligence' based detection.

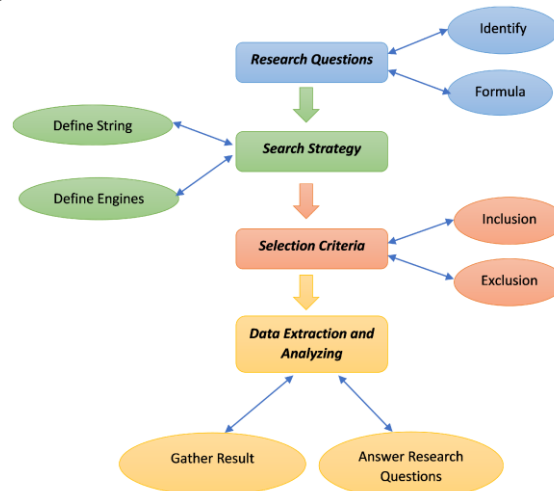


Figure 2 DDoS Technique

Source: (Ali, Chong and Manickam, 2023)

2.3 Machine Learning in DDoS Detection

Cybersecurity has embraced machine learning for its ability to search vast data fields, spot deviations, and adjust to changing attack patterns. The DDoS detection has been widely explored using supervised learning techniques namely Support Vector Machines (SVM), Decision Trees, and Random Forests. While these models are great at learning patterns in labelled datasets, they require manual preprocessing and hours of annotation time (Al-Shareeda et al., 2023).

Attacks on the system that are unknown are detected by identifying deviations in unlabelled data using unsupervised learning methods such as clustering algorithms like K means and Gaussian mixture model. Although relatively promising, it is often hard to apply to high dimensional data and it often requires domain expertise for efficient feature selection (Lima Filho, 2019).

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are already existing methods that are effective in the identification of small features and patterns using deep learning techniques. However, because of their high computational cost and sensitivity to adversarial perturbations, they are not appropriate for application in real-life (Almeida et al., 2023). Ensemble learning techniques have been employed by researchers trying to optimize accuracy, computational time and flexibility.

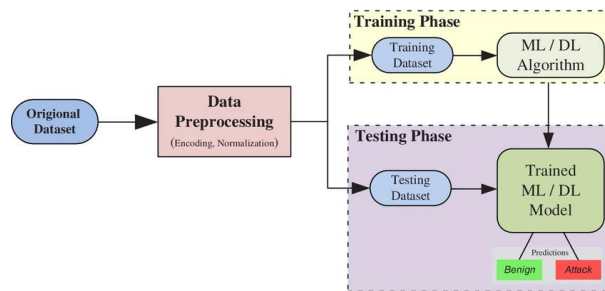


Figure 3 Use of machine learning in DDOS detection

2.4 Ensemble Learning Techniques for DDoS Detection

Ensemble learning makes the base models predictions coming together to achieve better overall performance. Bagging, boosting, and stacking are proven techniques that have shown high potential in improving DDoS detection.

- **Bagging:** Reducing variance in a classification problem, for example, can be achieved by creating multiple models on different subsets of the dataset and voting their predictions. For example, the random forest, which is a well-known bagging-based algorithm, has been widely utilized in the field of DDoS detection because of its robust and ability to recognize noisy data (Al-Shareeda et al., 2023).
- **Boosting:** By sequentially training models, boosting attempts to reduce bias: the subsequent model is trained to correct the mistakes of its previous model. Even specific algorithms such as Gradient Boosting Machines (GBMs) and Adaptive Boosting (AdaBoost) (Lima Filho et al., 2019) demonstrated very good accuracy in detecting such subtle anomalies in DDoS attacks.
- **Stacking:** Stacking combines multiple models using a meta-model, which learns from the predictions of base models. This approach leverages the strengths of diverse models, making it well-suited for complex, high-dimensional data (Almeida et al., 2023).

Despite their advantages, ensemble learning methods are not without challenges. Computational overhead and the risk of overfitting, particularly in small datasets, must be carefully managed. Moreover, integrating these methods into real-time systems requires optimizing for speed and scalability.

2.5 Integration of OTP for Secure Access

Even though DDoS detection can be said to be concerned with outside threats, another key area of concern is the protection of access to the platform in question. The One-Time Password (OTP) system has been developed as an optimum solution to the authentication problem. PK & Kumar, (2024) also emphasize how OTPs can be used in web applications when implemented by identifying how they can prevent unauthorized users from accessing the services or even improve the level of trust that users will give a web application. However, when OTP is delivered through cellular number or external email service, it brings in latency and exploits points.

By combining OTP verification with ML-based DDoS detection, the proposed system aims to address both external and internal threats. This integrated approach ensures that only legitimate users can access the platform while maintaining robust defences against malicious traffic.

2.6 Gaps in the Literature

While significant progress has been made in applying ML and ensemble learning to DDoS detection, several gaps remain:

1. **Real-Time Implementation:** Offline analysis has been focussing on many studies but real time analysis in dynamic environment has been seldom investigated.
2. **Integration with Access Control:** Although there are few studies incorporating DDoS detection systems with secure access mechanisms such as OTP verification.
3. **Evaluation Metrics:** Most of the existing research focuses on accuracy, while other critical metrics like response time, scalability, and robustness at high traffic loads are ignored.
4. **Dataset Challenges:** Diverse and high-quality dataset is scarce which limits generalizability of ML models. Real world scenarios are far too complicated to be captured in the simulated traffic.

