Efficient Network Traffic Management and Intelligent Decision-Making through Machine Learning and DNS Log Analysis

by

Syed Abed Hossain
21266019

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
August 2023

Declaration

It is hereby declared that

1. The thesis submitted is my our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Syed Abed Hossain

21266019

Approval

The thesis/project titled "Efficient Network Traffic Management and Intelligent Decision-Making through Machine Learning and DNS Log Analysis" submitted by

1. Syed Abed Hossain (21266019)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on August 23, 2023.

Examining Committee:

External Examiner:
(Member)

Dr. Mohammad Zahidur Rahman

Professor
Department of Computer Science and Engineering
Jahangirnagar University

Internal Examiner:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Internal Examiner:
(Member)

Muhammad Iqbal Hossain, PhD

Associate Professor
Department of Computer Science and Engineering
Brac University

Supervisor:
(Member)

Amitabha Chakrabarty, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

This research presents a comprehensive approach to network traffic management and analysis by leveraging DNS log analysis, machine learning techniques, and Software-Defined Networking (SDN) integration. In an office environment, a DNS server was set up to collect DNS logs from nearly 200 users over a month. The collected data was subjected to data cleaning and additional information extraction in Google BigQuery. Demographic analysis was conducted using Google LookerStudio, providing valuable insights into user behavior patterns during different office hours. Subsequently, various supervised and unsupervised machine learning models were employed to predict browsing categories based on the DNS log analysis. Among the models evaluated, the Random Forest Classifier (RFC) demonstrated exceptional performance, achieving high accuracy, precision, recall, and F1 Score during training, with values of 82.54%, 82.79%, 82.54%, and 81.81%, respectively. The trained RFC model showcased its robustness in minimizing the discrepancy between predicted probabilities and actual class values. The trained model was then exported and integrated into a virtual Linux machine to simulate an SDN environment. The experimental results showcased the system's high accuracy in categorizing DNS queries during real-time testing, with 100% accuracy achieved for categories like Ads and Entertainment, and impressive accuracy rates of 98.57%, 87.5%, and 87.21% for Search Engines, Social Networks, and CDNs, respectively. The system's reliability and effectiveness in intelligently managing network traffic were further demonstrated with slightly lower but still respectable accuracies of 81.82% and 80.95% for Computer/Technology and Learning categories, respectively. The predictive capabilities of the system have practical applications for office network management, including website blocking, traffic rerouting based on predictions, and bandwidth management, all facilitated through the SDN controller. The findings of this study highlight the efficacy of combining DNS log analysis, machine learning, and SDN integration for enhancing network security, optimizing resource allocation, and delivering an enhanced user experience in a standard office environment. The presented approach can serve as a blueprint for efficient network traffic management and intelligent decision-making in similar settings.

Keywords: DNS traffic management; Machine learning; SDN integration; Network security; Website categorization; DNS log analysis; Random Forest classification; Deep packet inspection; Dynamic feature selection; Virtual machine; Bind9 DNS software; User behavior analysis; Network traffic patterns.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our co-advisor Teacher Name sir for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, Name and the whole judging panel of Conference Name. Though our paper not accepted there, all the reviews they gave helped us a lot in our later works.
And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration i

Approval ii

Ethics Statement iv

Abstract iv

Dedication v

Acknowledgment v

Table of Contents vi

List of Figures ix

List of Tables xi

Nomenclature xi

1 Introduction 1
- 1.1 Problem statement . . . 2
- 1.2 Research Objectives . . . 2
- 1.3 Research Questions . . . 4

2 Literature Review 5
- 2.1 Review of relevant literature and prior research . . . 5
- 2.2 Research gaps . . . 8
- 2.3 Related Works . . . 12

3 Methodology 13
- 3.1 Research design and approach . . . 14
  - 3.1.1 Research Design . . . 14
  - 3.1.2 Variables and Measures . . . 14
  - 3.1.3 Exploratory Data Analysis . . . 15
  - 3.1.4 Ethical Considerations . . . 15
  - 3.1.5 Limitations . . . 15
- 3.2 Data Collection . . . 15
- 3.3 Overview of tools, technologies, and frameworks used . . . 16
  - 3.3.1 Virtualization Environment and Containerization . . . 16

3.3.2 DNS Log Collection and Forwarding . . . 16
3.3.3 Cloud Storage and Analysis . . . 16
3.3.4 Data Analysis and Visualization . . . 17
3.3.5 Data Manipulation . . . 17
3.3.6 Machine Learning . . . 17
3.3.7 Integration of Trained Model into SDN Environment . . . 22
3.3.8 Ethical Considerations . . . 23

4 Data Collection and Preprocessing 24
4.1 Description of the data collection process . . . 24
4.2 Explanation of data preprocessing techniques . . . 25
4.2.1 Data Integration and Aggregation . . . 26
4.2.2 Feature Extraction and Engineering . . . 28
4.2.3 Data Encoding and Transformation . . . 31
4.2.4 Data Splitting . . . 32

5 Analysis and Results 34
5.1 Visual Analysis and Insights . . . 34
5.1.1 Dataset Overview . . . 35
5.1.2 User Engagement Analysis Using Demographic Data . . . 35
5.1.3 User Browsing Analysis Using Demographic Data . . . 40
5.2 Analysis and Results of Machine Learning Models . . . 48
5.2.1 Supervised Learning Models . . . 48
5.2.2 Unsupervised Learning Models . . . 55

6 Discussion 63
6.1 Comparative Studies . . . 65
6.1.1 Focus and Scope . . . 65
6.1.2 Methodology . . . 66
6.1.3 Findings . . . 66
6.1.4 Contributions . . . 66

7 Proposed Solution 67
7.1 Detailed explanation of the proposed solution . . . 67
7.1.1 Data Collection and Model Training . . . 67
7.1.2 Model Export and Integration with SDN Controller . . . 67
7.1.3 DNS Packet Intercept and Feature Extraction . . . 67
7.1.4 Predicting Website Categories . . . 68
7.1.5 Network Management Tasks Based on Predictions . . . 68
7.1.6 Real-Time Decision-Making . . . 68
7.2 Explanation of the rationale behind the solution design . . . 68
7.2.1 Comprehensive Approach . . . 68
7.2.2 Integration with SDN Controllers . . . 69
7.2.3 Strengthening Firewall Systems . . . 70

8 Experimental Evaluation 71
8.1 Description of Experimental Setup . . . 71
8.2 Parameters . . . 72
8.3 Presentation of experimental results and analysis . . . 72

8.3.1 Prediction Accuracy 72
8.3.2 Real-Time Response 73
8.3.3 Scalability 74

9 Evaluation and Validation **75**
9.1 Validation of Results against Research Objectives 75
9.1.1 Objective 1: Comprehensive Approach for Network Traffic Management 75
9.1.2 Objective 2: Feasibility of Browsing Category Prediction and SDN Integration 75
9.1.3 Objective 3: Enhancing Firewall Capabilities for URL Identification and Policy Enforcement 76
9.2 Discussion of Limitations and Potential Improvements 76
9.2.1 Limitations 76
9.2.2 Potential Improvements 77

10 Conclusion **79**

Bibliography **85**

List of Figures

3.1 Research Framework 17

5.1 Gender Ratio By Age Group 36
5.2 Gender Engagement By Weekday 36
5.3 Age Group Engagement By Weekday 37
5.4 Age Group Engagement By Weekday Bar Chart 37
5.5 Gender Engagement By Office Hour of Each Weekday 38
5.6 Hourly Engagement Count By Office Hour 38
5.7 Hourly Engagement Count By Office Hour 39
5.8 Hourly Engagement Count By Office Hour 39
5.9 Hourly Engagement Count By Office Hour 40
5.10 Hourly Engagement Count By Office Hour 41
5.11 Domain Category By Gender Engagement 42
5.12 Domain Category By Age Group Engagement 43
5.13 Search Engine Engagement By Age Group And Gender 43
5.14 Social Media Engagement By Age Group And Gender 44
5.15 Youtube Engagement By Age Group And Gender 44
5.16 ChatGPT Engagement By Age Group And Gender 45
5.17 Online News Platform Engagement By Age Group And Gender . . . 46
5.18 CDN Engagement By Age Group And Gender 46
5.19 Torrent Engagement By Age Group And Gender 47
5.20 Feature Importance (Random Forest Classifier) 49
5.21 Performance (Random Forest Classifier) 49
5.22 Feature Importance (Naive Bayes Model) 49
5.23 Performance (Naive Bayes Model) 50
5.24 Feature Importance (SVM) 50
5.25 Performance (SVM) 51
5.26 Feature Importance (KNN) 51
5.27 Performance (KNN) 52
5.28 Feature Importance (Decision Tree) 52
5.29 Performance (Decision Tree) 53
5.30 Feature Importance (MLP) 53
5.31 Performance (MLP) 54
5.32 Feature Importance (AdaBoost) 54
5.33 Performance (AdaBoost) 55
5.34 Elbow Method 55
5.35 K-Means Clustering 56
5.36 K-Means Clusters Silhouette Score vs Number of Clusters 56

5.37 Silhouette Score For 3 and 4 Clusters 57
5.38 Silhouette Score For 6 and 7 Clusters 58
5.39 Silhouette Score For 9 and 10 Clusters 58
5.40 K-Means Clustering Inertia and Silhouette Score Comparison 59
5.41 DBSCAN Clustering 59
5.42 DBSCAN Performance Evaluation (Silhouette Score/Number of Clusters vs Minimum Samples) 60
5.43 Gaussian Mixture Model (GMM) Clustering 60
5.44 Gaussian Mixture Model (GMM) Silhouette Score vs Number of Clusters 61
5.45 Mean Shift Clustering 61
5.46 Mean Shift Clustering Performance Evaluation (Silhouette Score/Number of Clusters vs Quantile Number) 62

6.1 Log Loss Of Supervised Learning Models 63
6.2 Feature Importance Comparison Of Supervised Learning Models . . . 64
6.3 Performance Comparison Of Supervised Learning Models 64

7.1 Visual Representation Of Proposed Solution In SDN 69

8.1 Prediction Accuracy In Sumulation 72
8.2 Real time Response Sample 1 73
8.3 Real time Response Sample 2 73

List of Tables

2.1 Summery of Literature Review 1 9
2.2 Summery of Literature Review 2 10
2.3 Summery of Literature Review 3 11

5.1 DNS Type Count 35

Chapter 1

Introduction

In today's digital age, efficient network traffic management and analysis play a crucial role in ensuring optimal network performance and security. The increasing complexity of modern networks, coupled with the exponential growth of internet usage, necessitates innovative approaches to handle the ever-increasing volume of network traffic. The Domain Name System (DNS) [1] is a fundamental component of network infrastructure, responsible for translating human-readable domain names into IP addresses. DNS logs, which record the queries made by users and the corresponding responses, contain valuable information about user behavior, network usage patterns, and potential security threats.

Existing studies have explored the use of DNS logs for various purposes, such as anomaly detection, network monitoring, and user behavior analysis. However, there is still a need for comprehensive research that combines DNS log analysis with data manipulation, visualization, and machine learning [2] techniques to gain deeper insights into network traffic and enable proactive traffic management. By leveraging the power of machine learning algorithms, it is possible to predict browsing categories based on DNS log analysis, which can facilitate dynamic policy enforcement, traffic routing, and bandwidth management.

This thesis aims to address these gaps by presenting a novel approach to network traffic management and analysis. The research focuses on implementing a local DNS server, rerouting office network traffic to it, and collecting DNS logs for analysis. The collected DNS logs are then imported into Google BigQuery [3] for data modification and manipulation. Additional demographic information, such as age and gender, is incorporated into the dataset to provide a more comprehensive understanding of user behavior. Moreover, by deriving new features such as weekday and office hour from timestamps in the DNS logs, it becomes possible to analyze user engagement and browsing category preferences during different times of the day.

Furthermore, this research explores the visualization of the collected data using Google LookerStudio [4], providing in-depth analysis and data visualization of various aspects, including data overview, user engagement by office hour, and browsing category analysis by gender, age group, and office hour. The application of supervised and unsupervised machine learning algorithms to predict browsing categories based on DNS log analysis further enhances the capabilities of network traffic management. The resulting model, which achieves the highest accuracy, is exported and integrated into an SDN controller [5], enabling real-time prediction and allowing for the implementation of dynamic policies such as access control lists, traffic routing,

and bandwidth management.
By combining DNS log analysis, data visualization, and machine learning techniques, this research aims to contribute to the field of network traffic management and provide valuable insights for network administrators and policymakers. The findings of this study have the potential to improve network performance, enhance security, and optimize the utilization of network resources in various domains and industries.

1.1 Problem statement

In recent years, especially in the post-COVID [6] era, there has been a significant shift towards online operations and activities for corporate offices. This transition necessitates more dynamic and intelligent internet traffic management and manipulation to meet the ever-increasing demands and requirements of internet usage. Efficient network traffic management becomes crucial to ensure optimal performance and security in this evolving digital landscape.
Furthermore, the proliferation of URL shortening, modifications, and the use of multiple subdomains by popular websites to cater to global users pose challenges for existing firewall systems in accurately identifying and enforcing policies based on specific URLs. These adaptations and variations in domain names and URL structures hinder the effectiveness of traditional firewall systems.
Therefore, this thesis aims to address these interconnected challenges. The primary objective is to develop a comprehensive approach that combines dynamic internet traffic management, traffic manipulation techniques, and intelligent analysis of DNS logs. By leveraging data manipulation, visualization, and machine learning techniques, this research will enable more effective and proactive network traffic management in the context of corporate offices' online operations.
Additionally, the research will investigate methods to enhance URL [7] identification and policy enforcement capabilities in firewall systems. By studying the patterns, modifications, and variations in domain names and URL structures, innovative approaches will be developed to improve the accuracy and efficiency of URL filtering and policy enforcement, ensuring a more robust and adaptive security framework.
By addressing these challenges, this research seeks to contribute to the field of network traffic management by providing solutions to enhance internet traffic management, overcome URL identification issues, and improve security measures. The findings of this study will be valuable for network administrators, policymakers, and organizations looking to optimize network performance, ensure security compliance, and adapt to the dynamic online landscape in the post-COVID era.

1.2 Research Objectives

The primary objectives of this research are as follows:

1. Develop a comprehensive approach that combines DNS log analysis, data manipulation, visualization, and machine learning techniques for effective network traffic management.

2. Investigate the feasibility of predicting browsing categories based on DNS log analysis and integrate the resulting models into SDN controllers for real-time

traffic management.

3. Enhance URL identification and policy enforcement capabilities in firewall systems by studying patterns, modifications, and variations in domain names and URL structures.

To achieve these objectives, the research will encompass the following key areas:

1. Implementation of a local DNS server: A local DNS server will be implemented to reroute office network traffic, allowing for the collection of DNS logs for analysis.

2. Data modification and manipulation: The collected DNS logs will be imported into Google BigQuery for data modification and manipulation. Demographic information, such as age and gender, will be incorporated into the dataset for a more comprehensive understanding of user behavior.

3. Data visualization and analysis: Google LookerStudio will be utilized for data visualization, providing in-depth analysis of various aspects including data overview, user engagement by office hour, and browsing category analysis by gender, age group, and office hour.

4. Machine learning algorithms: Various supervised and unsupervised machine learning algorithms will be applied to predict browsing categories based on DNS log analysis. The resulting model with the highest accuracy will be exported and integrated into an SDN controller for real-time traffic management.

5. URL identification and policy enforcement enhancements: Innovative approaches will be developed to improve the accuracy and efficiency of URL filtering and policy enforcement in firewall systems, addressing challenges posed by URL shortening, modifications, and domain name variations.

The scope of this research will be focused on the implementation and analysis of a local DNS server within the context of corporate office networks. The study will utilize DNS logs collected over a specified period, and the analysis will primarily revolve around user behavior, network usage patterns, and browsing category prediction. The integration of the predictive model into an SDN controller will enable real-time traffic management based on browsing categories. Additionally, the research will explore enhancements to URL identification and policy enforcement capabilities in the context of firewall systems. However, it is important to note that this research will not cover broader aspects of network infrastructure and security beyond the scope outlined.

By accomplishing these research objectives and testing the hypotheses within the defined scope, this study aims to contribute to the field of network traffic management, provide valuable insights for network administrators and policymakers, and offer practical solutions for optimizing network performance, enhancing security, and adapting to the dynamic online landscape of corporate offices.

1.3 Research Questions

This research aims to address the following key research questions:

1. How can DNS log analysis, combined with data manipulation, visualization, and machine learning techniques, contribute to effective network traffic management?

    - This question focuses on exploring the potential benefits and insights gained from analyzing DNS logs using various techniques, such as data manipulation, visualization, and machine learning. The objective is to understand how these approaches can enhance network traffic management practices.

2. Can browsing categories be accurately predicted based on DNS log analysis, and how can this prediction capability be integrated into an SDN controller for real-time traffic management?

    - This question delves into the feasibility of predicting browsing categories using DNS log analysis. It seeks to determine the accuracy of such predictions and investigates methods for integrating these predictive models into an SDN controller to enable real-time traffic management based on browsing categories.

3. How can the identification and enforcement of specific policies in firewall systems be improved to address challenges posed by URL modifications and variations in domain names?

    - This question focuses on enhancing the capabilities of firewall systems to identify and enforce policies effectively in the presence of URL modifications and domain name variations. It seeks to explore innovative approaches to improve URL identification and policy enforcement, addressing the specific challenges posed by evolving URL structures.

4. What insights can be gained from the analysis of user behavior, network usage patterns, and browsing category preferences during different times of the day, office hours, and demographic factors?

    - This question aims to understand user behavior, network usage patterns, and browsing category preferences within specific contexts, such as different times of the day, office hours, and demographic factors. The objective is to uncover insights that can inform network traffic management strategies and optimize resource allocation based on user characteristics and browsing habits.

By addressing these research questions, this study seeks to contribute to the field of network traffic management by providing a comprehensive understanding of the benefits of DNS log analysis, predictive modeling for browsing categories, and enhancements to firewall systems' policy enforcement capabilities. The findings will offer valuable insights for network administrators, policymakers, and organizations looking to optimize network performance, enhance security, and adapt to the evolving digital landscape.

Chapter 2

Literature Review

The popularity of the internet has sparked numerous research endeavors focused on DNS log analysis. These studies encompass various aspects, including user behavior analysis, resource prediction and allocation, network activity monitoring, and security threat detection. Existing research has provided valuable insights into user behavior, enabling optimized resource allocation and improved user experience. Additionally, DNS log analysis has been instrumental in predicting future requests and facilitating proactive network management. Furthermore, researchers have utilized DNS logs to detect anomalies, identify malware, and predict cyber-attacks. This literature review aims to critically evaluate the existing research, identify gaps, and contribute to the field by incorporating data manipulation, visualization, and machine learning techniques for deeper insights into network behavior and proactive traffic management.

2.1 Review of relevant literature and prior research

Zhiyang Sun et al. [8] conducted a study in which they analyzed DNS queries from seven universities. Their objective was to accurately predict changes in the request volume of DNS queries, specifically for optimizing content delivery network (CDN) services. To achieve this, they employed machine learning and deep learning models, successfully demonstrating the effectiveness of their approach.
Jianfeng et al. [9] proposed a novel approach to overcome the challenges of behavior ambiguity and behavior polymorphism in DNS log analysis. They introduced the concept of pattern upward mapping and developed a multi-scale random forest classifier. By analyzing raw DNS queries, they were able to characterize user activity of interest, providing valuable insights into user behavior patterns.
Qingnan et al. [10] focused their research on understanding DNS lookup behaviors using DNS logs collected from three DNS servers located within a large university campus in China. In addition to analyzing user behaviors, they also aimed to identify potential security threats and gain an overall understanding of the network's tendencies. To enhance their analysis, they introduced visualization techniques, enabling intuitive interpretation and exploration of the collected data.
Jessie et al. [11] conducted a comprehensive study utilizing long-term flow-based internet traffic logs from a university. Their research encompassed two key aspects:

traffic flow prediction and user group analysis. They demonstrated that domestic traffic flow prediction is more reliable than international traffic flow prediction. Furthermore, they uncovered the significant influence of a user's academic field and occupation on their online behavior, shedding light on the complex interplay between user characteristics and internet usage patterns.

Shize et al. [12] embarked on a comprehensive measurement and analysis of adult websites and traffic within IPv6 networks. By analyzing adult website traffic, prospective users, and Internet Service Providers (ISPs), they aimed to gain insights into the behavior and strategies employed by adult website owners. Notably, their research highlighted the role of CDN vendors in promoting the development of adult websites within IPv6 networks, as well as the utilization of multi-domain policies by website owners to evade ISP restrictions.

Han Hu et al. [13] conducted a large-scale measurement to explore social video viewing behavior within the context of community classification. Their research was geared towards optimizing content delivery by formulating it as a constrained optimization problem. Their algorithm aimed to minimize operational costs while satisfying Quality of Service (QoS) requirements. The results demonstrated that their approach achieved a better tradeoff between monetary cost and QoS, outperforming traditional methods and resulting in reduced operational costs without compromising QoS.

Hadrien et al. [14] conducted a comparative study of DNS services, specifically between ISP's DNS servers and Google's public DNS. Their research employed Bayesian networks to evaluate the performance of these services. The findings revealed performance benefits for clients utilizing their ISP's DNS service, showcasing the advantages of leveraging ISP infrastructure for DNS resolution.

