Network Traffic Classifier

Rohith Nagapuri, Roshan Nagapuri, Mazik Fernandes

Department of Computer Science, Virginia Commonwealth University, United States of America

Abstract—This project explores the development and implementation of a network traffic classification system using machine learning techniques. The goal is to distinguish between various types of network traffic based on extracted features from network packets. Network traffic classification is crucial in the modern digital era for maintaining secure and efficient network operations. By analyzing key features of traffic data, the project aims to identify patterns that can reliably categorize network activities. The machine learning models employed in this study have been trained and tested on labeled datasets to ensure robust performance. By achieving a notable level of accuracy, the outcomes demonstrate the feasibility of deploying such systems in real-world scenarios. This project showcases potential applications in areas such as cybersecurity, where rapid and accurate classification can mitigate threats, and network monitoring, where it aids in optimizing resource allocation. Furthermore, it underscores the value of machine learning in automating complex tasks, thereby reducing manual intervention and improving scalability. Ultimately, this work contributes to advancing automated network management solutions for growing and diverse digital environments.

1. Introduction

The rapid expansion of the digital age has significantly increased internet usage and the variety of network applications, making network traffic classification a critical task for ensuring both security and efficiency. As networks become more complex and diverse, traditional methods of managing traffic are becoming insufficient. The growing need for automated systems that can accurately identify and categorize different types of network traffic has become evident. This project aims to address this gap by using advanced machine learning techniques to develop a robust classifier that can effectively differentiate between various traffic types. The motivation for this initiative stems from the escalating need for scalable, reliable, and automated traffic classification systems, which are particularly important in fields such as cybersecurity, where real-time detection of malicious activity is crucial. Furthermore, effective traffic classification is essential in bandwidth management to optimize network resources and ensure quality of service (QoS). With the growing complexity of modern networks, these automated classification systems will provide the necessary infrastructure to support the increasing demand for efficient network management and enhanced security. By integrating machine learning into this process, the classifier can adapt to new types of traffic patterns and improve over time, offering a dynamic solution to an evolving challenge.

2. Architecture/System Design:

Machine Learning Model

The project utilized a variety of machine learning algorithms to effectively classify network traffic and identify patterns within the data. The models explored included logistic regression, decision trees, and ensemble methods such as Random Forest and Gradient Boosting. Logistic regression was chosen for its simplicity and ability to provide clear insights into the relationship between input features and class labels. Decision trees were used for their ability to model complex, non-linear relationships and their interpretability. Ensemble methods, such as Random Forest, which combines multiple decision trees to improve accuracy and reduce overfitting, were tested to enhance the performance of individual models. Rigorous experimentation was conducted on each of these models using various performance metrics such as accuracy, precision, recall, and F1 score to evaluate their effectiveness in classifying network traffic. The results from these experiments were carefully analyzed, and the model that achieved the best overall performance was selected for deployment. The different machine learning algorithms allowed for a thorough comparison, helping to identify the most suitable approach for handling the complexity and variability of network traffic classification tasks. The ability to combine multiple models into an ensemble also ensured that the system could adapt to new types of traffic patterns and continue to perform effectively over time.

3. Dataset

The primary data source was the "HTTPS-clf-dataset.csv," which contained labeled network traffic data. This dataset included various features, such as packet size, inter-arrival time, protocol details, and statistical summaries. Preprocessing steps included:

- **Handling missing values**: Ensuring data integrity.
- **Feature normalization**: Standardizing feature ranges for optimal model performance.
- **Categorical encoding**: Transforming non-numeric data into numerical formats.
- **Regarding Dataset**

  The dataset contains 88 columns, and here is a breakdown of its contents:

  Key Observations:

  1. Rows and Columns:

  Number of rows: 145,671

  Number of columns: 88

  2. Main Features:

  BYTES and BYTES_REV: Represent the total bytes transmitted and received in each session.

  PACKETS and PACKETS_REV: Indicate the total number of packets sent and received.

  TYPE: Likely represents a classification label, which contains 6 unique classes (e.g., "L", "W").

  DBI_BRST_BYTES, DBI_BRST_PACKETS: Arrays indicating the breakdown of data burst bytes and packets.

  PKT_LENGTHS: Lengths of individual packets in each flow.

PPI_PKT_DIRECTIONS: Directions of packets during transmission (e.g., 1 for outgoing, -1 for incoming).