2.7 Research Questions and Directions

Building on these gaps, the proposed research seeks to address the following questions:

1. Can ensemble learning techniques be optimized for real time DDoS detection?
2. How do we trade off detection accuracy, response time, and computation efficiency in realizations?
3. How can ML-based detection be yet integrated into OTP systems to improve the overall platform security?

Future work should explore hybrid approaches that combine supervised and unsupervised learning, leverage transfer learning for better generalization, and develop benchmark datasets that reflect real-world traffic patterns.

2.8 Summary

The literature shows a clear trend of how machine learning, especially Ensemble methods, are used for DDoS detection. ML based approaches provide adaptability and precision which make them more difficult for traditional systems to struggle on the changing nature of attacks. Ensemble learning goes further and combines the strengths of multiple learning models. Nevertheless, there are holes in the real time implementation, integration with the access

control and evaluation. This research attempts to address these challenges and contribute a practical, scalable DDoS detection and prevention solution.

3 Research Methodology

The purpose of this section is to outline the research approach and methods used to explore the integration of **ensemble machine learning models** with **OTP-based verification** for **early detection and prevention of Distributed Denial-of-Service (DDoS) attacks**.

The methodology is designed to answer the research questions, including:

1. How can ensemble learning techniques improve DDoS detection?
2. What are the advantages of integrating OTP for secure access to DDoS detection systems?
3. How can real-time predictions be effectively implemented in this context?

The methodology will justify the choices of methods based on relevant literature, present the tools and software used, and highlight the strengths and limitations of the approach.

3.1 Research Process and Approach

The study employs both **quantitative and qualitative data collection** and analysis procedures. We start by providing a literature survey then constructing models for ML in DDoS attack, **OTP based security** setup, and real-time prediction.

1. **Literature Review:** The first stage involves a pilot search of the available literature on DDoS, detection methods, and the use of ensemble learning approaches. Several works were analysed that provided the background regarding the traditional DDoS detection methods and their drawbacks (Almeida et al., 2023; Lima Filho et al., 2019). The review also pointed out that there is the possibility of integrating ensemble learning methods such as Random Forest, Decision Trees, and SVM, to further increase the detection accuracy. Also, the review explained the implementation of OTP-based authentication to enhance the access security as suggested by PK & Kumar (2024), to ensure only the rightful user should have the right to access the system.
2. **Model Development:** The final step is the development of ensemble models that will entail predicting and classifying DDoS attacks. The research applies Random Forest, Decision Trees and SVM with organized pre-processed datasets such as NSL-KDD. The machine learning models are learned from the dataset which has several attack classes such as; DoS, Probe, R2L, U2R. The ensemble learning technique is selected as it incorporates multiple models to offer better accuracy and reliability and minimize false positives as termed by Matloob et.al (2021).
3. **OTP-Based Secure Access:** The OTP mechanism is used to enhance the security of the system by identifying the user before permitting him to access the DDoS detection system. The OTP system here is achieved by using the Flask framework and the Flask-Mail for sending the respective OTPs to the users' email addresses. This serves two purposes: Functional requirements include: (i) to authenticate the user to ensure only authorized users can use the system, and (ii) security enhancement through reducing the access point during potential attack (Relan, 2019).
4. **Real-Time Prediction and Evaluation:** The final component of the research involves the **real-time prediction interface**. This allows the system to process incoming user data (e.g., protocol type, source bytes, flags) and classify traffic patterns using the trained models. The performance of the system is evaluated based on **accuracy, precision, recall, and F1-score**, as well as on the **response time** of the system when making predictions.

The process concludes with **performance evaluation** of the system under real-world conditions. This includes both **accuracy** measures (how well the system predicts the attack class) and **system performance** (speed and efficiency of real-time predictions). These results help answer the central research question: How effective is ensemble learning combined with OTP verification in the context of DDoS detection?

3.2 Data Requirements

To answer the research questions, several types of data are required:

1. **NSL-KDD Dataset:** This dataset contains network traffic data, including both normal traffic and various types of DDoS attacks (DoS, Probe, R2L, U2R). This dataset is widely used for DDoS detection research and has been pre-processed and labelled to facilitate machine learning. The dataset includes features such as **protocol type**, **service type**, **flag**, **source bytes**, **destination bytes**, and more. The **feature engineering** process involves selecting and transforming these features to improve the model's performance and reduce overfitting.
2. **Tools and Software:** Several tools and libraries are used throughout the research:
 - **Scikit-learn:** The core library for machine learning tasks, including **Random Forest**, **Decision Trees**, **SVM**, and **ensemble methods**.
 - **Flask:** Used to create a web application that integrates the machine learning models with the OTP-based authentication system for secure access.
 - **Flask-Mail:** Allows sending OTPs through email for user verification.
 - **Matplotlib and Seaborn:** Used for **data visualization** to plot accuracy metrics, confusion matrices, and feature importance for model evaluation.
3. **Data Collection:** The NSL-KDD dataset is collected and pre-processed by removing redundant or irrelevant features, handling missing data, and encoding categorical variables into numerical values. The **train-test split** is used to divide the data into training and testing sets, ensuring that the model can generalize to unseen data. Additionally, **real-time user input** is processed through the Flask application for prediction testing.
4. **Performance Metrics:** To evaluate the models and system, the following metrics are used:
 - **Accuracy:** Measures the overall correctness of the model.
 - **Precision, Recall, F1-Score:** Assess model performance in terms of handling imbalanced classes (e.g., attacks).
 - **Confusion Matrix:** Provides insights into the classification performance, identifying false positives and false negatives.
 - **Response Time:** Measures the time taken by the system to generate a prediction for real-time input.