Weiwei Jiang [15] conducted an extensive study on internet traffic prediction, comparing deep neural network models with statistical baseline models. Utilizing an open Internet bandwidth usage dataset collected over six months, the research aimed to determine the superior approach for traffic prediction. The results demonstrated that deep neural networks consistently outperformed statistical baseline models, achieving lower Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values, thus exhibiting superior prediction accuracy.

Qinge Xe et al. [16] proposed a novel approach named Deep Incident-Aware Graph Convolutional Network (DIGC-Net) for traffic speed prediction. Their research incorporated various features such as traffic incidents, spatio-temporal patterns, periodicity, and contextual information. By leveraging real-world internet traffic datasets collected from Francisco and New York City, they demonstrated the effectiveness of their model in accurately predicting traffic speed.

Hongbo et al. [17] introduced a new method for analyzing DNS query log files, employing the Growing Hierarchical Self-Organizing Map (GHSOM). Their research aimed to identify differences in DNS query structures between normal and infected computers. The experimental results provided insights into the distinctive characteristics of DNS queries associated with infected hosts, enabling effective classification and detection of potential threats.

Sunil Kumar et al. [18] focused on detecting malicious activity at the DNS level within the DNS over TLS (DoH) environment. Their research employed various machine learning classifiers to identify and prevent DNS attacks on DoH traffic. The results confirmed the effectiveness of Random Forest (RF) and Gradient Boost-

ing (GB) classifiers in detecting and mitigating malicious activity, highlighting the potential of machine learning-based algorithms for enhancing DNS security.
Pratik et al. [19] proposed an anomaly-based intrusion detection system specifically designed for the DNS protocol. Their approach involved modeling the normal operations of the DNS protocol and accurately detecting any abnormal behavior or exploitation. By creating separate databases for normal and abnormal DNS traffic transitions and employing classification algorithms such as the Bagging algorithm, they achieved a high attack detection rate of 97
Ryo Nakamura et al. [20] developed a novel method to detect malicious hosts based on their behavior characteristics when sending SYN packets. They converted SYN packets into visual images, leveraging the unique features of hosts' behavior, and employed a convolutional neural network model for successful differentiation between malicious and normal hosts.
Igor Voronov and Kirill Gnezdilov [21] conducted research aimed at identifying and classifying client devices and applications based on DNS traffic analysis. Their study revealed that different operating systems (OS) and web services exhibit distinct DNS query patterns. They developed an algorithm that achieved 100% accuracy in identifying applications, while the accuracy for OS identification varied based on the length of datasets and time intervals.
Yue Li et al. [22] implemented a methodology to evaluate and optimize the performance of various machine learning and deep learning approaches for the real-time classification and detection of malicious DNS over HTTPS (DoH) traffic. Their research involved tuning hyperparameters and comparing models such as Random Forest, Decision Tree, K-Nearest Neighbors (KNN), 1D Convolutional Neural Network (CNN), 2D CNN, and Long Short-Term Memory (LSTM) networks. The evaluation metrics considered included Precision, Accuracy, Recall, and F1-Score, and the results highlighted the superiority of Random Forest and Decision Tree models, with precision and recall exceeding 99% and 98%, respectively, after hyperparameter optimization.
Keiichi Shima et al. [23] aimed to classify DNS servers based on their utilization as reflectors. They captured a small number of DNS response messages sent from DNS servers and employed machine learning techniques to develop a feature matrix. Classification using Support Vector Machines (SVM) yielded satisfactory precision, particularly when using training and test data from the same day.
Chunyu Han et al. [24] focused on domain classification methods and introduced the concept of "lonesome" DNS traffic. They defined lonesome DNS traffic as queries associated with domains that had few duplicate domain request records and short periods. The research involved real-world experiments to detect benign domains using machine learning methods such as Support Vector Machines, Logistic Regression, and Deep Neural Networks. The results showcased an average Area Under the Curve (AUC) of 95.64% and an average false positive rate of 0.542%, demonstrating the effectiveness of the applied machine learning techniques.
Keisuke ISHIBASHI et al. [25] proposed a novel method for identifying spatial characteristics of DNS traffic in a multi-resolution manner. They introduced the concept of hierarchical aggregate entropy, which involved recursively aggregating queries into upper domains along the DNS domain tree and calculating entropies. The research demonstrated that hierarchical aggregate entropy reduced classification errors compared to non-hierarchical entropies, achieving an accuracy of 92

Abdallah Moubayed et al. [26] presented an optimized machine learning-based framework for detecting botnets based on DNS queries. Their approach involved using information gain as a feature selection method and employing a genetic algorithm (GA) for hyperparameter optimization. The framework utilized a random forest (RF) classifier and achieved high detection accuracy, precision, recall, and F-score compared to the default classifier.
Martin et al. [27] conducted research on the classification of public DNS resolvers using data derived from NetFlow. By employing Random Forest and Gradient-Boosted Trees on various NetFlow-related features, they demonstrated the feasibility of classifying DNS resolvers as well-known or malicious with an Area Under the Receiver Operating Characteristic (AUROC) of 0.85 (approximately 0.80 using undersampling).

2.2 Research gaps

While several studies have focused on analyzing DNS logs for various purposes such as user behavior analysis, resource allocation, and malware detection, there is a lack of comprehensive research that combines DNS log analysis with data manipulation, visualization, and machine learning techniques for effective network traffic management. Existing research primarily focuses on specific aspects of DNS log analysis, such as anomaly detection, user behavior analysis, and security threat detection. However, there is a need for a holistic approach that encompasses multiple dimensions of DNS log analysis to provide a comprehensive understanding of network activity and potential security risks. Although some studies have explored the prediction of browsing categories based on DNS log analysis, there is still a gap in understanding the feasibility and accuracy of such predictions. Further research is needed to investigate the performance of machine learning models in predicting browsing categories and their potential integration into traffic management systems. The integration of predictive models based on DNS log analysis into Software-Defined Networking (SDN) controllers for real-time traffic management is an area that requires further exploration. Research is needed to evaluate the effectiveness and efficiency of integrating predictive models into SDN controllers to enhance network performance and security. The impact of the post-COVID era and the increasing reliance on online operations by corporate offices on network traffic management has received limited attention. There is a need to explore the specific challenges and requirements of dynamic and smart internet traffic management in the context of post-COVID corporate operations. The adaptations and variations in domain names, URL structures, and URL shortening techniques used by popular websites pose challenges for traditional firewall systems in accurately identifying and enforcing policies. Research is needed to develop innovative approaches to address these challenges and improve the effectiveness of URL filtering and policy enforcement mechanisms.
By addressing these research gaps, future studies can contribute to the field of network traffic management by providing comprehensive approaches, predictive models, and adaptive solutions that enhance network performance, security, and resource utilization in the evolving digital landscape.

Table 2.1: Summery of Literature Review 1

Author/Date	**Research Objective**	**Datasets Used**	**Methods used**	**Analysis and Results**	**Future research scope**
Zhiyang Sun, Tiancheng Guo, Shiyu Luo, Yingqiu Zhuang, Yuke Ma, Yang Chen, Xin Wang 2022	13 Chinese CDN companies' CNAME records analyzed for DNS access.	DNS request logs collected from top seven universities in shanghai.	Using statistical, machine learning, and deep learning models to predict DNS requests.	Deep learning models outperform traditional models, statistical models least effective.	Explore extended datasets and IPV6/IPV4 development in China.
Jianfeng, Xiaobo Ma, Li Guodong, Xiapu Luo, Junjie Zhang, Wei, Xiaohong Guan 2018	Develop pattern upward mapping and random forest classifier for analyzing DNS queries.	10-day network traffic analysis reveals 159 end users, 0.16% DNS queries.	New hierarchical characterization method and MRF classifier identify user activities in DNS.	Website visits vary by day, with "Baidu" experiencing high midnight visits.	Proposed methods accurately identify user activities in DNS queries.
Sunil Kumar singh, Pradeep Kumar Roy 2020	Detect malicious DNS activity in DoH environment using machine learning classifiers.	benchmark MoH dataset (CIRA-CIC-DoHBrw-2020).	Machine learning classifiers detect malicious DNS activity in DoH environment.	ML-based algorithms effectively prevent DNS attacks on DoH traffic.	N/A
Qingnan Lai, Changling Zhou, Hao Ma, Zhen Wu, Shiyang Chen 2015	Studying DNS lookup behaviors using Chinese university campus network logs.	Peking University's DNS logs show wired and wireless clients.	Passive DNS software captures interdomain messages, tcpdump captures network flow, and introduces DNSReduce.	4.5% failed name resolving requests; IPv4 and IPv6 IPv4 and IPv6 TTLs examined.	Proposed approaches aid in understanding macro- and micro-DNS lookup behaviors.
Jessie Hui Wang, Changqing An, Jiahai Yang 2011	Understanding university internet traffic, user behavior, and pricing policies for effective network planning.	Annual traffic and user logs on campus network boundary.	Manual prediction calculations using correlation coefficient and vector analysis.	74% of international traffic comes from 5 countries, with USA leading.	Study shows China university network traffic variation, useful for networking researchers.
Hongbo Shi and Kazuhiko Iwasaki 2013	To detect malware and DDoS attacks from DNS queries using machine learning.	DNS log file captured at a subnetwork of a campus network.	Cluster DNS queries using GHSOM machine learning algorithm.	Proposal reduces DNS log size and differentiates between normal and infected computers.	N/A
Pratik Satam, Hamid Alipour, Youssif Al-Nashif, Salim Hariri 2015	Present an anomaly-based intrusion detection system for DNS protocol.	Database comparing normal and abnormal DNS traffic transitions.	Classification algorithms like the Bagging algorithm.	Achieved 97% attack detection with low false positive alarms.	N/A
Ryo Nakamura, Yuji Sekiya, Daisuke Miyamoto, Kazuya Okada, Tomohiro Ishihara 2018	Novel method to detect malicious hosts based on their behavior characteristics.	1,815,267 SYN packets from 633 IP addresses.	Converting SYN packets to a visual image as input for neural networks.	Achieved 98% accuracy for the test data after 20 epochs in training.	Consider additional SYN fields, packet count, dataset accuracy, and comparisons.

Table 2.2: Summery of Literature Review 2

Author/Date	**Research Objective**	**Datasets Used**	**Methods used**	**Analysis and Results**	**Future research scope**
Igor Voronov and Kirill Gnezdilov 2021	Analyze DNS traffic to identify and classify client devices and applications.	Local network data collected using wireshark.	Classification algorithms match DNS queries to regular expressions.	OS and web services use unique DNS queries; algorithm accurately identifies applications.	N/A
Keiichi Shima, Ryo Nakamura, Kazuya Okada, Tomohiro Ishihara, Daisuke Miyamoto, Yuji Sekiya 2019	Classify DNS servers based on reflector usage..	Single day data of certain research network.	Converting SYN packets to visual input for neural networks..	SVM classification achieves precision with daily training and testing data.	Improve stability in results by investigating data, matrix generation, and classification algorithms.
Chunyu Han, Yongzheng Zhang, Yu Zhang 2020	Create a domain classification method for low-duplicate DNS records.	38,036,318 domain requests and response records; machine learning techniques used.	Machine learning techniques like Support Vector Machine, Logistic Regression.	Average AUC 95.64%, false positive rate 0.542%.	Subdivide domains for research purposes.
Keisuke ISHIBASHI, Kazumichi SATOH 2017	Hierarchical aggregate entropy calculates queries' recursive entropies in DNS domain trees.	Actual DNS query data measured during busy hours.	Identifying DNS client hosts' hierarchical aggregate entropies of queries.	Hierarchical aggregate entropy reduces classification error by 92%.	Identifying spammers can be incorporated in future study.
Abdallah Moubayed, MohammadNoor Injadat, and Abdallah Shami 2020	ML-optimized framework for detecting botnets based on DNS queries.	TI-2016 DNS dataset.	Utilize information gain and genetic algorithm for RF classifier parameter tuning.	High detection accuracy, precision, recall, F-score compared to default classifier.	Expanding framework, exploring hybrid ML models for online botnet detection.
Martin Fejrskov; Jens Myrup Pedersen; Emmanouil Vasilomanolakis 2022	Provide classification of public DNS resolvers using data derived from netflow.	405 resolvers seen in four weeks of NetFlow data from a national ISP.	Using Random Forest and Gradient-Boosted Trees on 7 different NetFlow.	AUROC 0.85 enables well-known/malicious resolver classification.	Hybrid approach can be considered to increase classification accuracy.
Yue Li, Abdulhalim Dandoush, Ji Liu 2021	Explore machine learning and deep learning for real-time malware detection.	N/A	Optimized models' performance using hyperparameter tuning and comparing metrics.	Random Forest and Decision Tree models outperform machine and deep learning models.	N/A
Abdallah Moubayed, MohammadNoor Injadat, and Abdallah Shami 2020	A novel optimized ML-based framework to detect botnets based on their corresponding DNS queries.	TI-2016 DNS dataset.	Utilize information gain and genetic algorithm for RF classifier parameter tuning.	High detection accuracy, precision, recall, F-score compared to default classifier.	Expanding framework, exploring hybrid ML models for online botnet detection.

Table 2.3: Summery of Literature Review 3

Author/Date	**Research Objective**	**Datasets Used**	**Methods used**	**Analysis and Results**	**Future research scope**
Shize Zhang, Hui Zhang, Jiahai Yang, Guanglei Song, Jianping Wu 2019	Analyze adult websites and IPv6 traffic for operational issues.	Raw packet traffic from CNGI-CERNET2 which is a pure IPv6 academic network in China.	Naïve Bayes algorithm classifies website content for targeted traffic filtering.	30% of adult websites accessible in IPv6, 95% in IPv4.	Future work focuses on IPv6 network security and adult website analysis.
Han Hu, Yonggang Wen, Tat-Seng Chua, Zhi Wang, Jian Huang, Wenwu Zhu, and Di Wu 2014	Algorithm optimizes operational cost, QoS, and outperforms traditional methods.	Utilized Sina Weibo API to retrieve 57,445 video tweets.	Lynapunov optimization framework verified dynamic algorithm using Weibo messages and Amazon cloud pricing model.	Derived algorithm improves balance, lowers costs, and enhances QoS.	N/A
Hadrien Hours, Ernst Biersackc, Patrick Loiseaua, Alessandro Finamore, Marco Mellia 2016	Compare ISP's DNS server and googles public DNS server.	N/A	Bayesian network	ISP DNS service offers performance benefits for clients.	Improve Causal knowledge inference framework performance with larger, partitioned data.
Weiwei Jiang 2021	Improve internet traffic prediction accuracy	Public internet traffic dataset collected for 6 months	13 deep neural networks compared to baseline models for 6 months.	Deep neural networks achieve lower RMSE, MAE, and prediction errors.	Deep neural networks investigate predicted bandwidth demand further.
Qinge Xie, Tiancheng Guo, Yang Chen, Yu Xiao, Xin Wang and Ben Y. Zhao 2020	Achieve a better prediction of traffic speed.	Real-world traffic datasets from Francisco and New York City..	DIGC-Net integrates traffic incident, spatio-temporal, periodic, and context features.	DIGC-Net outperforms framework, validates latent incident features effectiveness.	N/A
Syed Abed Hossain, Dr. Amitabha Chakrabarty 2023	Develop a comprehensive approach that combines DNS log analysis, data manipulation, visualization, and machine learning techniques for effective network traffic management	Real world DNS log collected from a local office in Dhaka.	Evaluate various machine learning models to predict the browsing category in integrate the best performing model in a real world SDN environment.	Random forest classifier stood out as the top-performing model, exhibiting high accuracy, precision, recall, and F1 score on mentioned dataset.	Dynamic feature selection and incremental learning to improve accuracy can be explored.

2.3 Related Works

This section presents a comprehensive overview of research conducted in the field of DNS log analysis and its applications in network traffic management. The studies reviewed cover a wide range of topics, including user behavior analysis, resource allocation, malware detection, traffic prediction, and security threat detection. Several researchers have explored the use of machine learning and deep learning models to analyze DNS queries and predict changes in request volume. For instance, Zhiyang Sun et al. [8] developed models to accurately predict the change in DNS request volume, focusing on optimizing CDN service performance. Similarly, Jianfeng et al. [9] proposed techniques to characterize user activity by analyzing raw DNS queries, overcoming behavior ambiguity and polymorphism challenges.

DNS log analysis has also been utilized to understand user behavior, security threats, and network tendencies. Qingnan et al. [10] conducted a study using DNS logs from a university campus to visualize DNS lookup behaviors and identify user patterns. Wang et al. [11] analyzed internet traffic logs to predict domestic and international traffic flows, and discovered the influence of user academic field and occupation on online behavior.Researchers have investigated the role of DNS logs in detecting and preventing cyber-attacks. Shize et al. [12] analyzed adult website traffic in IPv6 networks, while Sunil Kumar et al. [18] focused on detecting malicious activity at the DNS level in a DNS over TLS environment. Pratik et al. [19] developed an anomaly-based intrusion detection system for the DNS protocol, achieving high attack detection rates with low false positive alarm rates.

Additionally, studies have explored the classification and identification of DNS traffic and server behavior. Igor Voronov and Kirill Gnezdilov [21] analyzed DNS traffic to identify client devices and applications, while Keisuke Ishibashi et al. [25] introduced hierarchical aggregate entropy for spatial analysis of DNS traffic. These studies demonstrated the effectiveness of machine learning models, such as random forest, support vector machine, and neural networks, in classifying DNS traffic and detecting malicious behavior.

Overall, the studies reviewed provide valuable insights into user behavior, resource optimization, security threats, and the detection of malicious activities. However, certain research gaps exist, such as the need for comprehensive approaches, integration with Software-Defined Networking, and addressing emerging challenges in the post-COVID era. By addressing these gaps, future research can contribute to more effective and intelligent network traffic management systems.

Chapter 3

Methodology

The methodology section of this research paper outlines the approach and procedures followed to investigate the analysis of DNS logs and its application in network traffic management. In this section, we describe the step-by-step process that was undertaken, starting from the implementation of a local DNS server and culminating in the deployment of a predictive model integrated with an SDN controller for real-time traffic management.

To investigate the DNS log analysis and its implications for network traffic management, a systematic and rigorous methodology was followed. First, a diverse range of literature sources was extensively reviewed to gain a comprehensive understanding of existing techniques and approaches in the field. The literature review served as the foundation for designing our research methodology.

After that, a local DNS server was implemented within the office network environment. This server was responsible for capturing and logging DNS queries and responses generated by the network users. Over the course of one month, the DNS server collected a substantial amount of log data, capturing information about user browsing behavior and other relevant details. The collected DNS log data was then extracted and imported into Google BigQuery, a powerful platform for data modification and manipulation. Within this environment, various data processing techniques were applied to enhance the dataset. Additional features such as age and gender were incorporated, allowing for further analysis and segmentation of users based on demographic attributes. Moreover, specific columns such as weekday and office hour were derived from the timestamp information in the DNS logs. These additional columns facilitated the exploration of user engagement patterns during different times of the day and week. Subsequently, the modified dataset was imported into Google LookerStudio, a visualization tool, enabling in-depth analysis and data visualization of various aspects, including an overview of the data, user engagement patterns by office hour, and browsing category analysis based on gender and age groups.

In parallel to the data visualization and analysis, supervised and unsupervised machine learning algorithms were employed to predict browsing categories based on DNS log analysis. Several models were trained and evaluated using the collected dataset, with the aim of selecting the most accurate model for deployment. The selected model was then exported and integrated into an SDN controller, enabling the prediction of DNS categories in real-time traffic and the implementation of corresponding policies, such as access control lists (ACL), traffic routing, and bandwidth

management. The methodology employed in this research encompasses data collection through the local DNS server, data preprocessing and modification using Google BigQuery, visualization and analysis using Google LookerStudio, and the application of supervised and unsupervised machine learning techniques for browsing category prediction. The integration of the selected model with an SDN controller further demonstrates the practical implementation of the findings for real-time traffic management.
In the subsequent sections, each step of the methodology will be elaborated upon, providing a comprehensive understanding of the procedures undertaken and the insights gained from the analysis of DNS logs in the context of network traffic management.

3.1 Research design and approach

The Research Design and Approach section of this study presents the methodology and framework employed to investigate and analyze DNS log data collected from a local corporate office in Dhaka. The research design involved the selection of a target location and the implementation of NxFilter as the chosen DNS server for data collection. The collected DNS log data, along with additional derived variables, were imported into Google BigQuery for data manipulation and analysis. The section discusses the variables and measures used in the study, the data analysis techniques applied, ethical considerations regarding privacy and confidentiality, as well as the limitations to be considered throughout the research.

3.1.1 Research Design

For this study, a quantitative research design was employed to analyze the DNS log data collected from a local corporate office in Dhaka. The research design involved the selection of a target location and the implementation of a DNS server, specifically NxFilter, to collect and store the DNS log data. The choice of NxFilter was based on its ability to store long-term data and provide predefined sorting of the dataset, facilitating further analysis. Additionally, the data collected was enhanced by adding additional columns through cross-referencing with the office staff database using SQL queries. This research design aimed to explore and analyze the DNS log data for various insights and predictions.