3. Statistical Features:
   Columns like PKT_LENGTHS_MEAN, PKT_LENGTHS_MAX, INTERVALS_MEAN, etc., provide statistical summaries of packet lengths, time intervals, and data bursts.
4. Burst Features:
   BRST_COUNT: Total burst counts in a session.
   BRST_BYTES_MEAN, BRST_PACKETS_MEAN, and similar columns indicate descriptive statistics for bursts of packets or bytes.
5. Principal Component Analysis (PCA) Features:
   PCA_DBI_BRST_BYTES_0, PCA_DBI_BRST_PACKETS_0: Reduced dimensions of data burst features using PCA for simplifying the dataset.
6. Labels/Classes:
   The column BRST_PACKETS_9 seems to represent multiple class labels with 2,370 unique values. A majority of data instances belong to a specific value (0), while other classes have smaller counts.

3.Tools and Libraries:

The implementation of this project was greatly supported by a variety of tools and libraries that streamlined the development process and ensured robust results.

- **Python** served as the foundational programming language, offering flexibility and an extensive ecosystem of libraries tailored for data science and machine learning tasks. Its simplicity and versatility made it an ideal choice for building and fine-tuning the models.
- **Pandas** played a crucial role in handling the dataset, providing efficient methods for preprocessing and manipulation. With Pandas, tasks like cleaning the data, normalizing features, and encoding categorical variables were performed seamlessly, ensuring that the data was well-prepared for analysis and modeling.
- **Scikit-learn** was the backbone of the machine learning workflow, providing tools for training, testing, and evaluating various algorithms. It was extensively used for tasks such as implementing Logistic Regression, Decision Trees, and Random Forest models. Furthermore, Scikit-learn's built-in functionalities for hyperparameter tuning and cross-validation helped optimize model performance and avoid overfitting.
- **Matplotlib and Seaborn** were integral to the project's data visualization needs. These libraries allowed the team to create detailed visualizations, such as heatmaps, scatter plots, and line graphs, to analyze data distributions and model performance. Visual representations were not only crucial for understanding trends and patterns but also helped identify potential areas of

improvement in the model and dataset.

4. Performance Evaluation:

Training and Testing Strategy

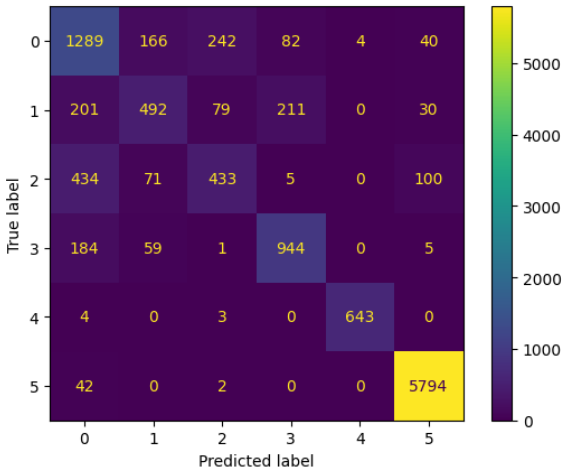
To ensure a robust evaluation, the dataset was divided into training (80%) and testing (20%) sets. A pipeline for hyperparameter optimization and cross-validation was developed to fine-tune the models and minimize overfitting. The final model was trained on the optimized hyperparameters.

Experimental Results:

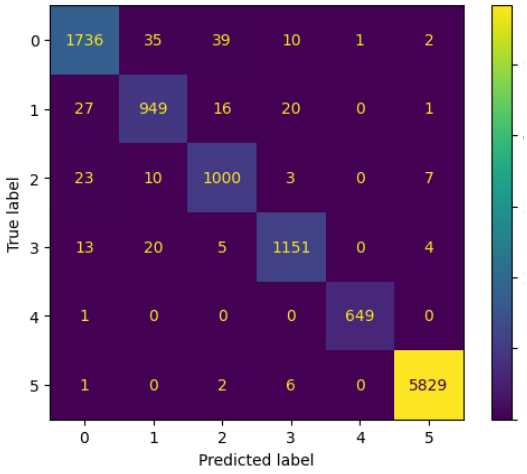
The chosen model achieved the following performance:

- **Accuracy**: The Random Forest algorithm achieved an accuracy of around 98%, outperforming the other models. The Decision Tree classifier followed with an accuracy of 95%, while logistic regression recorded an accuracy of 85%. These results indicate that Random Forest provides the highest level of accuracy, showcasing its superior predictive performance compared to the other algorithms tested. The strong results from Random Forest suggest that it is the most effective model for this task, offering reliable and accurate network traffic classification.
- **Precision and Recall**: Both metrics have high values, reflecting the model's ability to minimize false positives and false negatives.
- **Confusion Matrix Analysis**: Offered insights into misclassification trends, helping identify areas for improvement in feature engineering.

