CSCI434/534 COLL 400 Capstone Sample Project

AAA BBB
xxx@wm.edu
William & Mary
Williamsburg, Virginia, USA

ABSTRACT

This project addresses the task of monitoring and classifying network traffic by its originating website, with the goal of enhancing network management, security, and performance optimization. We collected a dataset of traffic records from five popular websites: GitHub, Instagram, LinkedIn, Reddit, and Weather, and employed three machine learning models for classification: Linear Regression, Logistic Regression, and Random Forest. Comprehensive evaluation showed that the Random Forest model achieved the highest accuracy (0.91) and Macro F1 score (0.89), demonstrating its ability to handle complex relationships within the dataset. The project contributes to the field of network traffic classification by demonstrating the feasibility of integrating machine learning models into network traffic analysis, highlighting the importance of dataset diversity, and providing a road-map for future developments.

KEYWORDS

Network Traffic Classification, Machine Learning

1 INTRODUCTION

Monitoring and classifying network traffic are critical tasks for maintaining efficient network management, security, and performance optimization [7]. The proliferation of diverse online services necessitates robust methods to categorize traffic patterns and accurately identify their sources [2]. This project aims to develop a machine learning model to classify network traffic by its originating website, leveraging the diversity of traffic patterns from different online services.

Accurate classification of network traffic patterns has significant real-world implications [4]. It facilitates network management by identifying traffic sources and patterns, helps security by detecting unusual behaviors, and optimizes performance by ensuring balanced traffic flows. These capabilities are essential for managing and protecting modern networks.

We collected a dataset of traffic records from five popular websites: GitHub, Instagram, LinkedIn, Reddit, and Weather. This dataset was split into training, validation, and test sets in a 6:2:2 ratio. We employed three models for classification: Linear Regression [6], Logistic Regression [5], and Random Forest [1].

Comprehensive evaluation showed that the Random Forest model achieved the highest accuracy (0.91) and Macro F1 score (0.89), outperforming the other models. This demonstrates its ability to handle complex relationships within the dataset, facilitating accurate classification.

The project's contribution lies in its integration of machine learning models into network traffic analysis, demonstrating the feasibility and effectiveness of this approach in classifying traffic patterns by their originating website.

In summary, this project contributes to the field of network traffic classification, offering practical applications for network management, security, and performance optimization.

2 PROPOSED METHOD

In this section, we present our methodology for monitoring, analyzing, and classifying computer network traffic based on its originating website.

2.1 Data Preparation

To effectively monitor and analyze computer network traffic and classify it by website, we follow a systematic data preparation process as shown in Fig 1.

2.1.1 Tool Selection. In this project, we use Wireshark [3] (as shown in Fig 2), a highly regarded and versatile network protocol analyzer, to collect and analyze network traffic data. Wireshark offers a robust suite of features that allow us to capture, decode, and inspect network traffic at the packet level. Firstly, Wireshark provides detailed information about individual packets, which includes information on protocols used, packet size, and content. This granular level of detail is crucial for our classification tasks, allowing us to discern specific characteristics and trends within network traffic. Secondly, the tool supports a wide variety of protocols and formats, making it an ideal choice for handling diverse network environments. This flexibility ensures we can handle various types of traffic effectively. Thirdly, Wireshark's graphical user interface offers an intuitive way to explore and interpret network traffic data. The interface provides features such as real-time packet capture, filtering options, and detailed views of packet content, streamlining our analysis process.

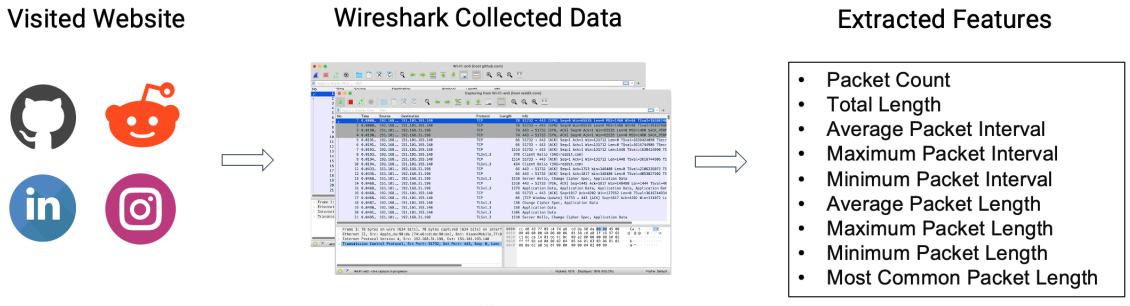

Figure 1: The pipeline of data preparation.