3.1.2 Variables and Measures

The collected DNS log dataset consisted of several variables, including count, type, domain, user, client IP, group, policy, and category. To augment the dataset, additional columns were extracted from the office staff database, such as gender, weekday, age, age group, weekday number, office hour, time hour, time hour order, time hour-minute, and age group order. These additional variables were primarily used for visual analysis purposes in Lookerstudio and did not directly influence the machine learning models. Moreover, certain variables essential for training the machine learning models were derived from the domain column using SQL queries. These variables included URL length, entropy, special character count, subdomain number, query parameter number, hyphen number, dot number, and number count.

3.1.3 Exploratory Data Analysis

The collected DNS log data, along with the derived variables, were imported into Google BigQuery for data manipulation and analysis. Various data analysis techniques, including descriptive statistics, visualization, and machine learning algorithms, were applied to gain insights and make predictions. The data analysis process involved exploring the overall data overview, examining user engagement patterns based on office hours, performing browsing category analysis by gender, age group, and office hour, and training machine learning models to predict the browsing category based on the DNS log analysis.

3.1.4 Ethical Considerations

Throughout the data collection process, privacy considerations were taken into account, and the names of the users were kept hidden to ensure confidentiality. The research adhered to ethical guidelines and protected the privacy of individuals by anonymizing personal identifiers. The study also complied with relevant data protection regulations and ensured the security and confidentiality of the collected DNS log data.

3.1.5 Limitations

While this research design and approach provide valuable insights into DNS log analysis, there are several limitations to consider. Firstly, the study focused on collecting and analyzing DNS log data from a specific local corporate office in Dhaka, which may limit the generalizability of the findings to other contexts. Additionally, the chosen DNS server, NxFilter, and the specific dataset columns may introduce certain biases and limitations in terms of the data collected and the variables available for analysis. Furthermore, the accuracy and effectiveness of the machine learning models in predicting the browsing category may depend on the quality and representativeness of the dataset, as well as the chosen algorithms. Finally, the study is subject to potential limitations associated with data collection, such as data incompleteness or missing values, which could impact the analysis and outcomes.

3.2 Data Collection

The data collection process involved extracting DNS logs from the selected DNS server, NxFilter, deployed within the local corporate office network. Each day, the DNS logs were extracted from NxFilter, capturing valuable information about the DNS queries made within the network. These daily log files were subsequently merged into a single consolidated table, facilitating comprehensive analysis and ensuring a unified dataset.

The merged dataset consisted of a total of 6,715,309 rows and 29 columns. Each row represented a unique DNS query recorded during the one-month data collection period. The 29 columns encompassed various attributes and characteristics of the DNS queries, providing valuable insights for subsequent analysis.

The columns included essential information such as the count of requests, query types, queried domain names, user identifiers, client IP addresses, user groups,

applied policies, and assigned categories. These attributes allowed for a detailed examination of DNS activities within the office network.
Furthermore, to enrich the dataset and enable more comprehensive analysis, additional columns were incorporated by leveraging SQL queries and cross-referencing with the office staff database. These supplementary columns included gender, weekday, age, age group, weekday number, office hour, time hour, time hour order, time hour-minute, age group order, and website. Although not directly utilized in the machine learning models, these columns provided valuable contextual information and enabled in-depth visual analysis using Lookerstudio.
In addition to the above, specific columns crucial for training the machine learning models were derived from the domain column using SQL queries. These derived columns, such as URL length, entropy, special character count, subdomain number, query parameter number, hyphen number, dot number, and number count, served as important features for predicting the category column values in the subsequent machine learning models.
Throughout the data collection process, utmost consideration was given to privacy and ethical concerns. To protect the identities of individual users and maintain confidentiality, the names of users were anonymized, ensuring compliance with privacy regulations and ethical guidelines.

3.3 Overview of tools, technologies, and frameworks used

To facilitate the research objectives, several tools, technologies, and frameworks were employed throughout the project. These resources played a crucial role in data collection, preprocessing, analysis, machine learning, and integration within a Software-Defined Networking (SDN) environment. The overview shown in figure 3.1 highlights the key components utilized in each stage.

3.3.1 Virtualization Environment and Containerization

To create a Linux-based container for hosting the DNS log collection server (NxFilter), a virtualization environment called Proxmox was employed. This allowed for the efficient utilization of hardware resources and isolation of the server environment. The container hosting NxFilter was configured with 2 cores, 2 GB RAM, 50 GB storage, and ran on Ubuntu 20.04 OS.

3.3.2 DNS Log Collection and Forwarding

NxFilter served as the DNS server responsible for receiving DNS requests forwarded by the core router of the office network. It logged the DNS queries and subsequently forwarded them to Google's DNS (8.8.8.8) for resolution. This setup ensured the capture of DNS traffic within the network for further analysis.

3.3.3 Cloud Storage and Analysis

Google BigQuery, a scalable data warehouse solution, was utilized for the permanent storage and analysis of the collected DNS log data. It provided a reliable and scalable

Data Collection	Set up a local DNS server using NxFilter. Configure the core router to forward DNS requests to NxFilter. Collect DNS log data from the office network for a one-month period.
Data Preprocessing and Storage	Import the DNS log data into Google BigQuery for storage and analysis. Perform data modification and manipulation using SQL queries in BigQuery. Add additional columns to the dataset by cross-referencing with the office staff database. Extract relevant columns for visualization and analysis using Google Lookerstudio.
Data Analysis and Visualization	Utilize Lookerstudio to explore and analyze the dataset. Generate visualizations such as pivot tables, charts, and graphs to gain insights into data patterns and user behavior.
Machine Learning	Import the modified dataset from BigQuery into a Python environment. Employ various supervised and unsupervised machine learning algorithms to predict browsing categories based on the DNS log analysis. Evaluate the performance of different models using metrics such as accuracy, log loss, F1 score, precision, and recall.
Model Deployment and Integration	Identify the model with the highest accuracy. Export the selected model as a .joblib file. Develop a Python script to integrate the trained model into an SDN environment. Intercept DNS requests, extract relevant information, and predict website categories using the deployed model. Apply appropriate policies and actions based on the predicted categories (e.g., blocking, rerouting) using SDN controller instructions.

Figure 3.1: Research Framework

platform for storing and processing large datasets.

3.3.4 Data Analysis and Visualization

Google Lookerstudio was employed for in-depth analysis and visualization of the DNS log data. It allowed for the exploration and comparison of various columns within the dataset using a wide range of visualization tools, including pivot tables, time series charts, bar charts, pie charts, line charts, and scatter plots. This facilitated a comprehensive understanding of the data patterns and trends.

3.3.5 Data Manipulation

Google Notebook was utilized as a Python environment directly connected to Google BigQuery. This approach enabled the analysis of the dataset without the need for data movement across multiple environments. Pandas and pandas_gbq libraries were used for data manipulation and importing BigQuery tables. Various encoders, such as category encoder and label encoder, were employed for preprocessing the data. Sklearn (scikit-learn) was leveraged for tasks such as train-test splitting, model training, and evaluation.

3.3.6 Machine Learning

Supervised machine learning models, including random forest classifier, naive Bayes, support vector machines, k-nearest neighbors (KNN), decision tree, MLP (multilayer

perceptron), and AdaBoost, were tested. Additionally, unsupervised learning techniques such as elbow method, silhouette method, k-means clustering, DBSCAN, Gaussian mixture models, and mean shift clustering were utilized for clustering analysis. A brief description of the machine learning models used in this research are given below.

Random Forest Classifier

Random Forest [28], [29] is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Naïve Bayes Classifier

Naïve Bayes [30], [31], [32] algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{3.1}$$

Where,
$P(A \mid B)$ is Posterior probability: Probability of hypothesis A on the observed event B.
$P(B \mid A)$ is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

K-Nearest Neighbors (KNN)

KNN [33], [34] is a non-parametric classification algorithm that makes predictions based on the majority class of its k-nearest neighbors. The distance metric (e.g., Euclidean distance) determines which neighbors are considered. KNN is simple and intuitive, but it can be sensitive to noisy or irrelevant features. The K-NN working can be explained on the basis of the below algorithm:

1. Select the number K of the neighbors
2. Calculate the Euclidean distance of K number of neighbors

3. Take the K nearest neighbors as per the calculated Euclidean distance
4. Among these K neighbors, count the number of data points in each category
5. Assign the new data points to that category for which the number of neighbors is maximum
6. Our model is ready

Decision Tree

A Decision Tree [35], [36] is a hierarchical tree-like structure where each internal node represents a decision based on a specific feature, and each leaf node represents a class label. It splits the data into subsets based on the values of features, aiming to classify instances correctly. Decision Trees are easy to understand and can handle both classification and regression tasks.

1. Begin the tree with the root node, says S, which contains the complete dataset.
2. Find the best attribute in the dataset using Attribute Selection Measure (ASM).
3. Divide S into subsets that contain possible values for the best attributes.
4. Generate the decision tree node, which contains the best attribute.
5. Recursively make new decision trees using the subsets of the dataset created in step 3. Continue this process until a stage is reached where we cannot further classify the nodes and call the final node a leaf node.

Multilayer Perceptron (MLP)

MLP [37], [38] is a type of artificial neural network that consists of multiple layers of interconnected nodes (neurons). A multilayer perceptron (MLP) is a feed forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network. MLP is a deep learning method.

AdaBoost

Boosting is an ensemble learning method that combines a set of weak learners into strong learners to minimize training errors.AdaBoost (Adaptive Boosting) [39], [40] is an ensemble learning method that focuses on improving the performance of weak learners by combining their predictions. It is implemented by combining several weak learners into a single strong learner. The weak learners in AdaBoost take into account a single input feature and draw out a single split decision tree called the decision stump. Each observation is weighted equally while drawing out the first decision stump. The results from the first decision stump are analyzed, and if any observations are wrongfully classified, they are assigned higher weights. A new decision stump is drawn by considering the higher-weight observations as more

significant. Again if any observations are misclassified, they're given a higher weight, and this process continues until all the observations fall into the right class. Here's how the algorithm works.

- Step 1: The base algorithm reads the data and assigns equal weight to each sample observation.
- Step 2: False predictions made by the base learner are identified. In the next iteration, these false predictions are assigned to the next base learner with a higher weightage on these incorrect predictions.
- Step 3: Repeat step 2 until the algorithm can correctly classify the output.

AdaBoost can be used for both classification and regression-based problems. However, it is more commonly used for classification purposes.

Elbow Method

The Elbow method [41], [42] is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 2 clusters) is given below:

$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 \qquad (3.2)$$

$\sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$: It is the sum of the square of the distances between each data point P_i and its centroid C_1 within Cluster 1, and the same for the other two terms. To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance. To find the optimal value of clusters, the elbow method follows the below steps.

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- If the plot shows a sharp point of bend or a point that looks like an elbow, then that point is considered as the best value of K.

Silhouette Method

The Silhouette Method [43] evaluates the quality of clustering by measuring how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

- Compute the silhouette score for each data point using the formula: Silhouette score = $\frac{b-a}{\max(a,b)}$, where "a" is the average distance to the other points in the same cluster, and "b" is the smallest average distance to the points in the other clusters.
- The silhouette score ranges from -1 to 1. A higher score indicates better-defined clusters.

K-Means Clustering

K-Means Clustering [44], [45] is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The working of the K-Means algorithm is explained in the below steps:

- Step-1: Select the number K to decide the number of clusters.
- Step-2: Select random K points or centroids. (It can be other from the input dataset).
- Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
- Step-4: Calculate the variance and place a new centroid of each cluster.
- Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
- Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
- Step-7: The model is ready.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN [46], [47] is a density-based algorithm that creates clusters based on dense regions separated by sparse areas. The DBSCAN algorithm is explained in below steps.

- Start with an arbitrary point. If it has enough neighbors within a specified radius (epsilon), it becomes a core point.
- Expand the cluster by adding all points within epsilon distance to the core point.
- Repeat for other core points and their neighbors.
- Points that are not within epsilon distance of any core point are considered noise.

Gaussian Mixture Models (GMM)

Gaussian mixture models [48] are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning. The GMM algorithm is explained in below steps.

- Initialize K Gaussian distributions.
- Assign data points to the Gaussian distribution that maximizes the probability of generating the point.
- Update the Gaussian distributions' parameters based on the assigned data points.
- Repeat steps 2 and 3 until convergence.

Mean Shift Clustering

Mean Shift [49], [50] is a non-parametric algorithm that identifies dense regions in data by moving points towards the region's mode (highest density). The basic idea behind mean-shift clustering is to shift each data point towards the mode (i.e., the highest density) of the distribution of points within a certain radius. The algorithm iteratively performs these shifts until the points converge to a local maximum of the density function. The Mean Shift algorithm is explained in below steps.

- Start with each data point as a cluster center.
- Calculate the mean shift vector for each point based on nearby data points.
- Move the point in the direction of the mean shift vector.
- Repeat steps 2 and 3 until convergence.

3.3.7 Integration of Trained Model into SDN Environment

A Python script was developed to integrate the trained model into a software-defined networking (SDN) environment and test the feasibility of the research findings. This script operated at the application layer and had the ability to intercept DNS requests arriving at the DNS server. The primary purpose of the script was to extract relevant values from the DNS request packet, which were necessary inputs for the trained model to predict the category of the requested website. By passing these extracted values through the trained model, the script generated predictions based on the model's classification capabilities. Upon receiving the prediction, the script executed further processing of the packets based on predefined policies defined within the script. These policies instructed the SDN controller to take specific actions accordingly. For instance, the script could block DNS packets, reroute packets to a designated uplink, or implement various other actions applicable in the given scenario. By developing this Python script and integrating it into the SDN environment, we were able to validate the practicality and effectiveness of the trained model

in real-time DNS traffic analysis and control. This integration demonstrated the applicability of the research findings in enhancing network security, optimizing traffic routing, and implementing other network policies within the SDN infrastructure.

3.3.8 Ethical Considerations

During the data collection process, privacy considerations were taken into account by keeping the names of users hidden to protect their identities. Additionally, the research adhered to ethical guidelines regarding data usage and ensured compliance with data protection regulations. The analysis and reporting of the data focused on aggregated and anonymized insights, preventing the identification of individual users. Ethical considerations regarding data privacy, security, and confidentiality were paramount throughout the research process.
The system architecture depicted in Figure 3.1 illustrates the flow of data and interactions between the components involved. It provides a comprehensive view of the experimental setup and the integration of various tools, technologies, and frameworks to achieve the research objectives. By leveraging this experimental setup and system architecture, we were able to collect, preprocess, analyze, and model DNS traffic within the corporate office network. This approach facilitated the optimization of resource allocation and network management, ultimately enhancing the overall performance and efficiency of the network."

Chapter 4

Data Collection and Preprocessing

The Data Collection and Preprocessing chapter presents a comprehensive overview of the processes involved in gathering and preparing the data for analysis in our research. This chapter is divided into three sections: Description of the data collection process, Explanation of data preprocessing techniques, and Data cleaning and transformation methods.
The first section provides insights into the data collection process employed in our study. We detail the steps taken to capture DNS logs from the corporate office network over a period of one month. By leveraging a Linux-based container and the NXFilter DNS server, we were able to log and store detailed information about DNS traffic patterns and attributes within the network.
In the second section, we delve into the data preprocessing techniques applied to the collected dataset. Here, we explain the various methods employed to transform the raw data into a structured format suitable for analysis. This includes extracting relevant columns, cross-referencing with additional data sources, and enriching the dataset with contextual information.
The third section focuses on the data cleaning and transformation methods used to ensure the quality and integrity of the dataset. We describe the steps taken to handle missing values, address inconsistencies, and standardize data formats. Additionally, we outline any normalization or encoding techniques applied to prepare the dataset for subsequent analysis and machine learning tasks.
By providing a detailed account of the data collection process, data preprocessing techniques, and data cleaning and transformation methods, this chapter establishes a solid foundation for the subsequent analysis and findings presented in our research. It ensures that the data used in our study is reliable, consistent, and suitable for the research objectives at hand.Now, let us delve into each section to gain a deeper understanding of the intricacies involved in collecting, preprocessing, and preparing the data for analysis.

4.1 Description of the data collection process

The data collection process in our research involved capturing DNS (Domain Name System) logs from the corporate office network over a period of one month. This section provides a detailed description of the data collection process employed in our study.

To begin with, we implemented a Linux-based container using the Proxmox virtualization environment. Within this container, we hosted the NXFilter DNS server, which acted as the primary DNS resolver and logger for the office network. By configuring the core router of the network, we directed all DNS requests originating from the office network to the NXFilter server. It equipped with a predefined sorting mechanism, logged the DNS requests received from the network. It stored information related to each request, including the count, type, domain, user, client IP, group, policy, and category. This logging process ensured the availability of detailed DNS traffic data for analysis.

Throughout the one-month data collection period, the NXFilter server continuously recorded DNS activities within the corporate office network. Each day, the DNS logs were extracted from the NXFilter server and merged into a single table for further analysis. To augment the dataset with additional contextual information, we cross-referenced it with the office staff database. By utilizing SQL queries, we extracted relevant attributes such as gender, weekday, age, age group, weekday number, office hour, and more. These supplementary columns enriched the dataset and provided valuable insights for visual analysis using tools like Google LookerStudio. This consolidation resulted in a comprehensive dataset comprising 6,715,309 rows and 29 columns, encompassing a wealth of information about DNS traffic patterns and attributes.

To facilitate long-term data storage and scalable analysis, we leveraged the power of Google BigQuery, a cloud-based data warehouse solution. The entire dataset, including the merged DNS logs, was imported into BigQuery for seamless data management. The platform's SQL capabilities enabled us to perform data modification and manipulation tasks efficiently.The combination of the NXFilter server, the merging of daily logs, and the utilization of Google BigQuery facilitated the creation of a robust and comprehensive dataset encompassing various DNS traffic attributes and contextual information. This dataset served as the foundation for our subsequent data analysis, visualization, and machine learning tasks.

By meticulously collecting and curating the DNS logs from the corporate office network, we were able to generate a rich dataset that captured the intricacies of network traffic patterns and user behavior. This dataset formed the basis for our research and enabled us to explore and optimize resource allocation, network management, and policy enforcement within the SDN environment.

4.2 Explanation of data preprocessing techniques

In the data collection process, we carefully filtered and selected the relevant information to ensure meaningful analysis and model training. From the DNS packets extracted using nxfilter, we retained key variables such as time, count, type, domain, and category. These variables provided crucial insights into the timing, frequency, type, and nature of the requested domains. To focus on the most significant data and optimize the dataset size, we implemented a specific time range for data inclusion. As the standard office hours in our country span approximately 9 hours from 8 am to 5 pm, we only considered data generated within this time period. This deliberate selection allowed us to capture the most relevant and representative browsing activities during the typical working hours. Given the scale of the target environment, with around 300 users accessing the office network daily, the data generation

was substantial. Therefore, by narrowing our focus to devices used for official purposes, we effectively filtered out extraneous browsing activities. This ensured that our analysis and models were based on data directly related to work-related tasks, enhancing the relevance and applicability of our findings.
Due to the specific configuration of our data collection setup, missing data was minimal. The core router's redirection of all DNS traffic to our DNS server provided comprehensive coverage of browsing activities. This meant that we captured a vast amount of data from various devices, including laptops, desktops, tablets, and mobile devices, connected to the office network. As a result, the occurrence of missing data was negligible, as our setup ensured that we received DNS requests from nearly all devices within the office premises. This comprehensive data collection approach enabled us to minimize potential biases or gaps in the dataset, providing a robust foundation for our analysis.

4.2.1 Data Integration and Aggregation

Data Integration: Merging Multiple Tables in BigQuery

To consolidate the DNS log data collected over a month-long period, we employed data integration and aggregation techniques using Google BigQuery's powerful querying capabilities. Each day's DNS log data was stored in separate tables within the BigQuery project. To create a comprehensive dataset for analysis, we utilized BigQuery's UNION ALL operator to combine these individual tables into a single dataset.
The integration process involved executing a series of SQL queries, as exemplified by the command below, which merged the DNS log tables for specific dates:

```
SELECT * FROM ‘DNS_Log.Day1‘
UNION ALL
SELECT * FROM ‘DNS_Log.Day2  ‘
UNION ALL
...
SELECT * FROM ‘DNS_Log.Day21‘
UNION ALL
SELECT * FROM ‘DNS_Log.Day22‘
```

By applying the UNION ALL operation, we combined the rows from each daily DNS log table into a cohesive dataset. This process allowed us to integrate the individual data points while preserving their original structures and columns. Through this data integration technique, we successfully merged the DNS log data from multiple tables into a unified dataset containing 6,715,309 rows and 29 columns. This consolidated dataset formed the foundation for our subsequent analysis and modeling tasks. The aggregated dataset encompassed a wide range of information, including the timestamp, count, type, domain, user, and category of each DNS request. By consolidating the data in this manner, we created a comprehensive and versatile dataset that enabled us to extract valuable insights and patterns regarding users' browsing behaviors within the target office environment.