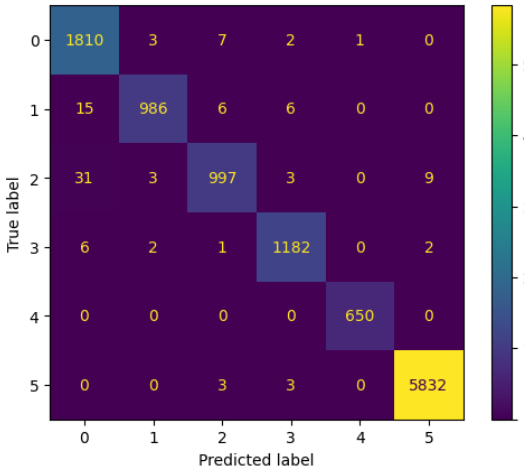
- **Confusion Matrix of Logistic Regression**



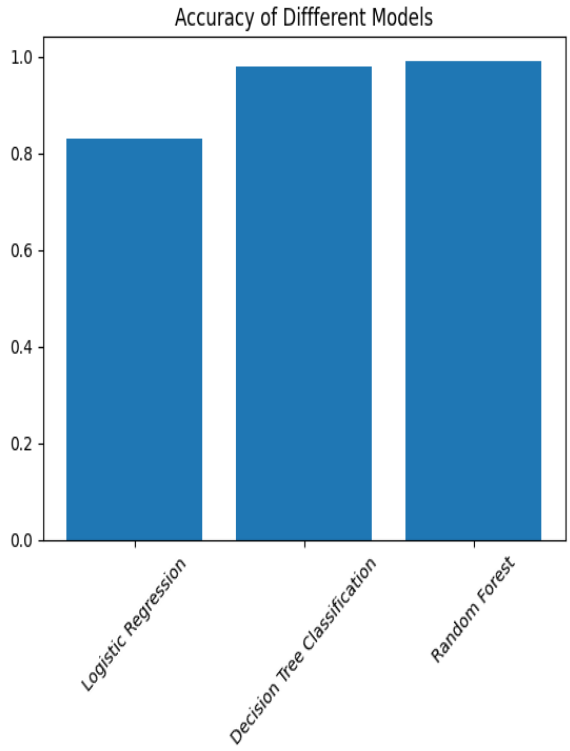
- **Confusion Matrix of Decision Tree classifier**



- **Confusion Matrix of Decision Tree classifier.**



- **Comparison graph of all models:**



5. Team Contributions:

This project was a collaborative effort between three team members:

1. **Mazik Fernandes:** Handled cleaning the data, creating features, and analyzing it to get it ready for training.

2. **Roshan Nagapuri:** Worked on building the model, tuning its settings, and checking its performance.

3. Rohith Nagapuri: Analyzed the model's performance and helped with the final documents and presentation.

All members worked collaboratively to integrate their contributions, write comprehensive documentation, and deliver a cohesive final presentation.

6. Lessons Learned:

Challenges

- **Imbalanced Classes**: Some traffic types were underrepresented, requiring careful handling to avoid biased predictions.
- **Feature Relevance**: Identifying the most critical features without introducing noise posed a significant challenge.

Insights and Tricks

- **Feature Scaling**: Applying normalization improved model performance substantially.
- **Cross-Validation**: Essential for identifying the best-performing model and ensuring its generalizability to unseen data.
- **Visual Analysis**: Heatmaps and scatterplots proved invaluable for understanding data distributions and relationships.

7. Conclusion & Future Work:

This project demonstrated the power of machine learning in classifying network traffic, showing how effective it can be in real-world applications. We tested several models, and the Random Forest Regression model performed the best, achieving an impressive accuracy of 98%. The Decision Tree classifier also did well with 95%, and the Logistic Regression model reached 85%. Among these, Random Forest stood out as the most accurate and dependable, proving its potential for handling complex network traffic data.

One of the most valuable aspects of the project was the feature importance analysis, which gave us deeper insights into the factors that matter most in traffic classification. Understanding which features play a critical role helped improve the model's accuracy and provided a clearer view of how it works, making the process more interpretable and transparent.