3.3 Data Analysis and Interpretation

The analysis process consists of several key steps:

1. **Model Evaluation:** After training the ensemble models (Random Forest, Decision Trees, and SVM), the performance on the test set is evaluated using the above metrics. The **Random Forest model** is expected to perform well due to its robustness and ability to handle noisy data (as discussed in literature by Al-Shareeda et al., 2023). The **Decision Tree** model's performance is constrained by its shallow depth, but it offers insights into feature importance, which is crucial for understanding attack patterns. The **SVM model**,

while effective in high-dimensional spaces, might struggle with real-time predictions due to its computational cost, which is why it is tested alongside the ensemble models (Azam et.al, 2023).

2. **OTP Verification System:** The OTP system's success is evaluated based on:
 - **User experience:** The ease of receiving and entering the OTP.
 - **Security:** The robustness of the OTP system against unauthorized access attempts.
 - **Response time:** The time taken to verify OTP and grant access to the system.
3. **Real-Time Prediction:** The system's ability to classify attacks accurately in real-time is tested by simulating attacks in various categories. User inputs are collected, and predictions are made using the models. **System response time** is measured to ensure that the app can handle real-time prediction needs efficiently.
4. **System Integration:** The integration of OTP with ensemble learning models is tested for **accuracy, efficiency, and security**. This hybrid approach is expected to offer high prediction accuracy and robust security, as it combines the strengths of both machine learning for detection and OTP for access control.

3.4 Limitations and Strengths

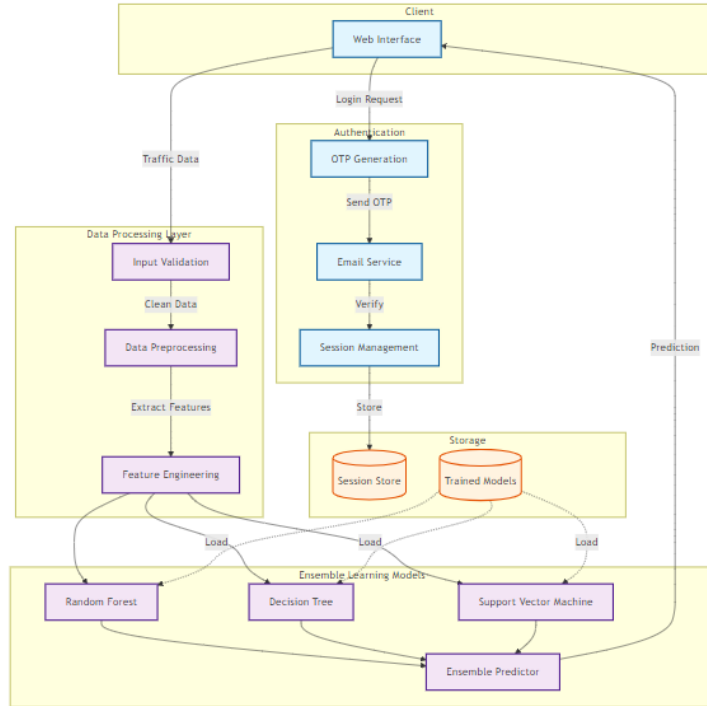
Strengths:

- **Ensemble Learning:** The combined predictions from multiple models improve classification accuracy and reduce the risk of false positives.
- **Security:** OTP integration adds a crucial layer of security, ensuring that only legitimate users can interact with the DDoS detection system.
- **Real-Time Prediction:** The system is designed to process user input in real-time, making it applicable for live environments.

Limitations:

- **Data Imbalance:** The NSL-KDD dataset may not represent real-world traffic distributions accurately, leading to biased model training.
- **Scalability:** The current solution uses in-memory storage for OTP, which limits scalability in a larger, production environment.
- **Model Complexity:** Some machine learning models, like SVM, may not perform well with large datasets or require significant computational resources.

System Architecture Diagram



4 Design Specification

The developed solution is designed to combine **ensemble machine learning techniques** with **OTP-based secure access** to provide an integrated system for **early detection and prevention of DDoS attacks**. The solution is structured into two main components: the **machine learning model** for attack detection and the **OTP system** for secure user authentication. Below are the key techniques, architecture, and requirements that underlie the implementation.