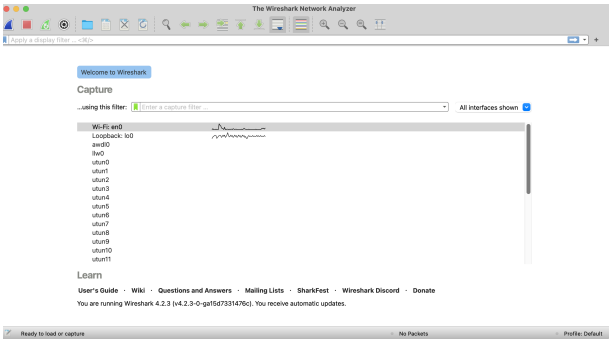

Figure 2: The interface of Wireshark, demonstrating its comprehensive design.

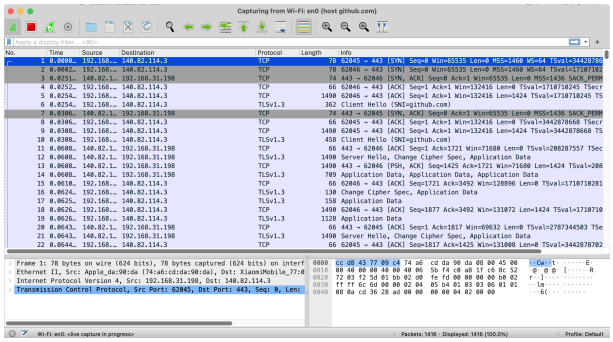

Figure 3: Sample of a data record, including 1423 packets with Time, Source, Destination, Protocol, and Length.

2.1.2 Data Collection. We initiate the data collection process by opening Wireshark, a network protocol analyzer, and configuring filters to capture traffic from specific websites: GitHub, Instagram, LinkedIn, Reddit, and Weather. This diverse selection of sites generates unique traffic patterns representative of popular online services. We visit each website, navigating its pages for a minimum of one minute to gather traffic data. This browsing session results in a single data record containing all packets captured during that time. We repeat this process for each of the five websites, creating a comprehensive raw dataset. The variety of captured packets ensures a representative sample of each website's traffic. The data record, as illustrated in Figure 3, comprises 1423 packets, with details including Time, Source, Destination, Protocol, and Length, providing a detailed snapshot of the network traffic.

2.1.3 Feature Engineering. After collecting a sufficient number of data records, each consisting of multiple packets, we proceed to feature engineering. The raw data cannot be directly used for training, thus we extract nine key features from each record, focusing on the dimensions of Time and Length (as shown in Fig 4):

- **Packet Count:** This feature indicates the total number of packets in each record, reflecting the overall communication volume. It's a key indicator of the traffic intensity and can reveal insights into the website's usage patterns.
- **Total Length:** This feature represents the cumulative size of all packets in a record, encompassing both headers and payloads. It provides a measure of the data throughput and is crucial for understanding the traffic load and capacity requirements.
- **Average Packet Interval:** This feature calculates the mean time difference between consecutive packets, offering insights into the temporal distribution of traffic. A higher average interval might indicate sporadic traffic, while a lower interval suggests a steady flow.
- **Maximum Packet Interval:** This feature captures the longest time gap between two packets in a record, helping identify periods of inactivity or delays. This can reflect