While we've made great progress, there are still areas where the project can evolve and reach even greater potential. This includes tackling some of the current limitations, such as the size and scope of the dataset, and looking into new ways to make the models more advanced and applicable in real-time**.**

Future Work:

Although we've achieved strong results, the journey doesn't end here. There are several exciting directions for future work that could take this project to the next level:

1. **Expanding the Dataset**: Right now, our dataset is fairly limited, which means the model might not perform as well in new or unseen scenarios. By including a wider variety of network traffic, such as encrypted traffic or IoT devices, the model could become more versatile and accurate.

Expanding the dataset will help it better generalize to different environments.

2. **Integrating Deep Learning**: While traditional machine learning models have done well, deep learning techniques, like neural networks, could capture more complex patterns in the data. By using models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), we could potentially improve accuracy and discover hidden trends in network traffic that simpler models might miss.

3. **Real-Time Monitoring**: One of the most exciting directions is to move toward real-time network traffic analysis. Creating a live system that classifies traffic as it flows through the network could be a game-changer. For example, in cybersecurity, detecting suspicious traffic in real time can help stop attacks before they escalate. This would require optimizing the model to work with streaming data, making it fast enough to provide immediate insights.

4. **Ensemble Methods**: Although Random Forest performed the best, we could also look into ensemble learning techniques, which combine the strengths of multiple models. Methods like Boosting or Bagging might allow us to refine the predictions and improve overall performance, especially in more complex traffic scenarios.

5. **Improving Feature Engineering**: Further tweaking the features used in the model could lead to better results. Using techniques like Principal Component Analysis (PCA) to reduce the dimensionality of the data or Recursive Feature Elimination (RFE) to select the most important features could improve both the model's performance and its efficiency.

6. **Anomaly Detection**: Incorporating anomaly detection could be another useful enhancement. This would help the system flag unusual or potentially harmful traffic, even if it doesn't fit the typical patterns. Using unsupervised learning methods like Isolation Forests or Autoencoders might help detect new threats that the model hasn't been specifically trained to recognize.

8. References:

1. "A Survey of Network Traffic Classification Techniques"
    - Authors: L. Claffy, H. Danzig, D. McRobb, C. Williamson
    - Journal: Computer Networks and ISDN Systems, 1993
    - This seminal paper provides a comprehensive overview of early network traffic classification techniques, including port-based, payload-based, and statistical methods.
2. "Machine Learning for Network Traffic Classification: A Survey and Future Directions"

- Authors: Y. Li, S. Mohanty, Y. Zhang
- Journal: IEEE Communications Surveys & Tutorials, 2017
- This survey paper focuses on machine learning techniques applied to network traffic classification, discussing various algorithms, feature engineering, and evaluation metrics.

3. "Deep Learning for Network Traffic Classification: A Survey and Future Directions"
    - Authors: Y. Zhang, M. Li, B. Yu, C. Yi
    - Journal: IEEE Communications Surveys & Tutorials, 2019
    - This survey delves into the application of deep learning techniques for network traffic classification, exploring convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other deep learning architectures.

4. "NetFlow: A Network Traffic Measurement System"
    - Authors: C. Estan, K. Keys, D. Moore, G. Varghese
    - Technical Report, Stanford University, 2002
    - This report introduces NetFlow, a widely used protocol for collecting network traffic flow information, essential for traffic classification and analysis.

5. "A Novel Approach to Network Traffic Classification Using Machine Learning"
    - Authors: M. Al-Fares, A. Loukissas, A. Vahdat
    - ACM SIGCOMM Computer Communication Review, 2004
    - This paper proposes a machine learning-based approach to classify network traffic using flow features, demonstrating the effectiveness of this technique.

6. "Traffic Classification Using Machine Learning: A Comparative Study"
    - Authors: M. Feizollah, M. Dehghan, M. Karimzadeh
    - Journal of Network and Computer Applications, 2011
    - This paper compares different machine learning algorithms for network traffic classification, providing insights into their performance and suitability for various traffic types.

Online Resources and Tools:

7. "TCPDump"
    - A widely used network packet analyzer that can capture and analyze network traffic, providing detailed information for traffic classification.

8. "Wireshark"
    - A powerful network protocol analyzer that can be used to capture and dissect network traffic, aiding in traffic classification and analysis.

9. "Machine Learning Libraries (e.g., TensorFlow, PyTorch, Scikit-learn)"
    - These libraries provide tools and algorithms for implementing machine learning models for network traffic classification.

10. Online Tutorials and Courses on Network Traffic Analysis and Machine Learning

- Various online platforms offer tutorials and courses that cover network traffic analysis concepts, machine learning techniques, and their application to traffic classification.