Extracting Demographic Information

In order to extract demographic data associated with user browsing behavior, we acquired a staff database that included information such as gender and age. To link this demographic data with our own database, we performed a matching operation between the staff names in the staff database and the usernames in our dataset. This process was accomplished using the following SQL query:

```
UPDATE maindataset
SET maindataset.age = staffdatabase.age,
    maindataset.gender = staffdatabase.gender
FROM staffdatabase
WHERE maindataset.username = staffdatabase.staff_name;
```

This query performs an update operation on the main dataset, setting the "age" and "gender" columns based on the matching values from the second table. The matching condition is specified using the WHERE clause, where the "username" column in the main dataset is compared with the "staff name" column in the staff database obtained from the organization HR. It's important to note that appropriate permissions were ensured to perform the update operation on our database. Furthermore, prior to executing this query, necessary precautions, such as creating backups, were taken to mitigate any potential unintended consequences.

Extracting Time-Related Features from Timestamps

To analyze user browsing behavior during different time periods within the office hour and on separate days and weeks throughout the month, we extracted several time-related values from the timestamp column of our main dataset. These values include the weekday, office hour, and time hour.
The weekday column represents the specific day of the week on which the browsing activity occurred. To extract this information, we utilized the following SQL query:

```
ALTER TABLE maindataset
ADD COLUMN weekday VARCHAR(20);

UPDATE maindataset
SET weekday = DATE_FORMAT(timestamp_column, '%W');
```

Here, the DATE_FORMAT function was employed to format the timestamp column as the full weekday name (%W). This query allowed us to retrieve the weekday value for each entry in our dataset.
Next, we extracted the office hour information, which denotes whether the browsing activity took place during the first half or second half of the office hours. The following SQL query was used:

```
ALTER TABLE maindataset
ADD COLUMN office_hour VARCHAR(20);

UPDATE maindataset
SET office_hour = CASE
    WHEN EXTRACT(HOUR FROM timestamp_column)
```

```
    >= 8 AND EXTRACT(HOUR FROM timestamp_column) <= 12 THEN
    'First Half'
    WHEN EXTRACT(HOUR FROM timestamp_column)
    = 13 AND EXTRACT(HOUR FROM timestamp_column) = 17 THEN
    'Second Half'
    ELSE 'Other'
END;
```

In this query, the EXTRACT function was utilized to extract the hour from the timestamp column. Subsequently, a CASE statement was employed to categorize the extracted hour as either the first half (8 AM to 12:59 PM), the second half (1 PM to 5 PM), or other hours. This allowed us to assign the corresponding label ('First Half', 'Second Half', or 'Other') to the office_hour column for each entry in our dataset.
We extracted the time hour, which represents the individual hour value within the timestamp column. The following SQL query was executed for this purpose:

```
ALTER TABLE maindataset
ADD COLUMN time_hour INT;

UPDATE maindataset
SET time_hour = EXTRACT(HOUR FROM timestamp_column);
```

This query simply employed the EXTRACT function to isolate the hour component from the timestamp column, providing us with the time_hour value for each entry in our dataset.
By extracting and analyzing these time-related values, we gained insights into user browsing behavior during different periods within the office hour and on separate days and weeks throughout the month.

4.2.2 Feature Extraction and Engineering

The extracted data collected from the DNS logs formed the basis for training machine learning models to predict website categories. However, the initial dataset consisted of only three key features: count, type, and domain, which were insufficient for building an accurate prediction model. To enhance the predictive power of the models, we performed further feature extraction from the domain values.
From the domain values, we extracted several numerical features that provided valuable insights into website characteristics. These additional features included

URL Length

```
ALTER TABLE maindataset
ADD COLUMN url_length INT;

UPDATE maindataset
SET url_length = LENGTH(domain);
```

The url_length column is added to the existing table to store the length of each domain. The LENGTH() function calculates the number of characters in the domain and updates the url_length column accordingly.

Entropy

```
ALTER TABLE maindataset
ADD COLUMN entropy FLOAT;

UPDATE maindataset
SET entropy =
-SUM((LENGTH(domain)/LENGTH(table))*LOG(LENGTH(domain)/LENGTH(table)))
    FROM maindataset
    GROUP BY table;
```

The entropy column is added to the table to capture the entropy of each domain. Entropy measures the randomness or unpredictability of the characters within the domain. The formula calculates the entropy value using the logarithmic function and updates the entropy column for each domain.
The calculation of entropy is performed using the following steps:

1. For each domain in the table:
    - Calculate the probability of each character by dividing the length of the domain by the total length of all domains (LENGTH(domain)/LENGTH(table)).
    - Multiply the probability by the logarithm of the probability.
    - Sum up these values for all characters in the domain.
2. The negative sum of these values is assigned to the entropy column for the respective domain.

Let's assume we have a domain D consisting of n characters, where each character has a probability of occurrence P(i). The entropy H(D) of the domain is given by:
H(D) = - P(i) * log2(P(i))
where represents the summation over all unique characters in the domain.
For example, let's consider a domain "google.com" For the domain "google.com", we can calculate the entropy as follows:
H(D) = - [(P('g') * log2(P('g'))) + (P('o') * log2(P('o'))) + (P('l') * log2(P('l'))) + (P('e') * log2(P('e'))) + (P('.') * log2(P('.'))) + (P('c') * log2(P('c'))) + (P('m') * log2(P('m')))]
Substituting the probabilities of each character in the formula, we can obtain the entropy value for the "google.com" domain.

Special Character Count

```
ALTER TABLE maindataset
ADD COLUMN special_char_count INT;

UPDATE maindataset
SET special_char_count = LENGTH(domain)
- LENGTH(REPLACE(domain, '-', ''))
- LENGTH(REPLACE(domain, '.', ''));
```

The special_char_count column is added to the table to store the count of special characters in each domain. The REPLACE() function is used to remove hyphens and dots from the domain, and the difference in lengths between the original domain and the modified domain gives the count of special characters.

Number of Subdomains

```
ALTER TABLE maindataset
ADD COLUMN subdomain_count INT;

UPDATE maindataset
SET subdomain_count =
LENGTH(domain) - LENGTH(REPLACE(domain, '.', '')) - 1;
```

The subdomain_count column is added to the table to capture the number of subdomains in each domain. The REPLACE() function removes the dots from the domain, and the difference in lengths between the original domain and the modified domain, minus 1, gives the count of subdomains.

Query Parameter Number

```
ALTER TABLE maindataset
ADD COLUMN query_param_count INT;

UPDATE maindataset
SET query_param_count = LENGTH(SUBSTRING_INDEX(domain, '?', -1))
- LENGTH(REPLACE(SUBSTRING_INDEX(domain, '?', -1), '&', '')) + 1;
```

The query_param_count column is added to the table to store the number of query parameters in each domain. The SUBSTRING_INDEX() function extracts the portion of the domain after the '?' symbol, and the difference in lengths between the original substring and the substring with ampersands removed gives the count of query parameters.

Hyphen Number

```
ALTER TABLE maindataset
ADD COLUMN hyphen_count INT;

UPDATE maindataset
SET hyphen_count = LENGTH(domain) - LENGTH(REPLACE(domain, '-', ''));
```

The hyphen_count column is added to the table to represent the number of hyphens in each domain. The REPLACE() function removes hyphens from the domain, and the difference in lengths between the original domain and the modified domain gives the count of hyphens.

Dot Number

```
ALTER TABLE maindataset
ADD COLUMN dot_count INT;

UPDATE maindataset
SET dot_count = LENGTH(domain) - LENGTH(REPLACE(domain, '.', ''));
```

The dot_count column is added to the table to store the count of dots in each domain. The REPLACE() function removes dots from the domain, and the difference in lengths between the original domain and the modified domain gives the count of dots.

Number of Numbers

```
ALTER TABLE maindataset
ADD COLUMN number_count INT;

UPDATE maindataset
SET number_count = LENGTH(domain)
- LENGTH(REPLACE(domain, '0', ''))
- LENGTH(REPLACE(domain, '1', ''))
- ... - LENGTH(REPLACE(domain, '9', ''));
```

The number_count column is added to the table to capture the count of numerical digits in each domain. The REPLACE() function removes each numerical digit from the domain, and the difference in lengths between the original domain and the modified domain gives the count of numbers.

4.2.3 Data Encoding and Transformation

One of the fundamental challenges we encountered in our research was the need to convert the domain values, a significant feature carrying high importance, into numerical representations. Our selected machine learning models, crucial for prediction and analysis, only support numeric input. To address this issue, we employed feature encoding techniques to transform the string-based features into numeric values.

There exist various encoding methods, each with its own advantages and considerations. Our options included binary encoding, one-hot encoding, category encoding, and label encoding. After careful evaluation, we determined that label encoding was the most suitable choice for our research objectives.Label encoding is a technique used to convert categorical or textual data into numeric form, enabling its utilization with machine learning algorithms. In our research, we encountered the challenge of working with string values in the domain feature, which held significant importance in our model. To address this issue, we employed label encoding to convert the vast amount of string values into numeric representations.Label encoding assigns a unique numeric label to each unique category in the data, preserving the order of appearance. This encoding technique is implemented using the LabelEncoder class from the scikit-learn library. The process involves fitting the encoder on the

categorical data to learn the mapping between categories and their corresponding numeric labels, followed by transforming the data to obtain the encoded values.
For instance, consider a categorical data set comprising colors: 'red', 'blue', 'green', 'red', 'blue'. By applying label encoding, 'red' may be assigned the label 2, 'blue' the label 0, and 'green' the label 1. This numeric representation enables machine learning models to interpret and process the data effectively.
In our research, label encoding was chosen due to its simplicity and suitability for handling large amounts of categorical data. While other encoding methods, such as one-hot encoding and binary encoding, are available, they can result in high-dimensional data representations and increased complexity. Label encoding, on the other hand, provides a compact representation of the data, ensuring compatibility with the selected machine learning models.To utilize label encoding in Google Co-lab or Google Notebook, the necessary libraries, such as scikit-learn, need to be imported. The LabelEncoder class is then instantiated, and the fit() function is used to learn the mapping between categories and labels. Finally, the transform() function is applied to convert the categorical data into encoded numeric values. Label encoding enables us to convert the domain feature, which initially consisted of string values, into a numeric representation suitable for machine learning models. This conversion enhances our ability to train and predict with the selected models, contributing to the overall effectiveness of our research.
By employing label encoding, we successfully transformed the domain feature from a string-based representation to a numerical format, empowering our machine learning models to effectively utilize this critical feature. The encoded values retained the essential characteristics of the original data while enabling compatibility with our chosen models and analysis pipeline.

4.2.4 Data Splitting

The process of data splitting is a crucial step in machine learning model development, as it allows for the evaluation of model performance on unseen data. In our research, we employed a consistent data splitting strategy across multiple machine learning models, dividing our dataset into two subsets: a training set and a testing set. The dataset was divided in a stratified manner, ensuring that the distribution of target categories remained consistent in both the training and testing sets. This is particularly important when dealing with imbalanced datasets, where the occurrence of certain categories may be significantly higher or lower than others. This said scenario perfectly matches with our dataset. By maintaining a representative distribution of categories in both sets, we aimed to minimize the potential for bias and obtain reliable performance estimates.
The data splitting ratio chosen for our research was 80% for the training set and 20% for the testing set. This allocation allowed for an adequate amount of data for training the models while reserving a portion for unbiased evaluation. The training set, comprising 80% of the dataset, was utilized to train the machine learning models on the provided features and corresponding target labels. The testing set, accounting for the remaining 20%, was held out to assess the performance and generalization capabilities of the trained models on unseen data. By separating the data into distinct training and testing sets, we were able to assess the models' ability to generalize beyond the training data and make accurate predictions on new,

unseen samples. This approach facilitated the identification of potential overfitting or underfitting issues, as well as providing an estimate of the models' performance in real-world scenarios. It is worth noting that during the data splitting process, we ensured the preservation of temporal order, particularly in time-series data, to maintain the integrity of the dataset. This helps to simulate real-world scenarios where models are trained on historical data and tested on future data.

The chapter encompassed several critical stages in the data preparation and preprocessing pipeline, including data integration and aggregation, feature extraction and engineering, data encoding and transformation, as well as data splitting for model evaluation. In the Data Integration and Aggregation section, we combined multiple tables using the UNION ALL operation in BigQuery, consolidating the DNS log data from various days into a single dataset. This allowed us to work with a comprehensive and unified dataset for further analysis. Next, in the Feature Extraction and Engineering section, we extracted informative features from the domain values to enhance the predictive power of our models. This involved deriving numerical values such as URL length, entropy, special character count, number of subdomains, query parameter number, hyphen number, dot number, and number of numbers. These features provided valuable insights into the characteristics of the domains and their potential impact on website categorization. To address the challenge of working with string values in our machine learning models, we employed data encoding techniques in the Data Encoding and Transformation section. Specifically, we adopted label encoding to convert string features into numeric representations, enabling compatibility with our selected models. Label encoding assigned a unique numerical label to each distinct domain value, facilitating the utilization of these features in our predictive models. Finally, in the Data Splitting section, we divided the dataset into training and testing sets, allocating 80% of the data for training and reserving 20% for evaluating model performance. This stratified splitting ensured the preservation of class distribution in both sets and provided a reliable estimate of the models' generalization capabilities on unseen data.

Collectively, these stages in the data preprocessing pipeline played a crucial role in preparing the dataset for subsequent machine learning model training and evaluation. The integration and aggregation of data, along with feature extraction, encoding, and appropriate data splitting, laid the foundation for robust and reliable analysis of website categorization using machine learning techniques.

Chapter 5

Analysis and Results

This chapter presents the analysis and results of our research, divided into two main parts: visual analysis and insights, and the analysis and results of the supervised and unsupervised machine learning models used for domain category prediction.

5.1 Visual Analysis and Insights

In the visual analysis section, we explored the dataset using various visualization techniques to gain insights into the browsing behavior and website categorization. By visualizing the distribution of website categories, we identified the prevalence of different categories and any potential imbalances or biases present in the dataset. Heatmaps and bar charts provided a comprehensive overview of the frequency and distribution of domain features, revealing patterns and outliers that could influence website categorization.

We present the visual analysis and insights gained from our extensive dataset. While not directly contributing to the training and testing of machine learning models, this comprehensive dataset provides a wealth of information for demographic analysis, understanding user browsing behavior, identifying browsing trends, and more. Before delving into the classification phase, we first explore and discuss the overall data overview, overall user engagement, user engagement by office hour, and analyze the browsing category interests by gender and age group.

The dataset at hand represents a massive collection of data points, encompassing a wide range of variables and capturing diverse aspects of user interaction with online content. By examining the volume, variety, and velocity of the data, we gain an understanding of its scale and significance. This data overview sets the stage for our subsequent analysis and highlights the potential insights that can be derived. We examine the overall user engagement within the dataset. Through various metrics and visualizations, we explore the total number of unique users, average requests per user, and the most prevalent browsing activities. This analysis provides valuable insights into the level of user engagement and helps us identify patterns and trends that shape user behavior. We investigate user engagement based on office hours. By segmenting the dataset into distinct time periods representing the first half and second half of the office hours, we aim to uncover variations in user activity and preferences throughout the workday. Analyzing metrics such as the number of requests and browsing duration during these periods allows us to gain insights into how user behavior evolves within the office environment. Demographic

analysis plays a crucial role in understanding user preferences and interests. Utilizing the demographic information available in the dataset, including gender and age, we explore the browsing category interests among different demographic segments. By visualizing the data and conducting statistical analyses, we uncover potential variations or preferences in browsing behavior based on gender and age groups. This analysis provides valuable insights into the browsing category preferences and interests across different demographic segments.

5.1.1 Dataset Overview

As shown in table 5.1 The most common query type is "A" with a count of 6,534,737, representing approximately 97.311% of the total queries. "AAAA" queries, used for IPv6 address retrieval, have a count of 134,760, accounting for about 2.007% of the queries. "PTR" queries, for reverse DNS lookups, have a count of 17,024, making up 0.254% of the total queries. "TXT" queries, used for retrieving text records, have a count of 15,011, representing 0.224% of the queries. Less frequent query types include "SVCB," "SRV," "NAPTR," "SOA," "CNAME," "NS," and "CERT" with counts ranging from 3 to 8,676. These insights highlight the dominance of "A" and "AAAA" queries, indicating a significant number of address resolution requests.

Table 5.1: DNS Type Count

Type	**Count**	**Percentage (%)**
A	6534737	97.311
AAAA	134760	2.0068
PTR	17024	0.2535
TXT	15011	0.2235
SVCB	8676	0.1292
SRV	3266	0.0486
NAPTR	1161	0.0173
SOA	259	0.0039
CNAME	214	0.0032
NS	198	0.0029
CERT	3	0

According to the demographic analysis of the dataset, it is observed that 47.8% of the total users are male, while 52.2% of the total users are female. Analyzing the age distribution, it is found that users within the age group of 25 to 35 years contributed the most to the log data. For user engagement by age group, Figure 5.1 showcases the activity levels of different age groups. Among these age groups, the most active user group is the 25-29 years category, with 18% being female and 20% being male. Additionally, the age group of 30-35 years also exhibits significant engagement, with 17% being female and 7% being male.

5.1.2 User Engagement Analysis Using Demographic Data

Figure 5.2 shows the analysis of weekday-wise gender engagement reveals interesting patterns in user behavior. On Sundays, the engagement is slightly skewed towards males, with 55.4% of the users being male and 44.6% being female. Mondays show a more balanced distribution, with a slightly higher male engagement of 50.5% compared to 49.5% female engagement. Tuesdays exhibit a similar trend, with 52.3%

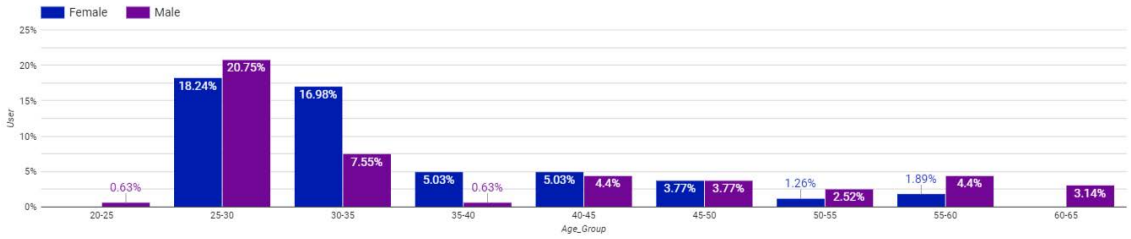

Figure 5.1: Gender Ratio By Age Group

male engagement and 47.7% female engagement. Wednesdays present an intriguing pattern, as the gender engagement is evenly split, with both males and females accounting for 50% of the users. Thursdays, on the other hand, showcase a slight shift towards male engagement, with 52.2% male users and 47.8% female users. These findings suggest that gender engagement varies across different weekdays, highlighting potential patterns in user behavior. Understanding such variations can aid in tailoring marketing strategies and content to specific days of the week, ensuring effective targeting and engagement with the desired audience segments.

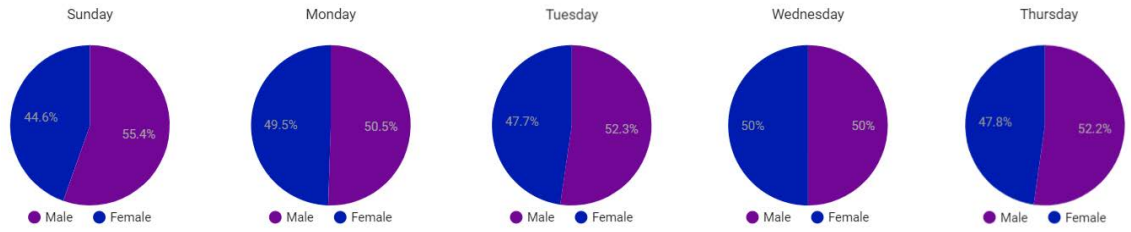

Figure 5.2: Gender Engagement By Weekday

Figure 5.3 and 5.4 shows the analysis of weekday-wise age group engagement provides insights into the distribution of user engagement across different age groups on specific weekdays. On Mondays, the age group of 25-30 shows the highest engagement, representing 10.44% of the total records, followed by the age groups of 30-35 and 40-45 with 6.25% and 2.33% respectively. Sundays display a different pattern, with the age group of 30-35 leading in engagement at 1.93%, followed by 35-40 and 45-50 at 0.61% and 0.68% respectively. Tuesdays demonstrate similar trends, with the age groups of 25-30 and 30-35 exhibiting the highest engagement rates at 9.38% and 4.83% respectively. Thursdays highlight the age group of 25-30 as the most engaged, representing 6.30% of the records, followed by 30-35 and 45-50 with 4.72% and 1.47% respectively. Wednesdays exhibit a similar pattern, with the age groups of 25-30 and 30-35 leading in engagement at 9.55% and 5.32% respectively. These findings suggest that different age groups show varying levels of engagement on different weekdays. Understanding these patterns can help tailor marketing strategies and content to specific age groups and weekdays, maximizing user engagement and overall campaign effectiveness.

The line chart data in figure 5.5 illustrates the engagement patterns of males and females across different office hours of each weekday.