Packet_Count	Total_Length	Average_Packet_Interval	Maximum_Packet_Interval	Minimum_Packet_Interval	Average_Packet_Length	Maximum_Packet_Length	Minimum_Packet_Length	Most_Common_Packet_Length	Label
101	70900	0.001849129	0.042969	0	701.980198	1514	54	1514	weather
56	51233	0.000313482	0.002943	0	914.875	1514	66	1514	weather
84	74910	0.000421786	0.017585	0	891.7857143	1514	66	1514	weather
153	190545	0.000639222	0.047127	0	1245.392157	1514	66	1514	weather
102	110266	0.00388449	0.145289	0	1081.039216	1514	66	1514	weather
65	95229	5.60E-05	0.001129	0	1465.061538	1514	66	1514	weather
59	77544	0.000539983	0.02796	0	1314.305085	1514	66	1514	weather
168	236569	0.000127006	0.010861	0	1408.14881	1514	66	1514	weather
184	172453	0.00482169	0.214216	0	937.2445652	1514	66	1514	weather
200	229299	0.000417265	0.037358	0	1146.495	1514	66	1514	weather
74	103514	0.000551419	0.019868	0	1398.837838	1514	66	1514	weather
151	131159	0.005243106	0.088537	0	868.602649	1514	66	1514	weather
48	22158	0.121455771	4.940232	0	461.625	1514	54	66	weather
168	158586	0.034517405	2.942865	0	943.9642857	1490	66	1490	github
175	172491	0.03063932	3.162229	0	985.6628571	1490	66	1490	github
58	67712	6.86E-05	0.001349	0	1167.448276	1490	66	1490	github
109	90037	0.076340367	7.674164	0	826.0275229	1490	66	1490	github
101	118506	0.01842604	0.959542	0	1173.326733	1490	66	1490	github
124	134854	0.029503105	0.970273	0	1087.532258	1490	66	1490	github
84	99274	0.006238798	0.446346	0	1181.833333	1490	66	1490	github
86	105150	0.044612233	3.225474	0	1222.674419	1490	66	1490	github
94	114664	0.005806277	0.270699	0	1219.829787	1490	66	1490	github

Figure 4: The samples of processed data.

the responsiveness of the website or the nature of its traffic.

- **Minimum Packet Interval:** This feature denotes the shortest time gap between two packets, indicating the speed at which data is transferred during active periods. It reflects the network's latency and the website's responsiveness.
- **Average Packet Length:** This feature calculates the mean size of all packets in a record, offering insights into the average data volume transferred per packet. It can indicate the nature of the website's content (e.g., text-heavy vs. media-heavy).
- **Maximum Packet Length:** This feature indicates the size of the largest packet in a record, reflecting the maximum data transfer capacity in a single packet. It can reveal the presence of large files or media content.
- **Minimum Packet Length:** This feature captures the size of the smallest packet in a record, reflecting minimal data transfer requirements. It can indicate small, frequent communications or requests.
- **Most Common Packet Length:** This feature represents the size that appears most frequently among the packets, revealing the typical packet size in the record. This can provide insights into the website's primary data types and content nature.

These features provide valuable insights into the nature of the traffic and aid in the subsequent model training phase, allowing for the automatic classification of network traffic by its originating website.

2.2 Employed Method

For our project on monitoring and analyzing network traffic, particularly from popular internet websites, and classifying the originating website, we employ the Random Forest algorithm [1]. This algorithm is integral to our solution due to its flexibility, robustness, and strong performance in handling complex classification tasks.

2.2.1 Random Forest. Random Forest is an ensemble learning method primarily used for classification and regression tasks. It operates by constructing multiple decision trees during the training phase and aggregating their outputs to make a final decision.

The process of Random Forest (as shown in Fig 5) is:

- Building the Forest: During training, multiple decision trees are generated. Each tree is built from a bootstrapped sample of the training dataset. In each tree, a subset of features is randomly chosen to split at each node, which helps introduce diversity among the trees.
- Voting Mechanism: In classification tasks, each tree votes on the class label for a given input. The class that receives the most votes from the trees in the forest becomes the final prediction.
- Robustness and Stability: This ensemble approach makes Random Forest resistant to overfitting, a common issue in single decision trees, and provides a stable and reliable model even in the face of noisy data or variations in feature sets.

The benefits of Random Forest includes:

- Handling Complex Features: The task of classifying network traffic by its originating website involves handling diverse and complex features, such as protocols and packet lengths. Random Forest's decision trees can efficiently capture interactions between these features, producing accurate classifications.
- Generalization: Due to its ensemble nature, Random Forest generalizes well to unseen data. This makes it highly suitable for our application, where network traffic patterns can vary significantly between different websites and even for the same website over time.
- Interpretability: While individual decision trees may be challenging to interpret, Random Forest offers useful insights into feature importance. This capability is valuable for understanding which features, such as protocol type