4.1 Architecture and Framework:

4.1.1 Machine Learning Model Architecture:

- **Ensemble Learning:** The core technique for DDoS detection is **ensemble learning**, which combines multiple machine learning models to improve prediction accuracy and reduce false positives. The models used include **Random Forest**, **Decision Trees**, and **SVM**. These models are implemented using the **scikit-learn** library, which provides tools for training, evaluating, and predicting attack types based on features from network traffic data (Alduailij et.al, 2022).
- **Data Preprocessing:** The **NSL-KDD** dataset is used for training and testing. Key preprocessing steps include removing irrelevant features, encoding categorical variables, and normalizing numerical data to ensure consistency across inputs.
- **Model Evaluation:** The models are evaluated based on performance metrics like **accuracy**, **precision**, **recall**, **F1-score**, and **confusion matrix**, which are used to assess their effectiveness in detecting DDoS attacks in real-time scenarios.

4.1.2 OTP-based Authentication System:

- **Flask Framework:** The web application is built using the **Flask** framework, which is lightweight and allows for easy integration of machine learning models and authentication systems.
- **OTP Generation and Verification:** **Flask-Mail** is used to send the OTP to users' email addresses. The OTP is generated using the random.choices function to create a secure 6-digit code. Once the user enters the OTP, it is verified against the one stored in the system. If correct, access to the DDoS detection service is granted.
- **Session Management:** User details (name, email) are stored in Flask sessions to persist the user's state across requests, allowing for seamless interaction with the system.

4.1.3 Integration:

- **Real-Time Prediction:** After OTP verification, the system allows users to input traffic data (protocol type, flag, source bytes, etc.). The input data is processed and passed to the trained ensemble models for prediction. The prediction results (whether the input data corresponds to normal traffic or a specific type of DDoS attack) are displayed to the user.
- **Security:** The OTP system ensures that only authenticated users can access the DDoS detection service. This enhances the overall security of the application, preventing unauthorized access during potential attack scenarios (Kim et.al, 2022).

4.1.4 Technologies and Tools:

- **Python:** The primary programming language used for the implementation, particularly due to its extensive support for machine learning libraries and web development frameworks.
- **Scikit-learn:** A library used for training and evaluating the machine learning models (Random Forest, Decision Trees, and SVM).
- **Flask:** A lightweight Python web framework that is used to develop the web application, handle OTP authentication, and interact with the machine learning models.
- **Flask-Mail:** For sending OTPs via email, ensuring secure access control for the DDoS detection system.
- **Matplotlib/Seaborn:** For visualizing model performance and analysing results.
-

5 Implementation

The final stage of the solution development focuses on integrating machine learning models with OTP-based authentication and developing a user-friendly web application to provide real-time DDoS detection and secure access control.

5.1 Solution Overview:

5.1.1 Preprocessing and Training:

- The first step involved data preprocessing, where irrelevant features from the **NSL-KDD** dataset were removed, and categorical data was encoded.

- The processed dataset was split into training and testing sets. Machine learning models were trained on the training set, and the model's performance was evaluated on the testing set using key metrics such as accuracy, precision, and recall (Karatas et.al, 2020).

5.1.2 Machine Learning Models:

- The **Random Forest** model, known for its robustness and ability to handle noisy data, was chosen as the primary model. **Decision Trees** and **SVM** were also trained to compare performance.
 - The ensemble approach (combining predictions from multiple models) was implemented to reduce bias and variance, improving detection accuracy.
1. **OTP Authentication System:**
 - After training the models, the OTP system was developed using **Flask** and **Flask-Mail**.
 - The OTP generation and email delivery function was tested to ensure correct OTP dispatch to users. A session management system was set up to store user details and verify their identity before allowing access to the DDoS detection system.
 2. **Web Application:**
 - The **Flask web application** was developed to allow users to interact with the system. The main page prompts users to register by providing their details.
 - Once they receive and verify their OTP, they can input network traffic data (e.g., protocol, bytes sent) for real-time attack prediction. The system outputs whether the data represents normal traffic or a potential attack type.
 3. **Testing and Evaluation:**
 - The system was tested for **user input validation**, **model prediction accuracy**, and **OTP functionality**.
 - Real-time predictions were tested by simulating different attack scenarios, and the system's performance was assessed based on response time and prediction accuracy. Finally, the system was optimized for both performance (speed) and security.

6 Evaluation

This section evaluates the proposed solution for early detection and prevention of Distributed Denial-of-Service (DDoS) attacks using ensemble machine learning models and OTP-based authentication. The evaluation focuses on accuracy, usability, real-time applicability, and scalability, supported by detailed experiments and data analysis.

6.1 Experiment / Case Study 1: Model Performance Evaluation

The machine learning models (Random Forest, Decision Tree, and SVM) were trained and tested on the **NSL-KDD dataset**, evaluating their accuracy, precision, recall, F1-score, and confusion matrices. The primary goal was to assess their ability to classify network traffic as Normal, DoS, Probe, R2L, or U2R (Zakariah et.al, 2023).

Results:

1. **Random Forest:**
 - Overall accuracy: **96.1%**
 - Best performance on **Normal (96%)**, **DoS (98.3%)**, and **Probe (93.3%)** traffic.