On Sundays, both males and females show a steady increase in engagement from 8 am to 12 pm, with a peak around 12 pm for males and 9 am for females. After reaching their respective peaks, engagement gradually declines for both genders

	Monday		Wednesday		Tuesday		Thursday		Sunday	
Age_Group	Record Count	Record Count (%)	Record Count	Record Count (%)	Record Count	Record Count (%)	Record Count	Record Count (%)	Record Count	Record Count (%)
25-30	347,188	10.45%	317,457	9.55%	311,815	9.38%	209,314	6.3%	146,938	4.42%
30-35	207,667	6.25%	176,978	5.32%	160,424	4.83%	156,972	4.72%	64,099	1.93%
40-45	77,358	2.33%	72,336	2.18%	58,302	1.75%	38,151	1.15%	22,338	0.67%
45-50	56,422	1.7%	57,578	1.73%	37,206	1.12%	48,706	1.47%	22,540	0.68%
35-40	55,905	1.68%	59,737	1.8%	54,205	1.63%	17,569	0.53%	20,160	0.61%
50-55	39,973	1.2%	52,978	1.59%	51,228	1.54%	43,680	1.31%	9,006	0.27%
60-65	31,943	0.96%	38,078	1.15%	45,273	1.36%	35,267	1.06%	12,553	0.38%
55-60	41,014	1.23%	32,977	0.99%	20,971	0.63%	32,384	0.97%	21,073	0.63%
20-25	3,707	0.11%	6,416	0.19%	3,974	0.12%	4,094	0.12%	-	-

Figure 5.3: Age Group Engagement By Weekday

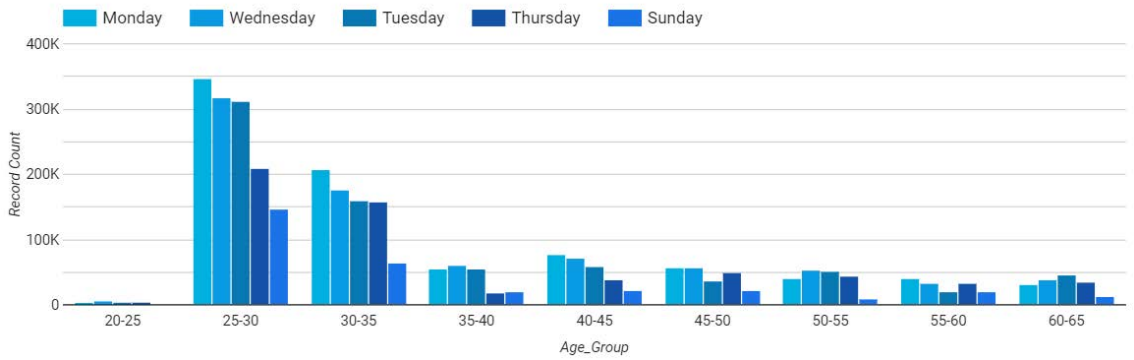

Figure 5.4: Age Group Engagement By Weekday Bar Chart

throughout the afternoon, with intermittent fluctuations. Mondays exhibit a similar trend, with males and females demonstrating higher engagement during the morning hours, peaking around 9 am for females and 10 am for males. Engagement remains relatively stable until early afternoon and then gradually decreases towards the end of the day. Tuesdays follow a comparable pattern, with both genders displaying higher engagement during the morning hours, peaking around 9 am for females and 10 am for males. Engagement remains relatively consistent throughout the day, with a slight dip during the late afternoon. Wednesdays showcase a similar trend, with peak engagement occurring during the morning hours for both males and females. The highest engagement levels are observed around 9 am for females and 10 am for males. Engagement remains relatively stable until early afternoon, followed by a gradual decline towards the end of the day. Thursdays exhibit a pattern akin to the previous weekdays, with higher engagement during the morning hours and peaks around 9 am for females and 10 am for males. Engagement remains relatively steady throughout the day, with a slight decline in the late afternoon.

Overall, the data suggests that mornings are generally the most active period for engagement across all weekdays, with a slightly higher engagement observed for males during peak hours.

The bar chart data in figure 5.6 represents the hourly engagement of office staff from 8 am to 5 pm.

The highest level of engagement is observed during the hours of 10 am to 11 am, with a record count of 885,730. This indicates that the majority of office staff are actively engaged during this time, potentially focusing on important tasks and meetings. Following closely behind, the hours of 11 am to 12 pm and 12 pm to 1 pm also show substantial engagement, with record counts of 837,502 and 842,846 respectively. These periods likely correspond to mid-morning and lunchtime activities, where employees are typically involved in various work-related activities or taking a break.

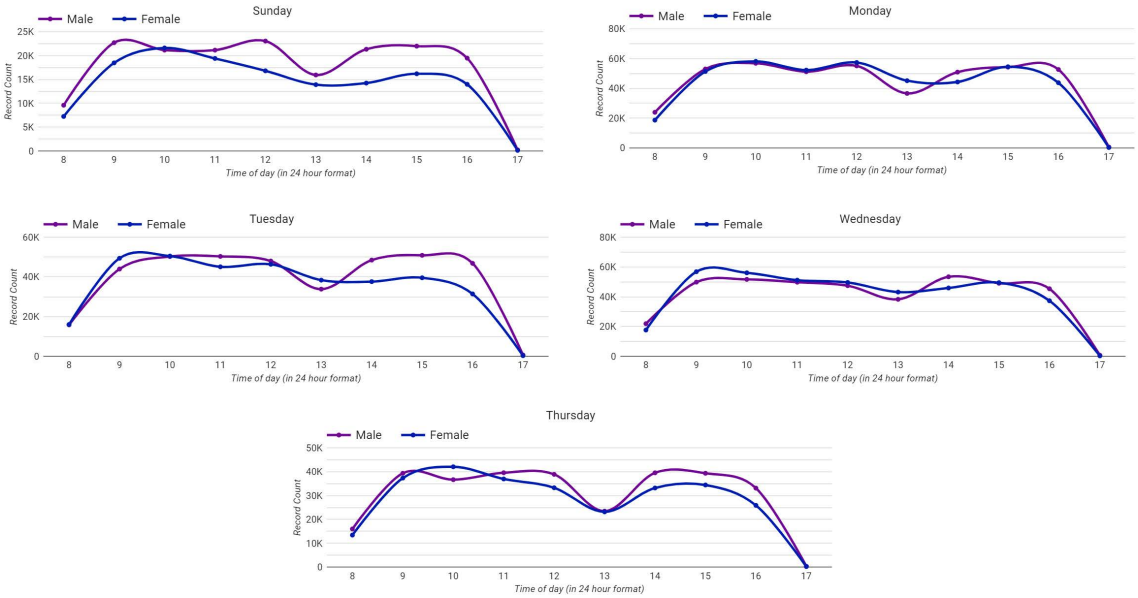

Figure 5.5: Gender Engagement By Office Hour of Each Weekday

The early morning hours of 8 am to 9 am exhibit relatively high engagement as well, with a record count of 318,779. This suggests that a significant number of employees are present and actively engaged right at the start of the workday, possibly focusing on planning, organizing, and addressing urgent tasks. During the afternoon hours, engagement remains consistently high but slightly lower compared to the morning hours. The hours of 1 pm to 2 pm and 3 pm to 4 pm show record counts of 650,497 and 788,261 respectively, indicating a continued level of productivity and involvement in work-related activities during this period. The data indicates a slight dip in engagement during the last hour of work, from 4 pm to 5 pm, with a record count of 749,203. This could be attributed to winding down activities, wrapping up tasks, or preparing for the end of the workday.

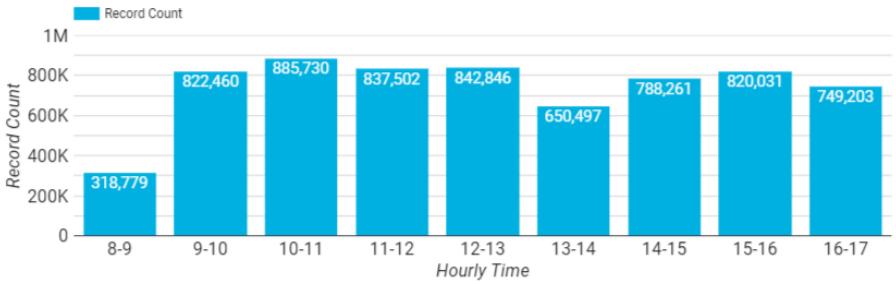

Figure 5.6: Hourly Engagement Count By Office Hour

Overall, the data suggests that the peak hours of engagement for office staff fall between 10 am and 12 pm, aligning with typical mid-morning productivity. Understanding these engagement patterns can help organizations optimize scheduling, allocate resources effectively, and ensure efficient workflow management during the most active periods of the workday.

The bar chart data in figure 5.7 represents the gender distribution of office staff engagement during the first half and second half of the workday.

During the first half of the workday, there is a relatively balanced gender distri-

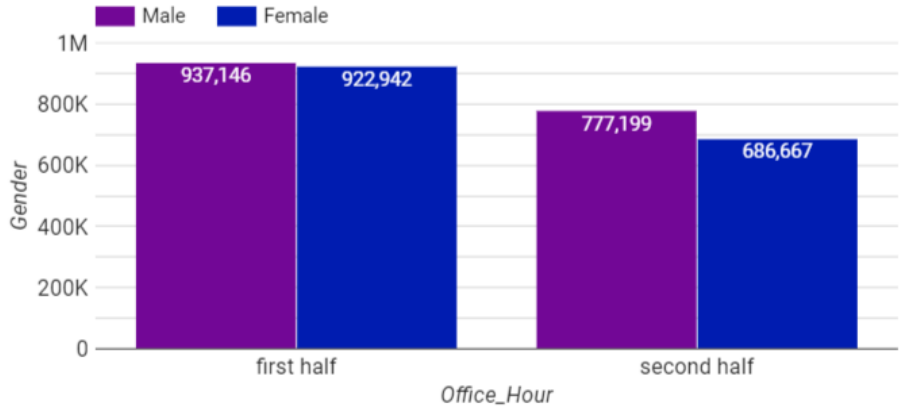

Figure 5.7: Hourly Engagement Count By Office Hour

bution in office staff engagement. The record count for males is 937,146, while for females it is slightly lower at 922,942. This indicates that both male and female employees are actively involved in work-related activities during the initial hours of the day.In the second half of the workday, the gender distribution shows a slight shift. The record count for males decreases to 777,199, while for females it further reduces to 686,667. This suggests that there may be a relatively higher level of male engagement during the later hours of the workday compared to females.Overall, the data indicates that there is a relatively balanced gender distribution during the first half of the workday, but a slight shift towards more male engagement in the second half. These findings may have implications for understanding gender-specific work patterns, preferences, or responsibilities within the office environment. It is important to note that further analysis and contextual information would be necessary to fully interpret the reasons behind these gender distributions and their potential impact on productivity, collaboration, and overall workplace dynamics.
The bar chart data in figure 5.8 presents the record count of different age groups during the first half and second half of the workday. Let's analyze the information:

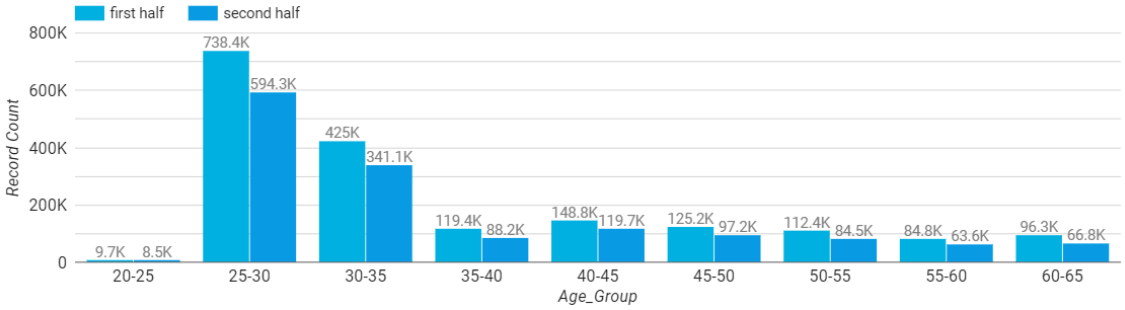

Figure 5.8: Hourly Engagement Count By Office Hour

During the first half of the workday, the age group with the highest record count is 25-30, with a substantial engagement of 738,411. It is followed by the age group 40-45 with 148,760 records, and 45-50 with 125,240 records. Other age groups, such as 20-25, 30-35, 35-40, 50-55, 55-60, and 60-65, also show varying levels of engagement during this period.In the second half of the workday, the age group 25-30 still maintains a significant record count of 594,301, although slightly lower compared to the first half. The age groups 30-35 and 40-45 also demonstrate notable

engagement with record counts of 341,091 and 119,725, respectively. Other age groups, such as 45-50, 50-55, 55-60, and 60-65, exhibit lower but still considerable levels of engagement during this time. Overall, the data suggests that the age group 25-30 consistently shows high engagement during both the first and second halves of the workday. The age groups 30-35 and 40-45 also exhibit considerable involvement in their respective office hours. It is worth noting that the record counts for each age group vary between the first and second halves, indicating potential differences in work patterns, priorities, or responsibilities across age groups.
The bar chart data in figure 5.9 presents the record count during different office hours for each weekday. Let's analyze the information:

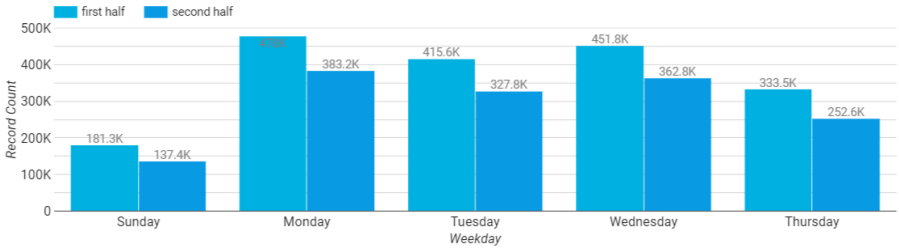

Figure 5.9: Hourly Engagement Count By Office Hour

On Sunday, there is a higher record count during the first half of the day, with 181,262 records, compared to the second half, which has 137,445 records. This indicates that there is relatively more engagement and activity during the morning and early afternoon hours on Sundays. For Monday, there is a similar trend where the first half of the day shows a higher record count of 477,972 compared to the second half, which has 383,205 records. This suggests that Mondays are generally more productive and have increased engagement during the earlier part of the workday. On Tuesday, the first half of the day also demonstrates a higher record count of 415,573 compared to the second half, which has 327,825 records. Similar to Monday, Tuesdays exhibit greater engagement during the earlier office hours. Wednesday follows a similar pattern, with the first half of the day showing a higher record count of 451,764 compared to the second half, which has 362,771 records. Wednesdays tend to have higher productivity and engagement during the morning and early afternoon. Thursday displays a slightly different pattern compared to the previous weekdays. In this case, the second half of the day shows a slightly lower record count of 252,620 compared to the first half, which has 333,517 records. This indicates a potential shift in productivity or engagement levels during the latter part of Thursdays, with a relatively more active first half. Overall, the data suggests variations in engagement and productivity levels across different weekdays and office hours. Mondays, Tuesdays, and Wednesdays generally exhibit higher engagement during the first half of the day, while Thursdays show a potential shift in activity during the second half. Further analysis and contextual information would be required to understand the reasons behind these patterns and their implications for work schedules, employee performance, and organizational dynamics.

5.1.3 User Browsing Analysis Using Demographic Data

The table 5.10 represents domain records and their respective categories, along with the corresponding record counts. The data showcases a sample of 20 domain records

out of a total of 67,807.

	Domain	Category	Record Count ▾	Record Cout %
1.	www.google.com	Search Engines	216,030	3.22%
2.	ssl.gstatic.com	CDN	215,218	3.2%
3.	signaler-pa.clients6.google.com	Search Engines	185,627	2.76%
4.	play.google.com	Game	136,455	2.03%
5.	graph.facebook.com	Social Network	117,906	1.76%
6.	beacons.gcp.gvt2.com	Computer/Technol...	106,391	1.58%
7.	mail.google.com	Webmail	95,878	1.43%
8.	safebrowsing.googleapis.com	Computer/Technol...	88,998	1.33%
9.	google.com	Search Engines	85,118	1.27%
10.	clients4.google.com	Search Engines	57,142	0.85%
11.	www.googleapis.com	CDN	51,724	0.77%
12.	mtalk.google.com	Chat	51,284	0.76%
13.	v10.events.data.microsoft.com	Computer/Technol...	50,532	0.75%
14.	www.gstatic.com	CDN	48,528	0.72%
15.	play.googleapis.com	Download	47,878	0.71%
16.	d27xxe7juh1us6.cloudfront.net	CDN	47,283	0.7%
17.	clients6.google.com	Search Engines	46,723	0.7%
18.	addons-pa.clients6.google.com	Search Engines	45,844	0.68%
19.	femetrics.grammarly.io	Religion	45,564	0.68%
20.	peoplestack-pa.clients6.google.c...	Search Engines	45,311	0.67%

Figure 5.10: Hourly Engagement Count By Office Hour

Analyzing this data provides valuable insights into the distribution and significance of different domains and categories. Upon examining the dataset, it becomes evident that certain domains stand out in terms of record counts. For instance, www.google.com and ssl.gstatic.com have the highest record counts, indicating their substantial presence in the dataset. These domains are closely followed by signaler-pa.clients6.google.com and play.google.com which also exhibit notable record counts. When considering the categories, it is interesting to observe that "Search Engines" and "Computer/Technology" dominate the dataset. The "Search Engines" category encompasses several domains such as www.google.com, signaler-pa.clients6.google.com and google.com highlighting the prevalence of search engine-related activities. The "Computer/Technology" category includes domains like beacons.gcp.gvt2.com, safebrowsing.googleapis.com and v10.events.data.microsoft.com indicating the significance of computer and technology-related domains in the dataset. Examining the record count percentages provides further insights into the dataset. By converting the record count column into percentages, it becomes evident that "www.google.com" has the highest record count percentage, indicating its dominant presence. Similarly, the "Search Engines" category exhibits the highest record count percentage, emphasizing its significant contribution to the overall records. It is worth noting that certain domains, such as "femetrics.grammarly.io" and "addons-pa.clients6.google.com," have relatively lower record count percentages. This suggests their relatively smaller presence in the dataset compared to other domains.

The pivot table 5.11 data showcases various categories of websites along with their corresponding record counts for male and female users.

The analysis of the pivot table data reveals that both males and females exhibit a

	Gender / Record Count / Record Count %			
	Male		Female	
Category	Record Count	Record Count %	Record Count	Record Count %
Computer/Technology	467,057	14.051%	449,974	13.537%
Search Engines	303,850	9.141%	321,407	9.669%
CDN	247,648	7.450%	222,644	6.698%
Ads	167,740	5.046%	126,960	3.820%
Business/Service	141,542	4.258%	119,198	3.586%
Game	50,851	1.530%	61,169	1.840%
Webmail	45,665	1.374%	51,599	1.552%
Learning	48,303	1.453%	30,655	0.922%
Social Network	40,979	1.233%	28,913	0.870%
Chat	33,094	0.996%	31,531	0.949%
Unclassified	17,134	0.516%	39,393	1.185%
Religion	26,148	0.787%	19,169	0.577%
Entertainment	19,378	0.583%	18,405	0.554%
Shopping	15,750	0.474%	16,759	0.504%
File Hosting	8,932	0.269%	11,959	0.360%
Tracker	9,622	0.290%	8,058	0.242%
News/Magazine	9,164	0.276%	5,707	0.172%
Download	7,290	0.219%	5,746	0.173%
Misc	5,630	0.169%	5,274	0.159%

Figure 5.11: Domain Category By Gender Engagement

strong preference for certain categories. For males, the top categories include computer/technology with a record count of 467,057 (14.05%), followed by search engines with 303,850 (9.14%), CDN (Content Delivery Network) with 247,648 (7.45%), ads with 167,740 (5.05%), and business/service with 141,542 (4.26%). Similarly, females also display a similar pattern, with computer/technology being the most popular category at 449,974 (13.54%), followed by search engines with 321,407 (9.67%), CDN with 222,644 (6.70%), ads with 126,960 (3.82%), and business/service with 119,198 (3.59%). These findings indicate a shared interest in technology and information retrieval among both genders. Additionally, categories such as CDN, ads, and business/service are also prominent for both males and females, suggesting a general attraction towards content delivery, online advertisements, and various business-related services. Recognizing these common preferences can assist businesses and content providers in tailoring their offerings to better align with the specific interests and needs of male and female users. From the analysis, it is evident that both males and females show a high preference for categories related to computer/technology and search engines. These categories dominate the top positions for both genders, indicating a shared interest in technology and information retrieval. Additionally, categories like CDN, ads, and business/service also feature prominently in the top categories for both males and females, suggesting a general interest in content delivery, online advertisements, and various business-related services. This analysis highlights the similarities in category preferences between males and females, with computer/technology and search engines being the most popular across both genders. Understanding these top categories can help businesses and content providers tailor their offerings to cater to the specific interests and needs of male and female users.

The pivot table 5.12 data shows the record count percentages for different categories and age groups.