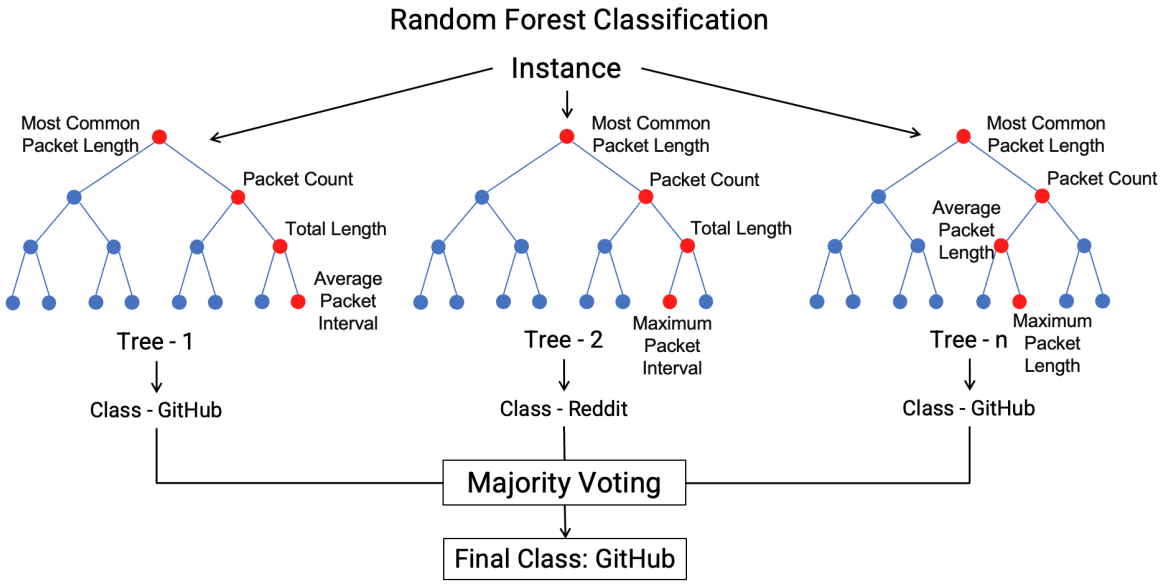

Figure 5: The model structure of Random Forest.

or packet length, contribute most to distinguishing traffic from different websites.

- Scalability: The Random Forest algorithm is highly scalable and can handle large datasets. This is particularly important for our project, as network traffic data can accumulate rapidly, necessitating a model that can efficiently handle large volumes of data.

2.2.2 Baselines. We compared two baselines:

- **Linear Regression** [6] serves as a baseline model for continuous numerical predictions due to its simplicity, interpret-ability, and straightforward implementation. It provides a linear relationship between the dependent variable and one or more independent variables, making it easy to understand and communicate results. The Ordinary Least Squares estimation minimizes the sum of squared errors, offering a quick way to generate initial predictions. Its assumptions, though restrictive, provide a useful framework for initial model evaluation, making it a reliable starting point for comparing more complex models in regression tasks.
- **Logistic Regression** [5] is chosen as a baseline model for binary and categorical classification tasks due to its simplicity, interpretability, and effectiveness in probability-based predictions. Its logistic function directly maps features to probabilities, offering clear insight into the likelihood of a particular outcome. Maximum Likelihood Estimation provides a robust way to fit the model to data, and its assumptions lay the groundwork for initial model validation. This makes it a valuable baseline for comparing more complex models in classification tasks.

3 EVALUATION

In this section, we perform an in-depth evaluation of our models and their performance on the dataset.

3.1 Dataset

After processing the collected data, we obtained the dataset shown in Table 1. The train, validation, and test sets were split with a ratio of 6:2:2.

Table 1: Details of Dataset

Label	Size of Processed Data
GitHub	24
Instagram	10
LinkedIn	18
Reddit	9
Weather	13
Total	74

3.2 Evaluation Metrics

We use two primary metrics to evaluate the models: Accuracy and Macro F1 score. These metrics provide a comprehensive view of performance across all classes.

The **Accuracy** is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

The **Precision** is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision indicates how many of the predicted positive samples are actually positive.

The **Recall** is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Recall reflects how many of the actual positive samples are correctly identified as positive.