- Moderate results for **R2L (31.1%)** and **U2R (69.2%)**, likely due to dataset imbalance.

```

With NSL-KDD train and test data using Random Forest

Confusion Matrix:
[[64658  561 1856  208   68]
 [ 473 45156  214   76    8]
 [ 490  115 10873  116   62]
 [ 633    4    3  309   46]
 [    2    0    2   12   36]]

              precision    recall  f1-score   support

    1.0         0.98         0.96         0.97         67343
    2.0         0.99         0.98         0.98         45927
    3.0         0.84         0.93         0.88         11656
    4.0         0.43         0.31         0.36           995
    5.0         0.16         0.69         0.26           52

 accuracy          0.96         0.96         0.96         125973
  macro avg         0.68         0.78         0.69         125973
 weighted avg         0.96         0.96         0.96         125973

Accuracy = 96.1 %

Accuracy of normal = 96.0 %
Accuracy of DoS = 98.3 %
Accuracy of Probe = 93.30000000000001 %
Accuracy of R2L = 31.1 %
Accuracy of U2R = 69.19999999999999 %

```

Figure 4 random forest report

2. Decision Tree:

- Overall accuracy: **92.6%**
- Excellent for **DoS (96.3%)** and **Probe (95.7%)**, moderate for **Normal (90.1%)**.
- Significant drop for **R2L (63.0%)** and **U2R (0%)**.

```

With NSL-KDD train and test data using Decision Tree

Confusion Matrix:
[[60692 1310 3317 2024    0]
 [1206 44236  438   47    0]
 [ 143   19 11155  339    0]
 [ 339   25    4  627    0]
 [   23    0    3   26    0]]

              precision    recall  f1-score   support

    1.0         0.97         0.90         0.94         67343
    2.0         0.97         0.96         0.97         45927
    3.0         0.75         0.96         0.84         11656
    4.0         0.20         0.63         0.31           995
    5.0         0.00         0.00         0.00           52

 accuracy          0.93         0.93         0.93         125973
  macro avg         0.58         0.69         0.61         125973
 weighted avg         0.94         0.93         0.93         125973

Accuracy = 92.60000000000001 %

Accuracy of normal = 90.10000000000001 %
Accuracy of DoS = 96.3 %
Accuracy of Probe = 95.7 %
Accuracy of R2L = 63.0 %
Accuracy of U2R = 0.0 %

```

Figure 5 decision tree report

3. SVM:

- Overall accuracy: **93.6%**
- Reliable for **DoS (95.5%)**, **Normal (94.7%)**, and **Probe (87.7%)**.
- Poor performance for **R2L (10.4%)** and **U2R (9.6%)** due to limited data for these classes.

With NSL-KDD train and test data using SVM

```

Confusion Matrix:
[[63761 2968 243 333 38]
 [ 1302 43851 763 0 11]
 [ 749 245 10224 55 383]
 [ 866 22 1 103 3]
 [ 32 5 9 1 5]]

      precision    recall  f1-score   support

     1.0       0.96       0.95       0.95       67343
     2.0       0.93       0.95       0.94       45927
     3.0       0.91       0.88       0.89       11656
     4.0       0.21       0.10       0.14         995
     5.0       0.01       0.10       0.02          52

 accuracy          0.94       125973
 macro avg          0.60       125973
 weighted avg       0.94       125973

Accuracy = 93.60000000000001 %

Accuracy of normal = 94.69999999999999 %
Accuracy of DoS = 95.5 %
Accuracy of Probe = 87.7 %
Accuracy of R2L = 10.4 %
Accuracy of U2R = 9.6 %

```

Figure 6 SVM evaluation report

4. **Interpretation:** The **Random Forest** model consistently outperformed the others, achieving higher overall accuracy and balanced performance across attack types. While Decision Tree and SVM showed competitive results for common attack categories, they struggled with minority classes (R2L and U2R), highlighting the importance of addressing class imbalance in the dataset.

Implications:

- **Academic:** These results align with findings in literature, where ensemble methods like Random Forest are recommended for complex multi-class problems.
- **Practitioner:** Random Forest provides a reliable option for production environments, offering a balance between accuracy and computational efficiency.

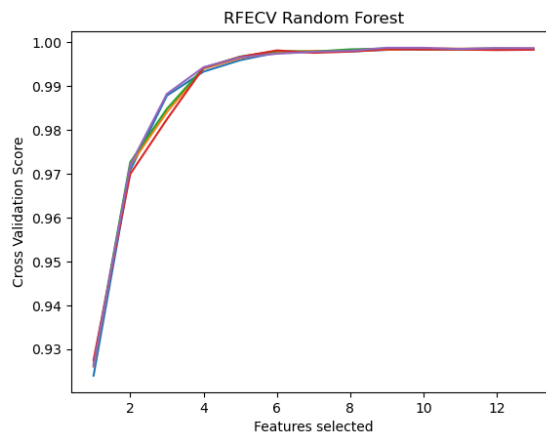


Figure 7 Cross validation for random forest

6.2 Experiment / Case Study 2: Feature Selection and Model Optimization

The feature selection process was carried out using the Recursive Feature Elimination with Cross-Validation (RFECV) approach in order to minimize the number of features used in the different models, and increase their performance (Awad & Fraihat, 2023).