Each category represents a type of website or online service, and the age groups are 20-25, 25-30, 30-35, 35-40, and 40-45. The record count percentages indicate

	Age_Group / Record Count (%)								
Category	25-30	30-35	40-45	45-50	35-40	50-55	60-65	55-60	20-25
Computer/Tech...	10.94%	6.71%	2.25%	1.92%	1.79%	1.25%	1.15%	1.43%	0.16%
Search Engines	7.83%	4.58%	1.52%	1.15%	1.18%	0.84%	0.67%	0.92%	0.13%
CDN	5.92%	3.38%	1.2%	0.91%	0.9%	0.57%	0.54%	0.65%	0.07%
Ads	3.09%	1.92%	0.76%	0.78%	0.38%	0.33%	1.19%	0.39%	0.04%
Business/Servi...	3.36%	1.7%	0.74%	0.47%	0.49%	0.37%	0.38%	0.3%	0.03%
Game	1.25%	0.72%	0.26%	0.24%	0.23%	0.39%	0.13%	0.13%	0.02%
Webmail	1.07%	0.63%	0.28%	0.26%	0.23%	0.23%	0.08%	0.12%	0.02%
Learning	1.28%	0.51%	0.1%	0.17%	0.12%	0.02%	0.11%	0.08%	-
Social Network	0.84%	0.5%	0.12%	0.06%	0.15%	0.25%	0.1%	0.07%	0.02%
Chat	0.68%	0.44%	0.17%	0.26%	0.17%	0.07%	0.05%	0.09%	0.01%
Unclassified	0.4%	0.11%	0.09%	0.02%	0.06%	0.94%	0.07%	0.01%	+0%

Figure 5.12: Domain Category By Age Group Engagement

the relative frequency of each category within each age group. For example, in the age group 20-25, the category with the highest record count percentage is "Computer/Technology" with 0.1607%. This suggests that a significant proportion of websites or online services visited by individuals in the 20-25 age group fall under the "Computer/Technology" category. Similarly, "Search Engines" and "CDN" are also popular categories in this age group. In the age group 25-30, the categories "Search Engines" and "Computer/Technology" have even higher record count percentages of 0.1301% and 0.1094% respectively, indicating their continued popularity. "CDN" and "Webmail" are also among the more frequented categories. The record count percentages vary across age groups and categories, reflecting the preferences and interests of different age groups when it comes to online activities. The data can be useful for analyzing trends in website usage and understanding the browsing habits of different age groups.
Figure 5.13 contains information about the distribution of students' age groups and the number of records collected from different websites and provides data on the gender distribution.

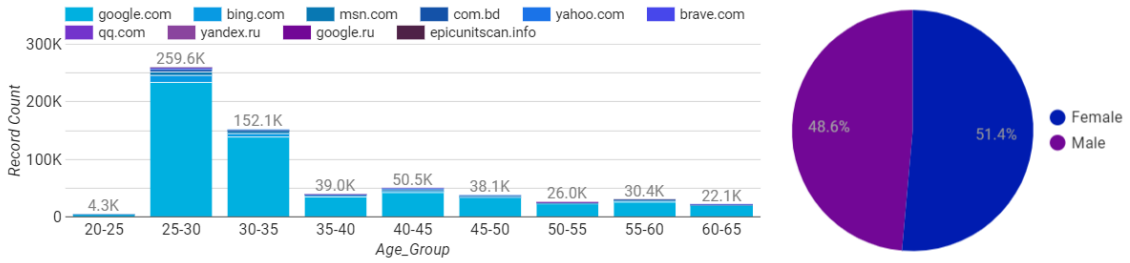

Figure 5.13: Search Engine Engagement By Age Group And Gender

The age groups range from 20 to 65, with each group representing a five-year range (e.g., 20-25, 25-30, 30-35, etc.). The data appears to be collected from an educational or research survey, as it focuses on students' age distribution. Notably, the table's age distribution seems to be weighted towards younger individuals, with a higher number of records for the 20-25 and 25-30 age groups compared to older groups. Multiple websites are included in the data, such as Google, Bing, Yahoo, etc. The "google.com" website has the most significant number of records across several age groups, suggesting that it is the most popular or widely used search engine among students of different ages. The data indicates that the population surveyed is relatively balanced in terms of gender, with approximately 51.4% identified as female and 48.6% as male.
In figure 5.14 we have information about the social media engagement of different

age groups and gender.

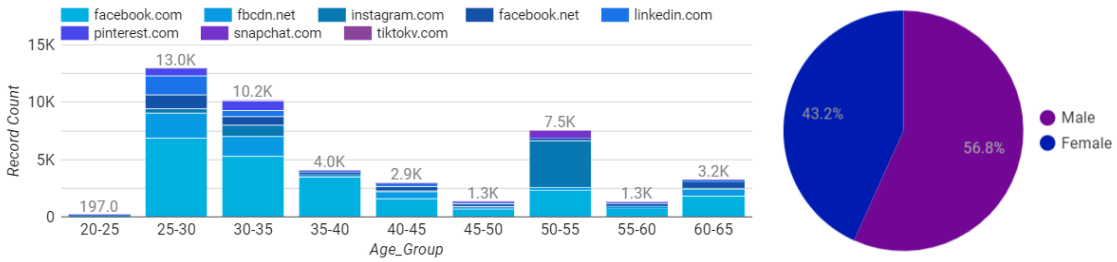

Figure 5.14: Social Media Engagement By Age Group And Gender

The age groups range from 20 to 65, with each group representing a five-year range. The data indicates the number of records collected for various social media platforms, such as Facebook, Instagram, Pinterest, Snapchat, LinkedIn, and TikTok. Among the age groups, individuals between 25 and 35 appear to have the highest social media engagement, as reflected by the larger number of records across multiple platforms. Facebook, with its different domain extensions like ".com," ".net," and ".fbcdn.net," consistently has a substantial number of records across most age groups, indicating its popularity across different demographics. Instagram also shows significant engagement, particularly among individuals aged 25 to 60, as reflected by the higher record counts. LinkedIn seems to have relatively consistent engagement across various age groups, while Snapchat and Pinterest show lower engagement overall. The bar chart presents the gender distribution within the surveyed population. The data indicates that approximately 56.8% of the surveyed population identifies as male, while 43.2% identifies as female. In summary, the provided tables offer insights into social media engagement patterns across different age groups and the gender distribution within the surveyed population. The data suggests that individuals between 25 and 35 tend to have higher social media engagement, and platforms like Facebook and Instagram are popular across multiple age groups. However, it's important to note that without additional context about the survey's purpose and methodology, these observations should be interpreted with caution.
In figure 5.15, we have information about the YouTube engagement of different age groups.

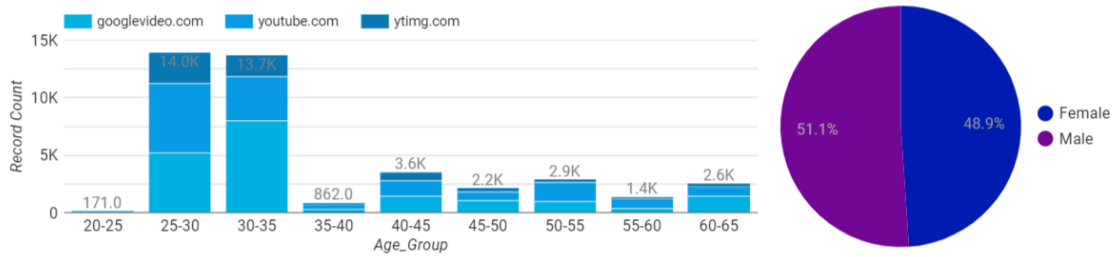

Figure 5.15: Youtube Engagement By Age Group And Gender

The age groups range from 20 to 65, with each group representing a five-year range. The data indicates the number of records collected for various components of YouTube, such as the main website (youtube.com), image server (ytimg.com), and

video server (googlevideo.com). Among the age groups, individuals between 25 and 35 appear to have the highest YouTube engagement, as reflected by the larger number of records across all three components. This suggests that individuals within this age range are actively using YouTube for watching videos. The engagement remains relatively high for individuals between 20 and 45, with noticeable engagement from individuals up to 60 years old. When looking at the components, the main YouTube website (youtube.com) consistently shows higher engagement across all age groups, followed by the image server (ytimg.com) and video server (googlevideo.com). The bar chart prepresents gender distribution within the surveyed population for YouTube engagement. The data indicates that approximately 51.1% of the surveyed population identifies as male, while 48.9% identifies as female. In summary, the provided tables offer insights into YouTube engagement patterns across different age groups and the gender distribution within the surveyed population. The data suggests that individuals between 25 and 35 have the highest YouTube engagement, with the main YouTube website (youtube.com) being the most commonly accessed component. Additionally, the gender distribution shows a relatively balanced representation, with a slightly higher proportion of males.
In figure 5.16, we have information about the ChatGPT engagement of different age groups.

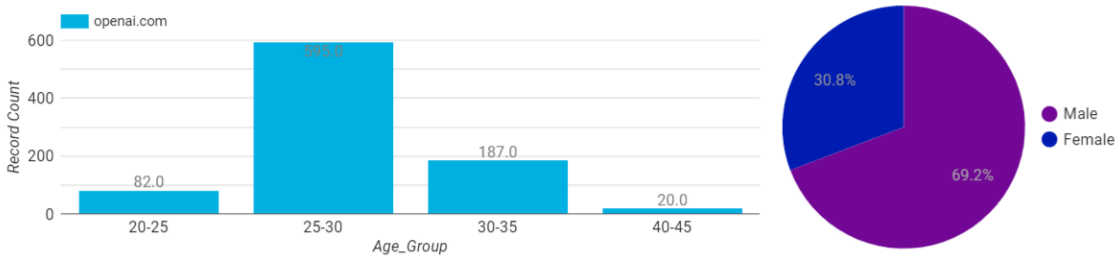

Figure 5.16: ChatGPT Engagement By Age Group And Gender

The age groups range from 20 to 45, with each group representing a five-year range. The data indicates the number of records collected for the website openai.com, which represents engagement with ChatGPT. Among the age groups, individuals between 25 and 30 appear to have the highest ChatGPT engagement, with a significant number of records collected. This suggests that individuals within this age range are actively engaging with ChatGPT on the openai.com website. The engagement remains relatively high for individuals between 20 and 35, with a smaller number of records for individuals between 40 and 45. The pie chart represents the gender distribution within the surveyed population for ChatGPT engagement. The data indicates that approximately 69.2% of the surveyed population identifies as male, while 30.8% identifies as female. In summary, the provided tables offer insights into ChatGPT engagement patterns across different age groups and the gender distribution within the surveyed population. The data suggests that individuals between 25 and 30 have the highest ChatGPT engagement, with the openai.com website being the primary platform for interaction. Additionally, the gender distribution shows a higher proportion of males compared to females.
In figure 5.17, we have information about the engagement of different age groups with online newspapers.
The age groups range from 20 to 65, with each group representing a five-year range.

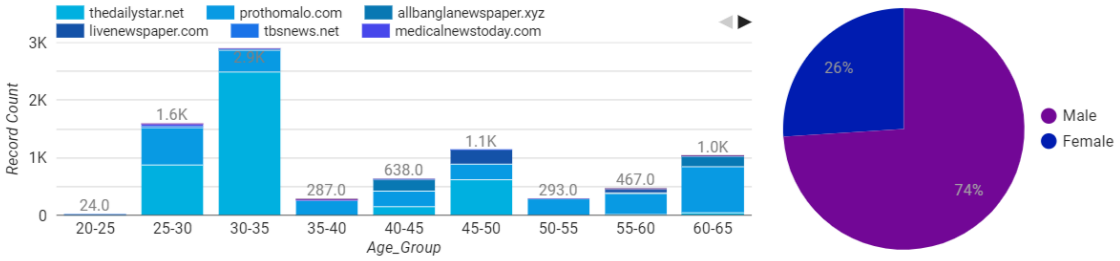

Figure 5.17: Online News Platform Engagement By Age Group And Gender

The data indicates the number of records collected for various websites, representing engagement with online newspapers. Among the age groups, individuals between 30 and 35 appear to have the highest engagement with online newspapers, particularly with websites such as prothomalo.com and thedailystar.net, which have a significant number of records. The engagement remains relatively high for individuals between 40 and 60, with varying levels of engagement across different websites. Table 2 presents the gender distribution within the surveyed population for online newspaper engagement. The data indicates that approximately 73.9% of the surveyed population identifies as male, while 26.0% identifies as female. In summary, the provided tables offer insights into the engagement patterns of different age groups with online newspapers and the gender distribution within the surveyed population. The data suggests that individuals between 30 and 35 have the highest engagement with online newspapers, primarily with websites like prothomalo.com and thedailystar.net. Additionally, the gender distribution shows a higher proportion of males compared to females in terms of online newspaper engagement.
In figure 5.18, we have information about the engagement of different age groups with CDN (Content Delivery Network) services.

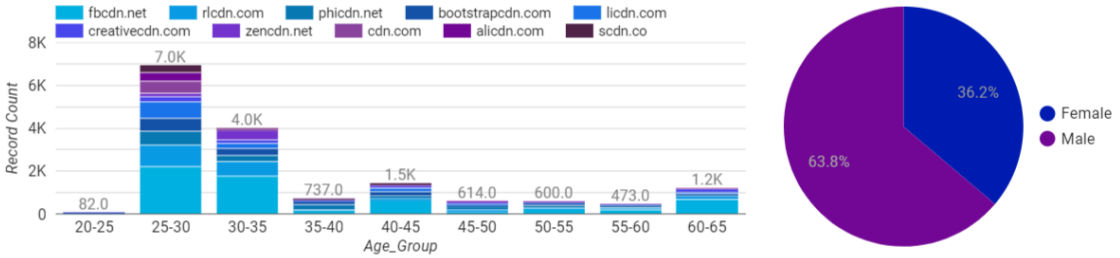

Figure 5.18: CDN Engagement By Age Group And Gender

The age groups range from 20 to 65, with each group representing a five-year range. The data indicates the number of records collected for various CDN service websites, representing their engagement with these services. Among the age groups, individuals between 25 and 35 show the highest engagement with CDN services, particularly with websites like bootstrapcdn.com, fbcdn.net, and rlcdn.com, which have a significant number of records. The engagement remains relatively high for individuals between 35 and 60, with varying levels of engagement across different CDN service websites. The bar chart presents the gender distribution within the surveyed population for CDN service engagement. The data indicates that approximately 63.8% of the surveyed population identifies as male, while 36.2% identifies as female. In

summary, the provided tables offer insights into the engagement patterns of different age groups with CDN services and the gender distribution within the surveyed population. The data suggests that individuals between 25 and 35 have the highest engagement with CDN services, primarily with websites like bootstrapcdn.com and fbcdn.net. Additionally, the gender distribution shows a higher proportion of males compared to females in terms of CDN service engagement.
In figure 5.19, we have information about the engagement of different age groups with Torrent services.

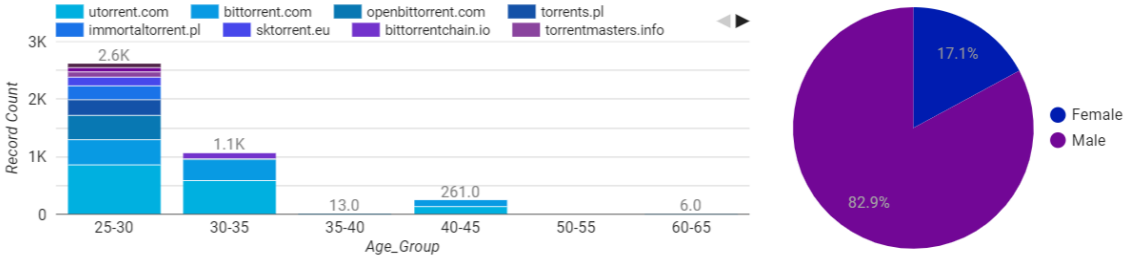

Figure 5.19: Torrent Engagement By Age Group And Gender

The age groups range from 25 to 65, with each group representing a five-year range. The data indicates the number of records collected for various Torrent service websites, representing their engagement with these services. Among the age groups, individuals between 25 and 30 show the highest engagement with Torrent services, particularly with websites like utorrent.com, openbittorrent.com, and bittorrent.com, which have a significant number of records. This age group also demonstrates engagement with other Torrent service websites, such as torrentmasters.info, immortaltorrent.pl, and torrentsmd.com. The engagement decreases as the age groups progress, with limited records for individuals between 30 and 40, 40 and 45, and 60 and 65. The bar chart presents the gender distribution within the surveyed population for Torrent service engagement. The data indicates that approximately 82.87% of the surveyed population identifies as male, while 17.13% identifies as female. In summary, the provided tables offer insights into the engagement patterns of different age groups with Torrent services and the gender distribution within the surveyed population. The data suggests that individuals between 25 and 30 have the highest engagement with Torrent services, primarily with websites like utorrent.com and openbittorrent.com. Additionally, the gender distribution shows a higher proportion of males compared to females in terms of Torrent service engagement.
The collected data provides insights into various digital service engagements, including search engines, social media, YouTube, ChatGPT, online news, CDN services, and Torrent services. The age distribution in the data indicates a focus on younger individuals, with a higher number of records for the 20-25 and 25-30 age groups. Google appears to be the most popular search engine among students of different ages. Social media engagement is highest among individuals aged 25-35, with platforms like Facebook and Instagram being popular across multiple age groups. YouTube engagement is also highest among individuals aged 25-35, with the main YouTube website being the most accessed. ChatGPT engagement shows a peak in the 25-30 age group, primarily on the openai.com platform. Online news engagement is highest in the 30-35 age group, with websites like prothomalo.com and thedailystar.net being prominent. CDN service engagement is most notable in the 25-35

age group, with websites like bootstrapcdn.com and fbcdn.net being commonly accessed. Torrent service engagement peaks in the 25-30 age group, with utorrent.com and openbittorrent.com being popular choices. The gender distribution across all engagements is relatively balanced, with a slightly higher proportion of males. Overall, these findings provide valuable insights into the digital service preferences and usage patterns of different age groups and genders within the surveyed population.

5.2 Analysis and Results of Machine Learning Models

In this section, we present the analysis and results of the various supervised and unsupervised machine learning models employed for domain category prediction. We conducted a comprehensive evaluation of these models to determine their effectiveness in accurately classifying website categories. For the supervised learning models, including decision trees, random forests, logistic regression, and support vector machines, we trained the models on the preprocessed dataset and evaluated their performance using appropriate evaluation metrics such as accuracy, precision, recall, and F1 score [51]. We also employed techniques like cross-validation to assess the models' generalization capabilities and mitigate overfitting. We explored unsupervised learning techniques such as clustering algorithms, including k-means clustering and hierarchical clustering, to identify patterns and group similar domains together based on their features. We evaluated the quality of the clusters using metrics such as silhouette score and visual inspection of cluster assignments. The analysis and results of the machine learning models provide valuable insights into the predictive power and performance of each model for domain category prediction. By comparing the performance metrics and examining the models' strengths and weaknesses, we gain a comprehensive understanding of their suitability for our research objectives. Overall, the analysis and results chapter sheds light on both the visual analysis conducted to gain insights into the dataset and the performance of various machine learning models in predicting website categories. The findings from this chapter serve as a foundation for the subsequent discussion and conclusion, contributing to our understanding of website categorization and its implications in the context of our research.

5.2.1 Supervised Learning Models

Random Forest classifier

The feature analysis from the Random Forest classifier model shown in figure 5.20 indicates the relative importance of different features in URL classification. The "Domain" feature has the highest importance with a score of 34.63%, followed by "Entropy" at 25.94%. The "URL_Length" feature holds a significance of 17.62%, while the presence of numerical digits ("Number_Count") contributes 6.71%. The impact of special characters ("Special_Char_Count") is relatively lower at 4.58%. Features related to the structure of the URL, such as subdomain count ("SubdomainNumber"), dot count ("DotNumber"), hyphen count ("HypernNumber"), and query parameter count ("QueryParamNumber"), have relatively lower importance scores ranging from 3.04% to 3.79%.

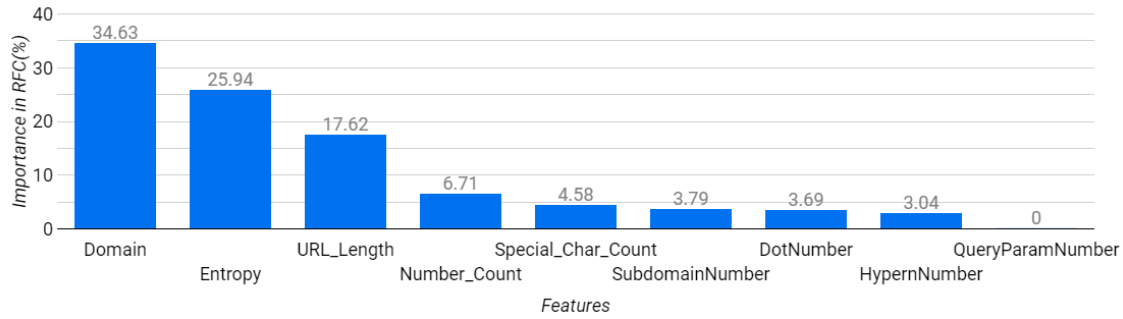

Figure 5.20: Feature Importance (Random Forest Classifier)

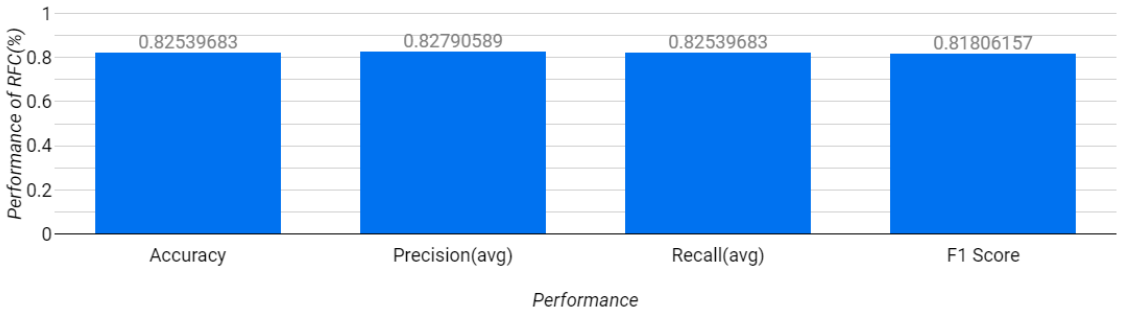

Figure 5.21: Performance (Random Forest Classifier)

The bar chart data shown in figure 5.21 represents the performance metrics of a Random Forest classifier model. The model achieves an overall accuracy of 82.54%, indicating its ability to correctly classify instances. The average precision score is 82.79%, which measures the model's ability to make precise predictions. The average recall score is also 82.54%, indicating the model's capability to accurately identify positive instances. The F1 score, which considers both precision and recall, is 81.81%, demonstrating a balance between precision and recall. The log loss value of 1.74 suggests the model's degree of uncertainty in its predictions, with lower values indicating more confident predictions. Overall, these metrics suggest that the Random Forest classifier performs well in this particular classification task, achieving a relatively high level of accuracy and a good balance between precision and recall.