The **Macro F1 Score** is calculated by taking the harmonic mean of precision and recall for each class, and then averaging these scores across all classes:

$$\text{Macro F1} = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

3.3 Evaluation Results

The results in Table 2 show that the Random Forest model achieves the highest accuracy (0.91) and Macro F1 score (0.89), outperforming both Linear and Logistic Regression models. This indicates that the Random Forest model is more effective at capturing complex relationships in the data.

Table 2: The comparison result

Model	Accuracy	Macro F1 Score
Linear Regression	0.39	0.28
Logistic Regression	0.83	0.84
Random Forest	**0.91**	**0.89**

3.4 Visualization Results

To evaluate how each model's predictions align with actual values, we visualize the predictions and real values for each model: Linear Regression (Fig 6), Logistic Regression (Fig 7), and Random Forest (Fig 8).

The results show that the Random Forest model exhibits the closest alignment between predictions and actual values, reinforcing its superior performance observed in the quantitative metrics.

4 DISCUSSION & FUTURE WORK

4.1 Interpretation of Results

Three models were tested: Linear Regression, Logistic Regression, and Random Forest. The results indicate that Random Forest outperformed both Linear and Logistic Regression in terms of both Accuracy and Macro F1 score. Specifically, Random Forest achieved an accuracy of 0.91 and a Macro F1 score of 0.89, compared to Logistic Regression's accuracy of 0.83 and Macro F1 score of 0.84. This performance gap highlights Random Forest's ability to capture complex relationships in the data, making it particularly suitable for network traffic classification, where patterns can vary significantly.

The visualizations further reinforce the quantitative findings. The figures show that the Random Forest model's predictions align closely with the actual values, providing a visual affirmation of its superior performance.

The evaluation and comparison of these models contribute significantly to the problem at hand: monitoring and classifying network traffic. The high accuracy and precision of the Random Forest model allow for effective classification of diverse traffic patterns, supporting efficient network management and optimization. This is particularly relevant for identifying unusual or potentially harmful traffic patterns, which is crucial for network security.

4.2 Future Work

Despite the strong performance of Random Forest, the project faced some challenges:

- **Dataset Size:** The relatively small dataset size might limit the model's ability to generalize to broader traffic patterns. This may also impact the model's performance when encountering new traffic types or variations in existing ones.
- **Handling Real-Time Data:** The current methodology captures data from static browsing sessions, but real-world network environments often involve rapidly accumulating and dynamic traffic. This necessitates handling real-time data streams effectively.
- **Model Complexity:** While the Random Forest model performs well, its ensemble nature and depth can introduce interpretability challenges, making it harder to understand which specific features drive its decisions.

To address these challenges and improve the project further, we propose the following directions:

- **Dataset Expansion:** Expanding the dataset to include more websites and traffic records, as well as capturing traffic patterns over longer durations, will enhance the model's training, making it more generalizable and robust.

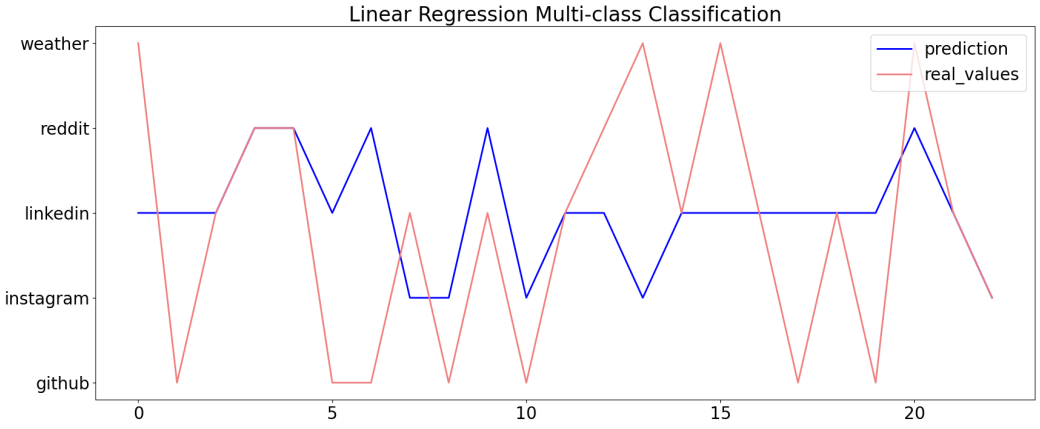

Figure 6: Predictions vs. actual values for Linear Regression.

Figure 7: Predictions vs. actual values for Logistic Regression.