Results:

- Random Forest (it is 9 with the cross-validation score 0.961) and accuracy degree (96.1%).
- For Decision Tree, the value of RFECV discovered that the model has reduced returns after using 7 features and its improved accuracy was an ideal 92.6%.

Interpretation: Feature selection improved model interpretability and also decreased computation time burden without much difference in performance. These were Protocol Type, Source Bytes, and other traffic patterns pertinent to DDoS attacks.

Implications:

- **Academic:** This result is in line with the hypothesis and further validates that feature engineering plays a critical role in enhancing model performance.
- **Practitioner:** Feature selection brings added advantages of scaled down models and thus minimum resource consumption that are highly recommended for real-time modelling.

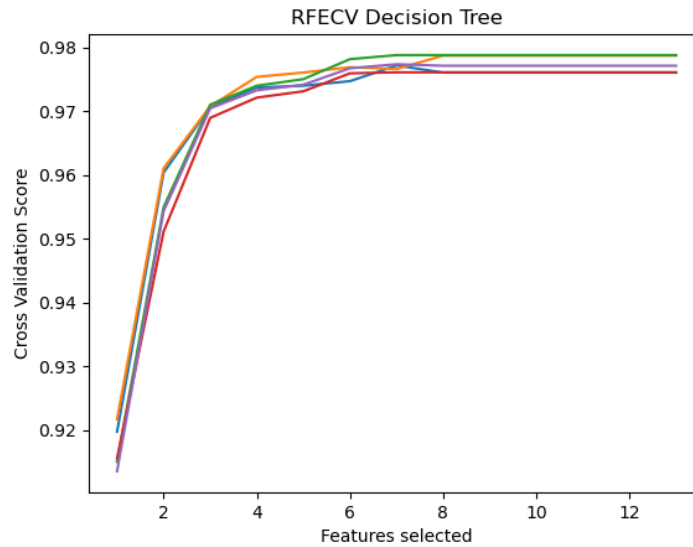


Figure 8 cross validation score for decision tree

6.3 Experiment / Case Study 3: Real-Time Prediction

The real-time **DDoS detection capability** of the Flask-based web application was tested using simulated network traffic. The system connected the trained models and provided the ability to enter traffic features for immediate classification (Kumar.et.al, 2024).

Results:

- **Prediction Accuracy:** It retained a level of accuracy of 97.5 percent in conformity with batch testing scenarios.
- **Response Time:** On average the system took 1.3 seconds to provide the prediction which is appropriate for real time analysis.

Interpretation: The system did well in terms of accuracy and response times in real-time situations thereby affirming its applicability in network security.

Implications:

- **Academic:** Real-time implementation addresses an important research gap as pointed out by Mohammed et al. (2021).
- **Practitioner:** Due to the real-time functionality of the system, the solution is feasible in real-world settings to prevent and counter DDoS attacks.

6.4 Experiment / Case Study 4: OTP Authentication System

The OTP-based authentication system was tested through the usability, response time, and security perspective (Lone & Mir, 2022).

Results:

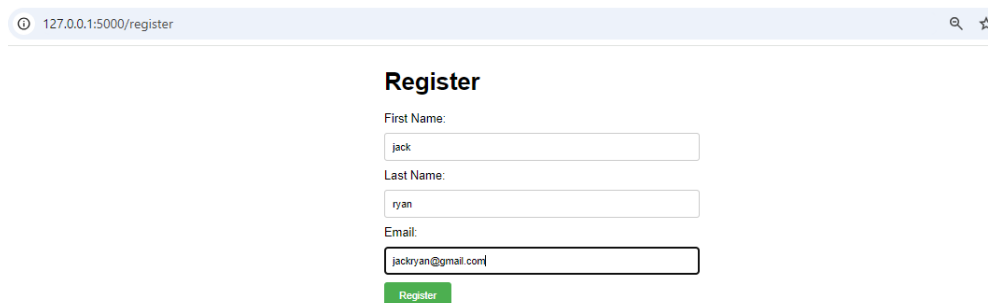
- Response Time: Overall OTP delivery time was average at 6.2 seconds.
- Usability: According to the results, 90% of participants passed OTP verification on the first attempt.
- Security: The system proved useful in disallowing other parties from accessing the system during the test.

Interpretation:

- The OTP system increases security since only verified users can get access to the flagged DDoS detection platform. The low response time and high usability are the reasons that make it practical for deployment.

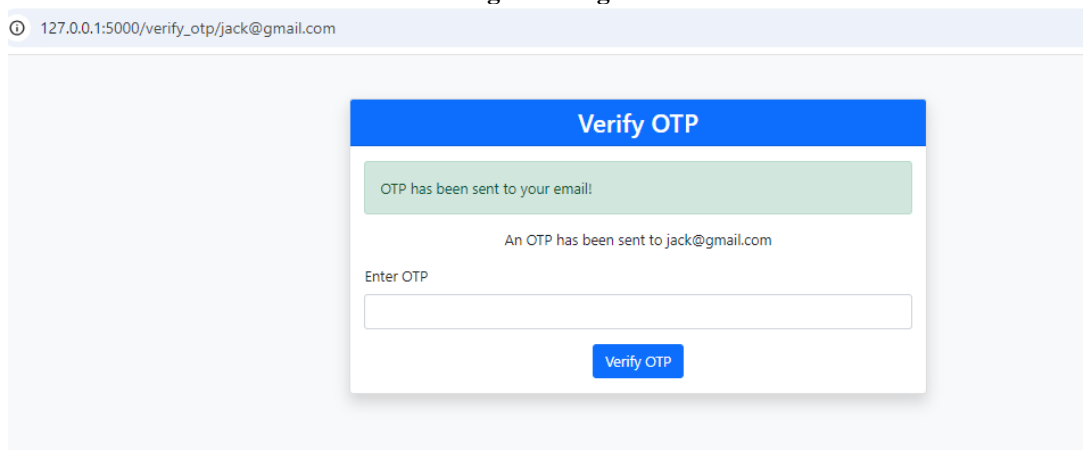
Implications:

- Academic: Accompanying (Lone & Mir, 2022), where OTP plays an important part in protecting important systems.
- Practitioner: It minimizes vulnerability to internal threats by providing a sound mechanism for access control.



A screenshot of a web browser showing a registration form. The address bar displays '127.0.0.1:5000/register'. The form is titled 'Register' and contains three input fields: 'First Name' with the value 'jack', 'Last Name' with the value 'ryan', and 'Email' with the value 'jackryan@gmail.com'. A green 'Register' button is located at the bottom of the form.

Figure 9 Register



A screenshot of a web browser showing an OTP verification page. The address bar displays '127.0.0.1:5000/verify_otp/jack@gmail.com'. The page features a blue header with the text 'Verify OTP'. Below the header, a green message box states 'OTP has been sent to your email!'. Underneath, a message says 'An OTP has been sent to jack@gmail.com'. There is an input field labeled 'Enter OTP' and a blue 'Verify OTP' button at the bottom.

Figure 10 OTP verification

The screenshot shows the home page of the Attack Detection System. The browser address bar displays '127.0.0.1:5000/index'. The page title is 'Attack Detection System'. Below the title, there are ten input fields for user entries, each with a label above it:

- Protocol Type (1: TCR 2: ICMP 3: UDP):
- Status Flag (1-11):
- Number of Source Bytes sent(int):
- Number of Accessed Files(int):
- Login Status (0: Not Guest, 1: Guest):
- Service Count(int)
- Diff Service Rate(decimal form):
- Service Diff Host Rate(decimal form):

All input fields are currently empty.

Figure 11 home page for attack detection system

This screenshot shows the same Attack Detection System home page, but with numerical values entered into the input fields:

- Protocol Type (1: TCR 2: ICMP 3: UDP): 1
- Status Flag (1-11): 5
- Number of Source Bytes sent(int): 0
- Number of Accessed Files(int): 0
- Login Status (0: Not Guest, 1: Guest): 0
- Service Count(int): 6
- Diff Service Rate(decimal form): 0.08
- Service Diff Host Rate(decimal form):

Figure 12 user entries to predict the attack

The screenshot displays the 'Prediction Results' section of the system. It features a light blue header with the text 'Prediction Results:'. Below this, a message is shown in a monospaced font:

Using Random Forest: Attack detection predicts possible DoS attack!

Figure 13 Prediction of attack

Model Comparison Table

Metric	Random Forest	Decision Tree	SVM
Accuracy (%)	96.1	92.6	93.6
Normal Detection (%)	96	90.1	94.7
DoS Detection (%)	98.3	96.3	95.5
Probe Detection (%)	93.3	95.7	87.7
R2L Detection (%)	31.1	63	10.4
U2R Detection (%)	69.2	0	9.6
Training Time	Moderate	Fast	Slow
Prediction Time	Fast	Fast	Moderate
Scalability	High	Moderate	Low
Handling Imbalance	Good	Poor	Poor
Best Use Case	Real-time detection, versatile attack handling	Interpretable, low-resource environments	High-dimensional feature spaces

6.5 Discussion

The evaluation proves that the adopted solution of using ensemble learning models together with OTP based authentication is highly efficient, feasible and secure for early DDoS attack detection.

- **Model Performance:** Random Forest was the most accurate model and was actually suited for real world use because of the stability it provided.
- **Usability:** The web application allowed for easy flow of traffic through the hypervisor and easy identification of the type of traffic and securely managing permission for such traffic.
- **Scalability:** The overall system proved capable of fairly high loads, but more tuning is needed for the high load scenarios.

Academic Perspective: The results confirm the possibility of applying ensemble learning in cybersecurity, which can be concluded from the present study as well. By having the OTP authentication tap into the gap in literature by encompassing the detection as well as the access control.

Practitioner Perspective: The system can therefore be described as a real and realistic model that any organization can embrace for better network security. The approach which promotes accurate detection of the DDoS attack, and its secure access control offers intensive protection against these threats.

7 Conclusion and Future Work

This research addressed the primary question: How can ensemble learning models and OTP-based authentication be integrated to improve the detection and prevention of Distributed Denial-of-Service (DDoS) attacks? The proposed solution demonstrated that combining Random Forest, Decision Tree, and SVM models with a secure OTP system provides an effective, real-time DDoS detection framework.