Naive Bayes Model

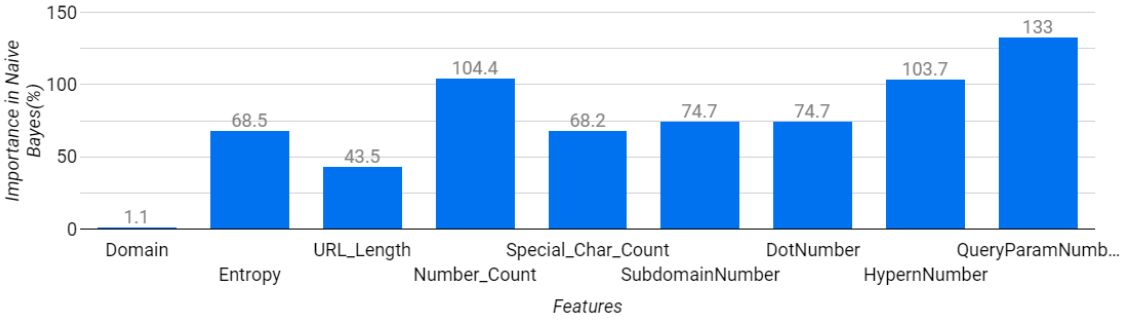

Figure 5.22: Feature Importance (Naive Bayes Model)

The feature importance values shown in figure 5.22 for the Naive Bayes model indicate the following percentages of importance: "Domain" has low importance (1.09%), "Entropy" has high importance (68.52%), "URL_Length" has moderate importance (43.49%), "Number_Count" has high importance (104.36%), "Special Character Count" has high importance (68.15%), "SubdomainNumber," "DotNumber," and "HypernNumber" have similar importance (around 74.69%), and "QueryParamNumber" has the highest importance (132.98%). These percentages suggest that features like "Entropy," "Number_Count," "Special_Char_Count," and "QueryParamNumber" are influential in distinguishing legitimate and suspicious domains. It is shown in figure 5.23 that Naive Bayes model achieved an accuracy of 0.32,

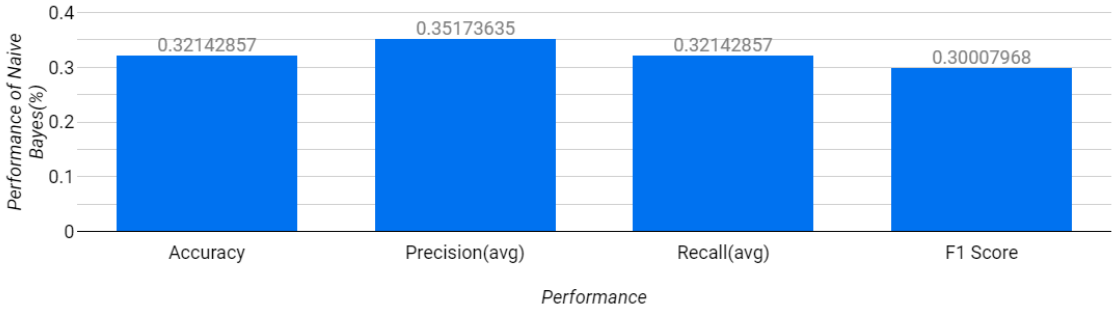

Figure 5.23: Performance (Naive Bayes Model)

indicating that its predictions were correct approximately 32% of the time. The average precision was 0.35, suggesting that around 35% of the positive predictions were accurate. The average recall was 0.32, meaning that the model identified approximately 32% of the actual positive instances. The F1 score, a measure of overall model performance, was 0.30, indicating a balance between precision and recall. These metrics suggest that the model's performance can be improved, and further analysis is needed to enhance its effectiveness.

Support Vector Machines (SVM)

Naive Bayes Model

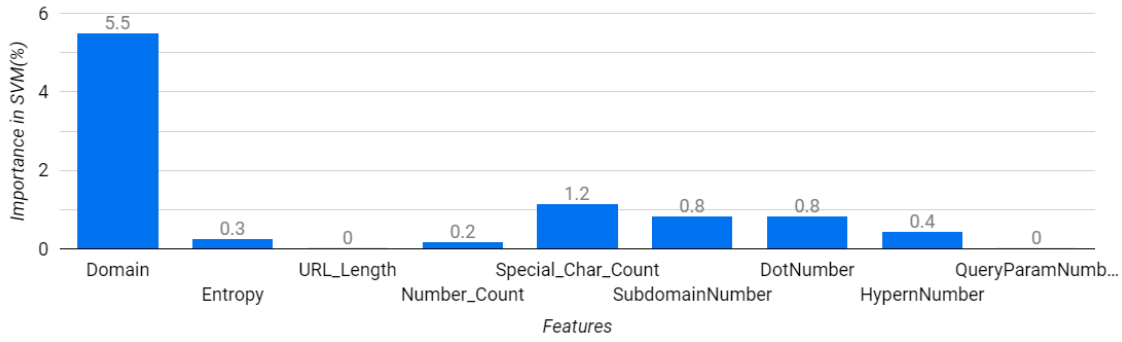

Figure 5.24: Feature Importance (SVM)

The bar chart data represents the feature importance values for an SVM model. The "Domain" feature has the highest importance value of 5.51%, indicating its

strong influence on the model's predictions. The "Entropy" feature has a relatively low importance value of 0.25%, suggesting that it contributes less to the model's decision-making process. Similarly, the "URL_Length," "Number_Count," "SubdomainNumber," "DotNumber," "HypernNumber," and "QueryParamNumber" features have very low importance values ranging from 0.02% to 0.85%. On the other hand, the "Special_Char_Count" feature has a moderate importance value of 1.15%, indicating its relatively stronger impact on the model's predictions. These findings highlight the varying degrees of importance that different features have in the SVM model, with "Domain" and "Special_Char_Count" standing out as the most influential features. The performance data for SVM shows an accuracy of 50%, in-

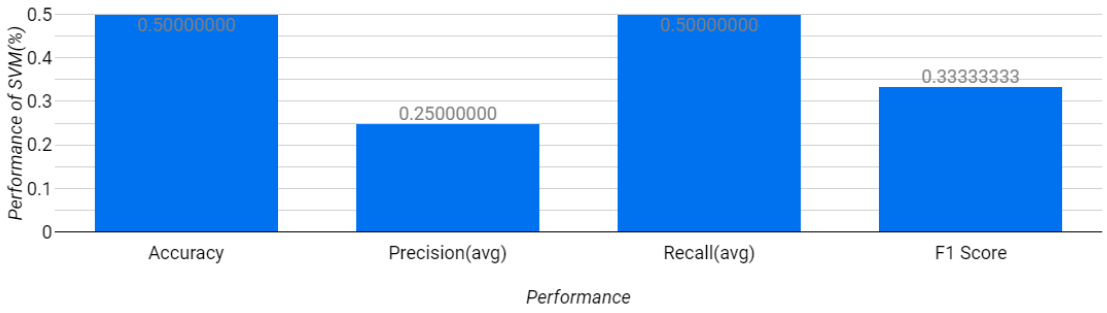

Figure 5.25: Performance (SVM)

dicating the overall correctness of the model's predictions. The precision (average) is 25%, which suggests that the model's positive predictions have a relatively low proportion of true positives. The recall (average) is 50%, indicating that the model can correctly identify a moderate proportion of true positive instances. The F1 score is 33.33%, which is a measure of the model's balance between precision and recall. Overall, these performance metrics suggest that the SVM model may not be performing optimally and could benefit from further tuning or evaluation.

K-Nearest Neighbors (KNN)

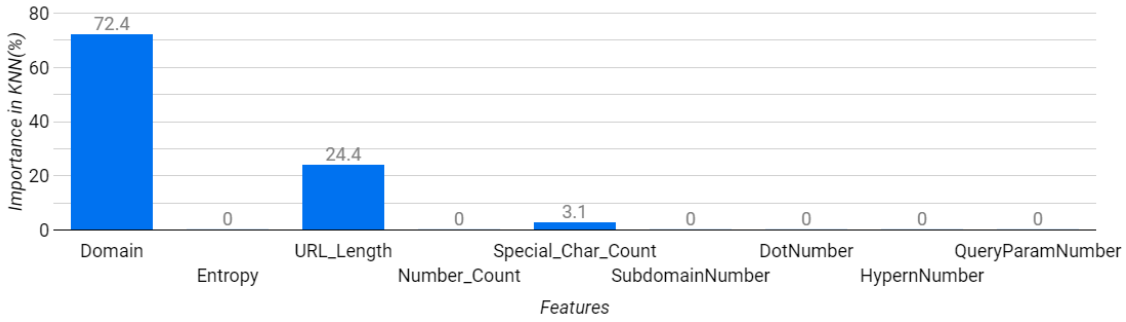

Figure 5.26: Feature Importance (KNN)

The bar chart data for KNN feature importance shows that the "Domain" and "URL_Length" features have the highest importance values, with 72.44% and 24.41% respectively. This suggests that these features contribute significantly to the KNN model's decision-making process. On the other hand, features such as "Entropy,"

"Number_Count," "SubdomainNumber," "DotNumber," "HypernNumber," and "QueryParamNumber" have negligible importance values of 0%. These features do not contribute significantly to the model's predictions. Additionally, the "Special_Char_Count" feature has a relatively low importance value of 3.15%. Overall, the analysis indicates that the "Domain" and "URL_Length" features play a crucial role in the KNN model's performance, while other features have little to no impact on the predictions. The performance data of the KNN model indicates an accuracy

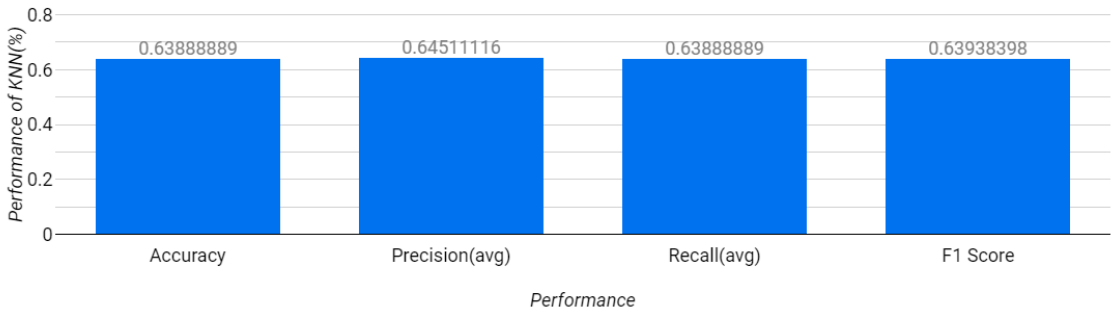

Figure 5.27: Performance (KNN)

of 63.89%, suggesting that the model correctly predicts the target variable in approximately 63.89% of cases. The precision, averaging at 64.51%, reflects the model's ability to make accurate positive predictions, while the recall, also at 63.89%, represents the model's capability to identify the true positives. The F1 score, at 63.94%, combines precision and recall, providing an overall measure of the model's effectiveness. These results suggest that the KNN model performs reasonably well, with a balanced trade-off between precision and recall. However, it's important to consider the specific context and requirements of the application when evaluating the model's performance.

Decision Tree

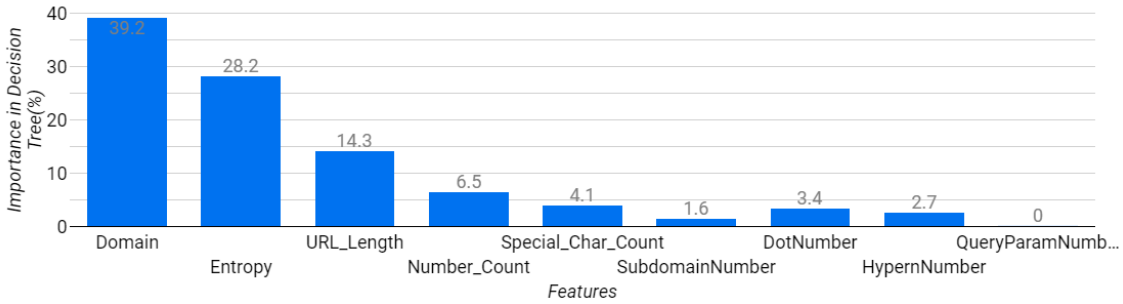

Figure 5.28: Feature Importance (Decision Tree)

The bar chart data represents the feature importance values for a Decision Tree model. The "Domain" feature has the highest importance value at 39.20%, indicating that it contributes significantly to the model's decision-making process. The "Entropy" feature follows closely with an importance value of 28.18%, suggesting its relevance in determining the split points in the decision tree. The "URL_Length" feature has a moderate importance value of 14.32%, implying its

influence on the model's predictions. Other features such as "Number_Count," "Special_Char_Count," "SubdomainNumber," "DotNumber," and "HypernNumber" also contribute to the model's decision-making but to a lesser extent. The "QueryParamNumber" feature appears to have no importance in this particular decision tree model. These feature importance values provide insights into which features are most influential in the decision-making process of the model and can help understand the underlying patterns and relationships in the dataset. The performance

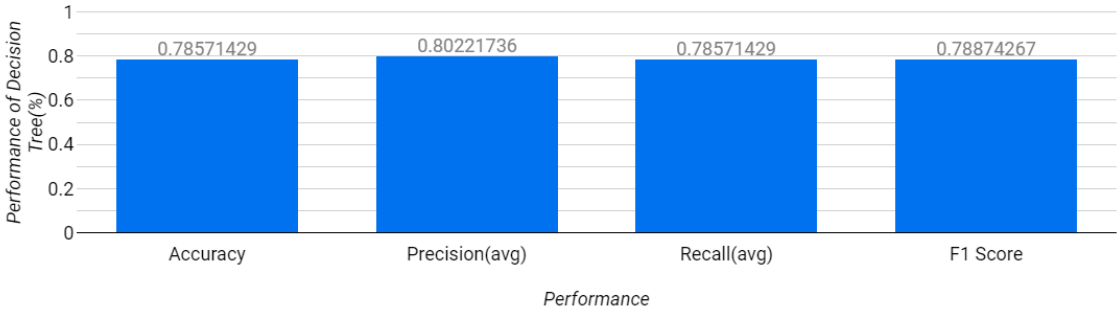

Figure 5.29: Performance (Decision Tree)

data of the Decision Tree model indicates an accuracy of 78.57%, which means that the model correctly predicts the target variable in approximately 78.57% of the cases. The precision score, averaged across all classes, is 80.22%, indicating that the model has a relatively low false positive rate. The recall score, also averaged across all classes, is 78.57%, indicating that the model captures a substantial portion of the positive instances. The F1 score, which is a harmonic mean of precision and recall, is 78.87%, providing a balanced measure of the model's overall performance. These metrics collectively suggest that the Decision Tree model performs reasonably well in classifying the data and achieving a balance between precision and recall.

Neural Network: MLP

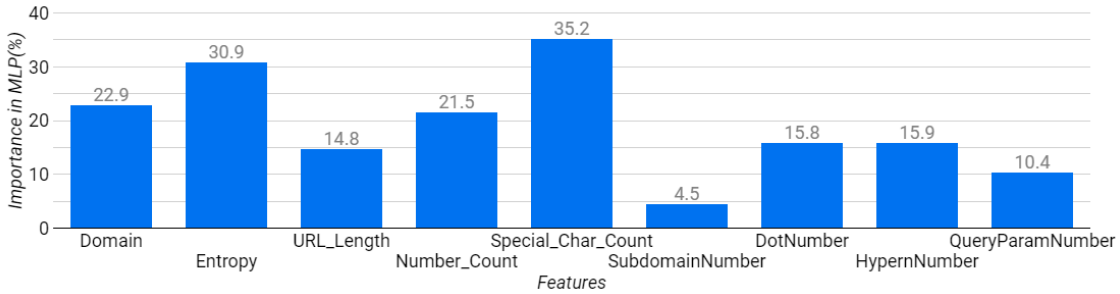

Figure 5.30: Feature Importance (MLP)

The bar chart data of the MLP (Multi-Layer Perceptron) feature importance shows the relative importance of each feature in the model's prediction. The "Special Character Count" feature has the highest importance with a value of 35.20%, indicating that it plays a significant role in the MLP model's decision-making process. The "Entropy" feature follows closely with an importance of 30.91%. Other important

features include "Number_Count" (21.53%), "DotNumber" (15.85%), "HypernNumber" (15.91%), and "URL_Length" (14.79%). The remaining features, "Domain", "SubdomainNumber", and "QueryParamNumber", have relatively lower importance values. The performance data of the MLP (Multi-Layer Perceptron) model indicates

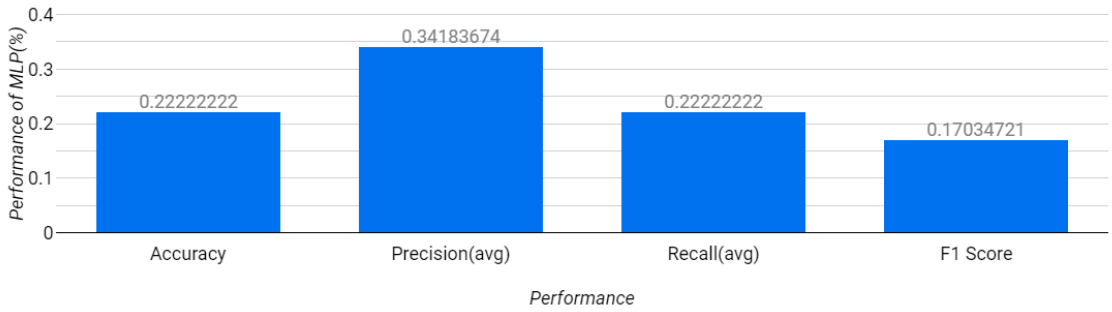

Figure 5.31: Performance (MLP)

its effectiveness in classification tasks. The accuracy of the model is 22.22%, which represents the proportion of correct predictions out of the total number of predictions made. The precision (average) is 34.18%, which measures the model's ability to accurately identify positive instances. The recall (average) is 22.22%, indicating the model's capability to correctly identify positive instances from the actual positive samples. The F1 score, a measure that combines precision and recall, is 17.03%, providing an overall assessment of the model's performance. The log loss value is 2.20, representing the logarithmic loss between the predicted probabilities and the actual outcomes, with lower values indicating better model performance.