- **Advanced Models:** Exploring other machine learning models, such as deep learning or ensemble approaches, could further improve classification accuracy by capturing more complex traffic relationships.
- **Real-Time Processing:** Enhancing the model to handle real-time data streams effectively will improve its practical utility in dynamic network environments. This could involve integration with streaming analytics platforms or developing custom solutions for real-time traffic capture and analysis.
- **Practical Applications:** Future research should also focus on practical applications, integrating the model into real-world systems for network monitoring, management, and security, demonstrating its value beyond academic contexts. This could involve collaborating with industry partners to refine and implement the model in diverse environments.

5 CONCLUSION

In this project, we developed and evaluated a model for monitoring and classifying network traffic by its originating website. Using a dataset collected from diverse websites, we employed three models: Linear Regression, Logistic Regression, and Random Forest. Through comprehensive evaluation, the Random Forest model was found to outperform the others, achieving the highest accuracy and Macro F1 score.

This work demonstrates the feasibility of machine learning models, particularly Random Forest, for handling complex relationships in network traffic data. Accurate classification of network traffic patterns contributes significantly to efficient

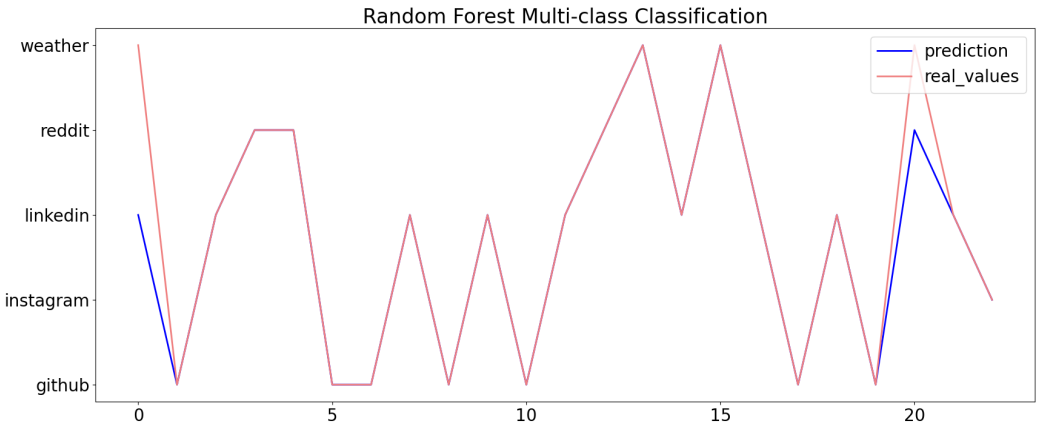

Figure 8: Predictions vs. actual values for Random Forest.

network management, security, and performance optimization.

The main insights include: 1) The Random Forest model's ensemble approach provides robustness and superior performance, particularly in handling diverse traffic patterns. 2) The project's results indicate the importance of dataset size and diversity in model training, highlighting the need for expansion in future work. 3) The project's visualizations reveal how each model's predictions align with actual values, providing further insights into their effectiveness.

This project serves as a foundation for future developments in network traffic analysis, with practical implications for real-world systems and applications. By integrating machine learning models into network monitoring, we can enhance performance, security, and management.

REFERENCES

[1] Gérard Biau and Erwan Scornet. 2016. A random forest guided tour. *Test* 25 (2016), 197–227.

[2] Arthur Callado, Carlos Kamienski, Géza Szabó, Balázs Péter Gero, Judith Kelner, Stênio Fernandes, and Djamel Sadok. 2009. A survey on internet traffic identification. *IEEE communications surveys & tutorials* 11, 3 (2009), 37–52.

[3] Wireshark Foundation. 2024. Wireshark. https://www.wireshark.org/.

[4] Hyunchul Kim, Kimberly C Claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, and KiYoung Lee. 2008. Internet traffic classification demystified: myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT conference*. 1–12.

[5] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. 2002. *Logistic regression*. Springer.

[6] George AF Seber and Alan J Lee. 2012. *Linear regression analysis*. John Wiley & Sons.

[7] Silvio Valenti, Dario Rossi, Alberto Dainotti, Antonio Pescapè, Alessandro Finamore, and Marco Mellia. 2013. Reviewing traffic classification. *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience* (2013), 123–147.