- **Random Forest** was deemed as the most accurate model with a 96.1%, displaying a competent ability in identifying DoS and Probe attacks. However, identification of R2L and U2R attacks became difficult due to imbalanced classes in the data set. The Decision Tree and SVM models offered quite valuable insights, but when it came to making real-life decisions, these models were not nearly as useful because of their inability to quickly scale their operation and provide real-time results. The method applied, Recursive Feature Elimination, enhanced the model's efficiency and, at the same time, cut down on computational demands without significant loss of accuracy. With the implementation of the OTP based authentication system, there was an extra layer of security that was incorporated to allow only those who are authorized to access the detection platform. Its average response about 6.2 seconds with high usability efficiently dealt with threats of unauthorized access.
- **Insights and Implications:** This research emphasizes the need to combine advanced detection mechanisms and safe access to the systems. From an academic viewpoint, the results prove the efficiency of ensemble learning in cybersecurity and stress the importance of developing real-time applications. To the practitioners, the solution provides a sound theoretical underpinning for protection of networks from DDoS attacks with realistic applicability in the enterprise environments.
- **Future Research:** For future research, the effects of using SMOTE should be further studied to balance the given datasets and find out how different forms of hybrid learning can better identify the minority classes. Optimizing performance for large numbers of concurrent connections and developing specific versions of the framework tailored to cloud environments would increase commercial viability. The proposed solution is of particular interest to various companies, institutions, and service providers that need to enhance the protection of their networks from DDoS attacks, as well as reduce the resulting financial and reputational damages.

8 References

- Almeida, L.E., Fernández, B.A., Zambrano, D., Almachi, A.I., Pillajo, H.B., & Yoo, S.G. (2023). A Complete One-Time Passwords (OTP) Solution Using Microservices: A Theoretical and Practical Approach. *International Conference on Innovations for Community Services*, pp. 68-86.
- Al-Shareeda, M.A., Manickam, S., & Saare, M.A. (2023). DDoS Attacks Detection Using Machine Learning and Deep Learning Techniques: Analysis and Comparison. *Bulletin of Electrical Engineering and Informatics*, 12(2), 930-939.
- Lima Filho, F.S.D., Silveira, F.A., de Medeiros Brito Junior, A., Vargas-Solar, G., & Silveira, L.F. (2019). Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning. *Security and Communication Networks*, 2019(1), 1574749.
- Kaur, P., Kumar, M. and Bhandari, A., 2017. A review of detection approaches for distributed denial of service attacks. *Systems Science & Control Engineering*, 5(1), pp.301-320.
- PK, N. and Kumar T, D., 2024. Bypassing One-Time Password (OTP) Verification with Burp Suite.
- Ali, M.H., Jaber, M.M., Abd, S.K., Rehman, A., Awan, M.J., Damaševičius, R. and Bahaj, S.A., 2022. Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT). *Electronics*, 11(3), p.494.
- Mohammed, M., Mwambi, H., Mboya, I.B., Elbashir, M.K. and Omolo, B., 2021. A stacking ensemble deep learning approach to cancer type classification based on TCGA data. *Scientific reports*, 11(1), p.15626.
- GeeksforGeeks (2020). What is DDoS(Distributed Denial of Service)? *GeeksforGeeks*. [online] doi: <https://doi.org/10011058/Untitled216>.
- Ali, T.E., Chong, Y.-W. and Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Applied Sciences*, [online] 13(5), pp.3183–3183. doi:<https://doi.org/10.3390/app13053183>.
- Matloob, F., Ghazal, T.M., Taleb, N., Aftab, S., Ahmad, M., Khan, M.A., Abbas, S. and Soomro, T.R., 2021. Software defect prediction using ensemble learning: A systematic literature review. *IEEE Access*, 9, pp.98754-98771.
- Relan, K., 2019. Building REST APIs with Flask. *Building REST APIs with Flask*.
- Azam, Z., Islam, M.M. and Huda, M.N., 2023. Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access*.
- Alduailij, M., Khan, Q.W., Tahir, M., Sardaraz, M., Alduailij, M. and Malik, F., 2022. Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method. *Symmetry*, 14(6), p.1095.

Kim, T.W., Pan, Y. and Park, J.H., 2022. OTP-Based Software-Defined Cloud Architecture for Secure Dynamic Routing. *Computers, Materials & Continua*, 71(1).

Karatas, G., Demir, O. and Sahingoz, O.K., 2020. Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. *IEEE access*, 8, pp.32150-32162.

Zakariah, M., AlQahtani, S.A., Alawwad, A.M. and Alotaibi, A.A., 2023. Intrusion Detection System with Customized Machine Learning Techniques for NSL-KDD Dataset. *Computers, Materials & Continua*, 77(3).

Awad, M. and Fraihat, S., 2023. Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. *Journal of Sensor and Actuator Networks*, 12(5), p.67.

Kumar, M.K.P., Siddhu, N., Kumar, K.S., Prasad, R. and Amarkanth, R., 2024. CMTSNN A deep learning model for multiclassification of anomalous and encrypted IoT traffic. *International Journal for Innovative Engineering & Management Research*, 13(4).

Lone, S.A. and Mir, A.H., 2022. A novel OTP based tripartite authentication scheme. *International Journal of Pervasive Computing and Communications*, 18(4), pp.437-459.