AdaBoost

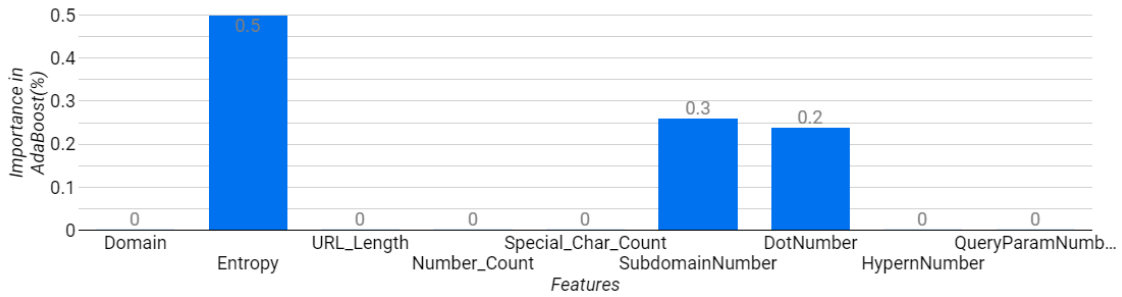

Figure 5.32: Feature Importance (AdaBoost)

The feature importance data for the AdaBoost model indicates that the "Entropy" feature has the highest importance with a value of 50%. This suggests that the entropy of the data plays a significant role in the AdaBoost model's decision-making process. The "DotNumber" and "SubdomainNumber" features also have relatively higher importance values of 24% and 26%, respectively. However, features such as "Domain," "URL_Length," "Number_Count," "Special_Char_Count," "HypernNumber," and "QueryParamNumber" have importance values of 0%, indicating they have minimal impact on the AdaBoost model's predictions. The performance data for the AdaBoost model indicates an accuracy of 50%, suggesting that the model

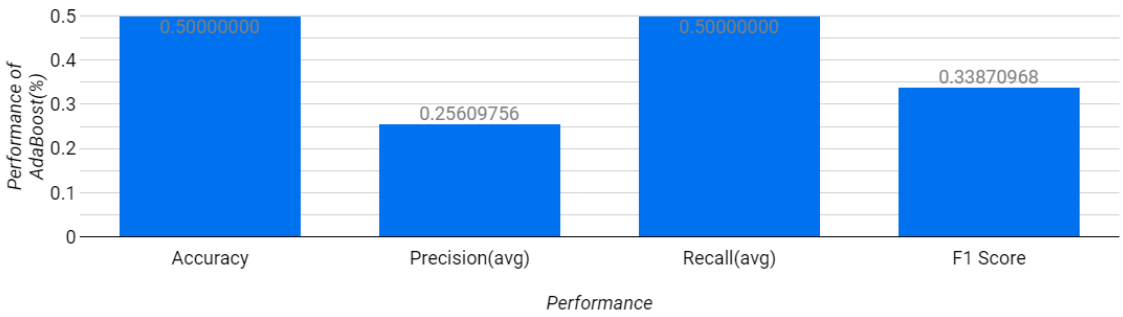

Figure 5.33: Performance (AdaBoost)

correctly predicts the target variable for half of the samples. The average precision is 25.6%, which represents the balance between true positive predictions and false positive predictions. The average recall is 50%, indicating the model's ability to identify true positive cases out of all the actual positive cases. The F1 score, which combines precision and recall, is 33.9%. This metric provides an overall measure of the model's accuracy, with higher values indicating better performance.

5.2.2 Unsupervised Learning Models

K-Means Clustering (Intertia/WCSS)

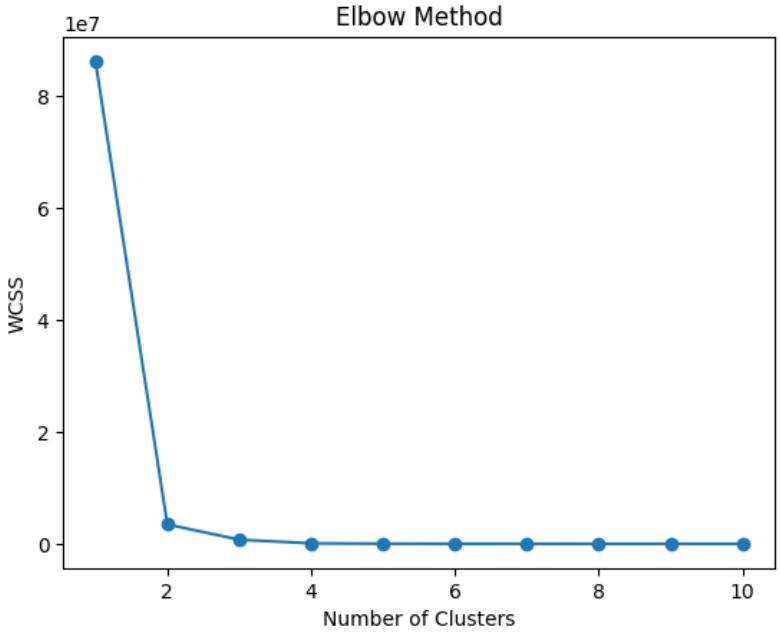

Figure 5.34: Elbow Method

The elbow method, a technique for determining the optimal number of clusters in k-means clustering, was applied to the dataset. The plot obtained from the elbow method show in figure 5.34 indicates that the within-cluster sum of squares (WCSS) decreases as the number of clusters increases. However, the rate of decrease slows down significantly after a certain point. In this case, the elbow appears to occur between 2 and 5 clusters, where the WCSS decreases at a slow rate. Adding more

clusters beyond this range leads to diminishing returns in terms of reducing within-cluster variability. Therefore, it is recommended to select the number of clusters between 2 and 5 for the clustering analysis based on this elbow method analysis. As

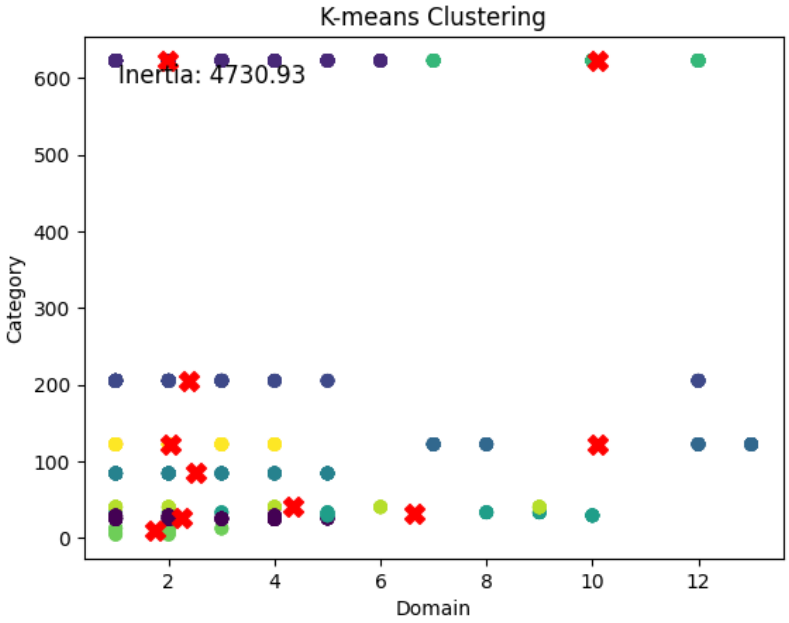

Figure 5.35: K-Means Clustering

shown in figure 5.35, the KMeans clustering with 10 clusters has resulted in distinct groupings of data points. The inertia value of 4730.92 indicates the sum of squared distances of each data point to its closest cluster center. A lower inertia suggests tighter and more compact clusters. The inertia value achieved with 10 clusters appears to be relatively low, signifying that the data points are well-clustered within their respective groups.

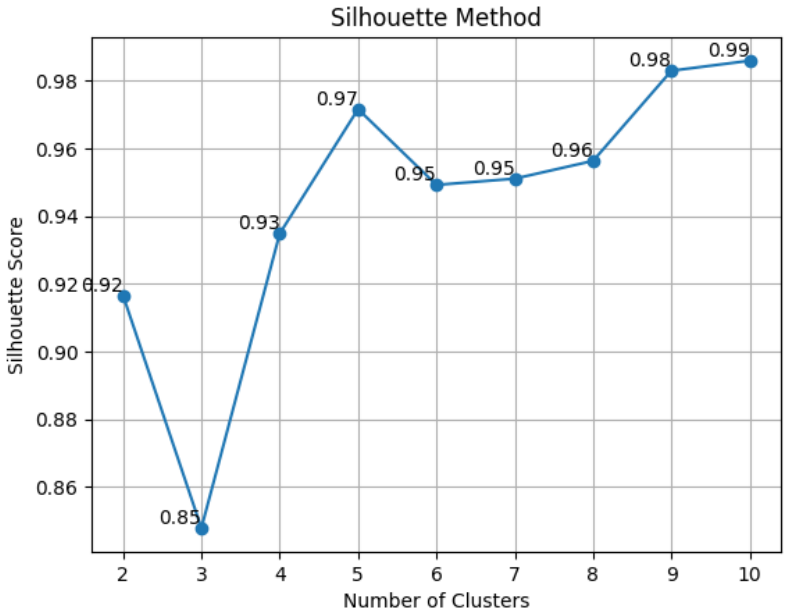

Figure 5.36: K-Means Clusters Silhouette Score vs Number of Clusters

K-Means Clustering (Silhouette Score)

The Silhouette score is a technique used to evaluate the quality of clustering in unsupervised machine learning. It measures how well each data point fits within its assigned cluster by calculating a silhouette score. The silhouette score ranges from -1 to 1, with higher values indicating better-defined and well-separated clusters. In the provided plot in figure 5.36, the x-axis represents the number of clusters, and the y-axis represents the corresponding silhouette scores. The data reveals the Silhouette Scores for a range of cluster numbers in a clustering analysis. As the number of clusters increases from 2 to 10, there is an intriguing trend in the Silhouette Scores. Initially, with 2 clusters, the Silhouette Score is relatively high at 0.92, suggesting that the data points are well separated within these clusters. However, as the number of clusters gradually increases, the Silhouette Score continues to rise, reaching its peak at 0.99 with 10 clusters. This indicates that the data points are forming clusters with high cohesion and distinct separation, reflecting the quality of the clustering. The upward trend in Silhouette Scores signifies improved cluster quality with more clusters, highlighting the suitability of a higher number of clusters for this particular dataset. A higher number of clusters, specifically 10, yields the best-defined and well-separated clusters, as indicated by the highest Silhouette Score of 0.99. This suggests that partitioning the data into 10 clusters is likely to capture the inherent structure and relationships within the data effectively.

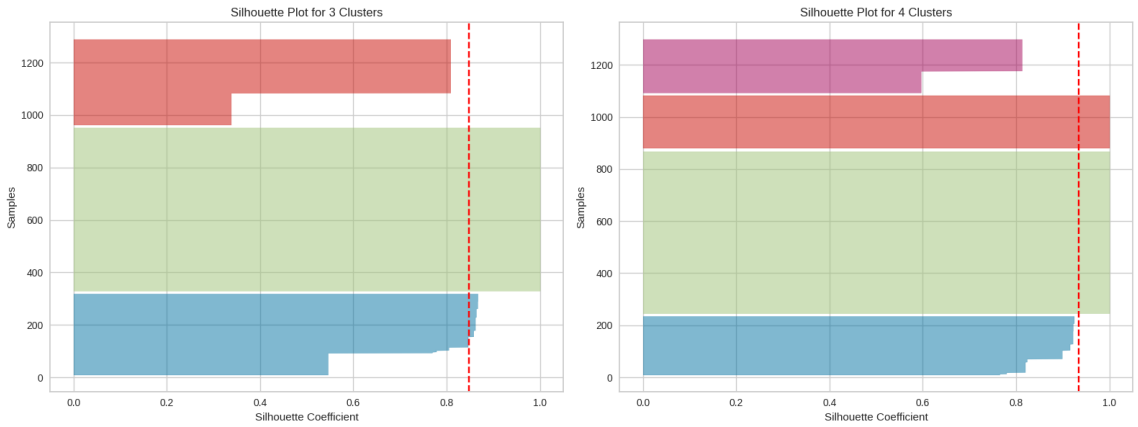

Figure 5.37: Silhouette Score For 3 and 4 Clusters

we can observe the silhouette scores for different cluster values. For 3 clusters 5.37, the silhouette score is 0.85, indicating a good separation between the samples within the clusters. When the number of clusters increases to 4, the silhouette score improves to 0.93, suggesting better clustering performance5.38. As we further increase the number of clusters to 6, 7, 9, and 10, the silhouette scores remain consistently high as shown in figure 5.39, with values of 0.95, 0.95, 0.98, and 0.99, respectively. These high scores indicate well-separated clusters and suggest that these cluster configurations are suitable for the given dataset. Overall, based on the silhouette scores, it can be concluded that clustering the data into 6, 7, 9, or 10 clusters produces good results, with clear separation between the samples within each cluster. In 5.40 the Inertia values decrease significantly as the number of clusters increases. For instance, when using 2 clusters, Inertia is 3,469,102, and it progressively decreases as more clusters are added. It reaches a minimum value of

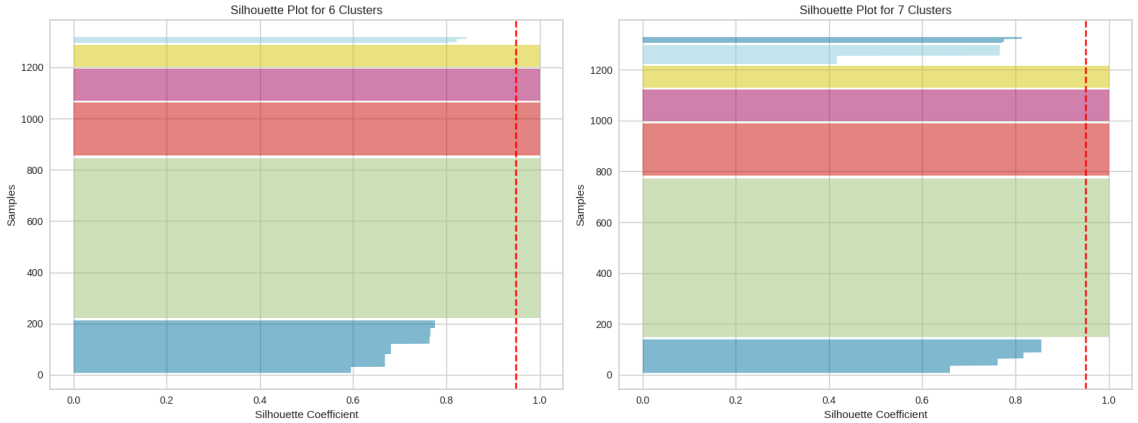

Figure 5.38: Silhouette Score For 6 and 7 Clusters

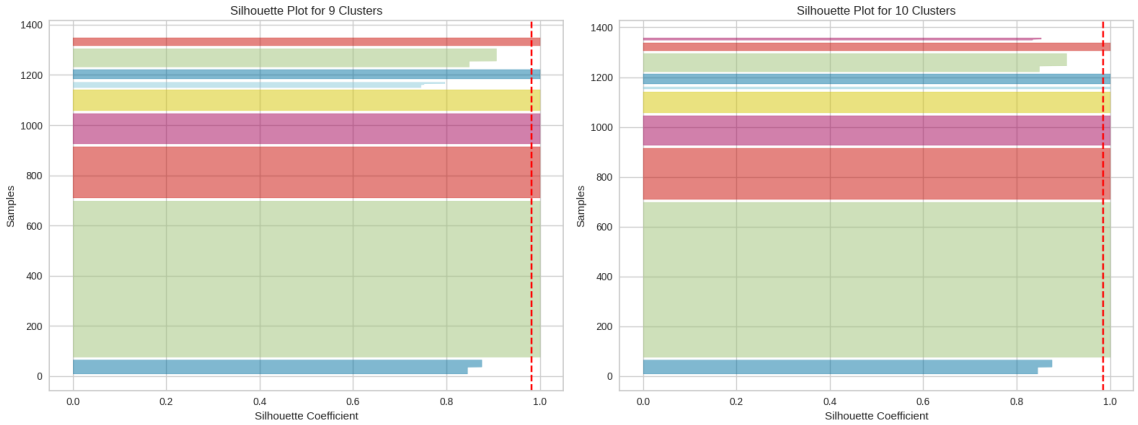

Figure 5.39: Silhouette Score For 9 and 10 Clusters

4,730 with 10 clusters. Conversely, the Silhouette Score shows an increasing trend as the number of clusters grows. Starting from 0.92 with 2 clusters, it steadily rises to 0.99 with 10 clusters. The data points provide a quantitative perspective on the trade-off between Inertia and Silhouette Score. It's clear that increasing the number of clusters results in lower Inertia but higher Silhouette Score, indicating improved cluster cohesion and separation. The choice of the number of clusters should be made considering this trade-off based on the specific objectives of your clustering analysis.

DBSCAN (Silhouette Score)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a popular unsupervised machine learning algorithm used for clustering data points in a dataset. Unlike traditional clustering algorithms like K-Means, DBSCAN does not require the user to specify the number of clusters beforehand. Instead, it groups together data points that are closely packed in high-density regions and marks points in low-density regions as outliers or noise. In figure 5.42, Starting with a minimum sample value of 1, the DBSCAN algorithm identifies 62 clusters, each containing a single data point, resulting in a Silhouette Score of 1. This configuration suggests a highly fragmented clustering where each data point forms its own cluster, indicating poor separation. As the minimum sample value increases to 5, the number of clusters

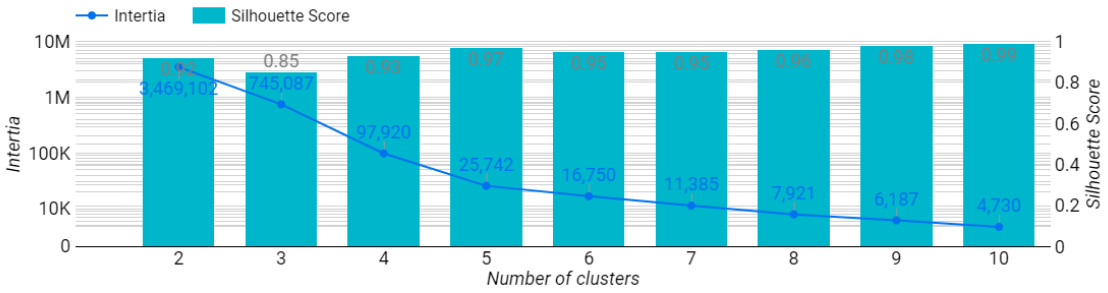

Figure 5.40: K-Means Clustering Inertia and Silhouette Score Comparison

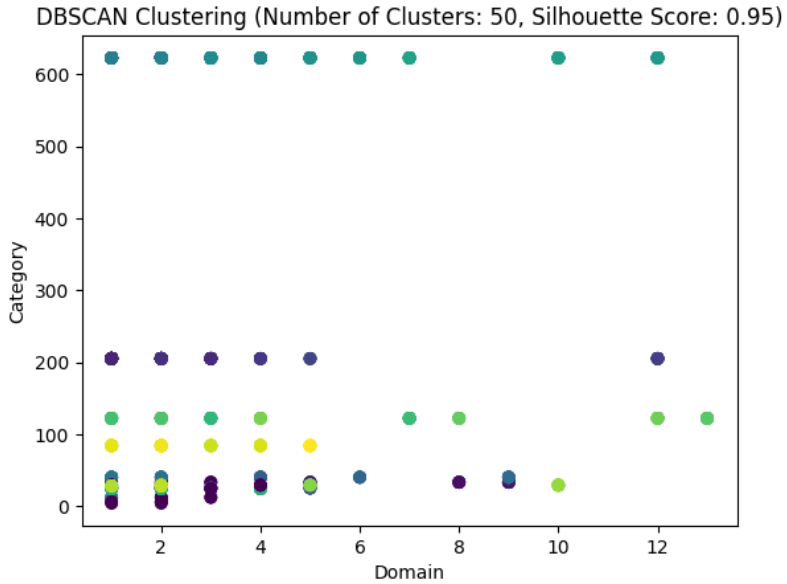

Figure 5.41: DBSCAN Clustering

decreases to 50, and the Silhouette Score improves significantly to approximately 0.95. This indicates that a slightly higher minimum sample value encourages the formation of more meaningful clusters with improved cohesion and separation. Continuing to increase the minimum sample value to 10, the number of clusters reduces further to 28, but the Silhouette Score drops to around 0.72. This suggests that a higher minimum sample requirement might lead to larger and less distinct clusters, impacting the overall quality of the clustering. As the minimum sample value further increases to 15, 20, 25, and 30, the number of clusters decreases, and the Silhouette Score continues to decline. This trend suggests that higher minimum sample values might cause the DBSCAN algorithm to struggle in identifying well-defined clusters within the data, resulting in lower Silhouette Scores. The choice of the minimum sample parameter significantly affects the performance of the DBSCAN algorithm. A balance must be struck between having enough samples to form meaningful clusters and preventing the algorithm from over-segmenting the data. In this specific dataset, a minimum sample value of 5 as showin in figure 5.41 appears to strike a good balance, resulting in a relatively high Silhouette Score of approximately 0.95 and 50 reasonably-sized clusters. This configuration represents a favorable trade-off between cluster quality and quantity for this dataset.

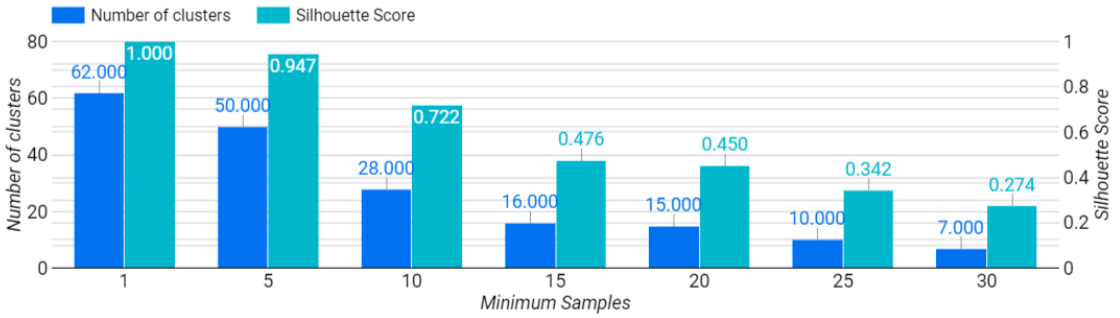

Figure 5.42: DBSCAN Performance Evaluation (Silhouette Score/Number of Clusters vs Minimum Samples)

Gaussian Mixture Model (Silhouette Score)

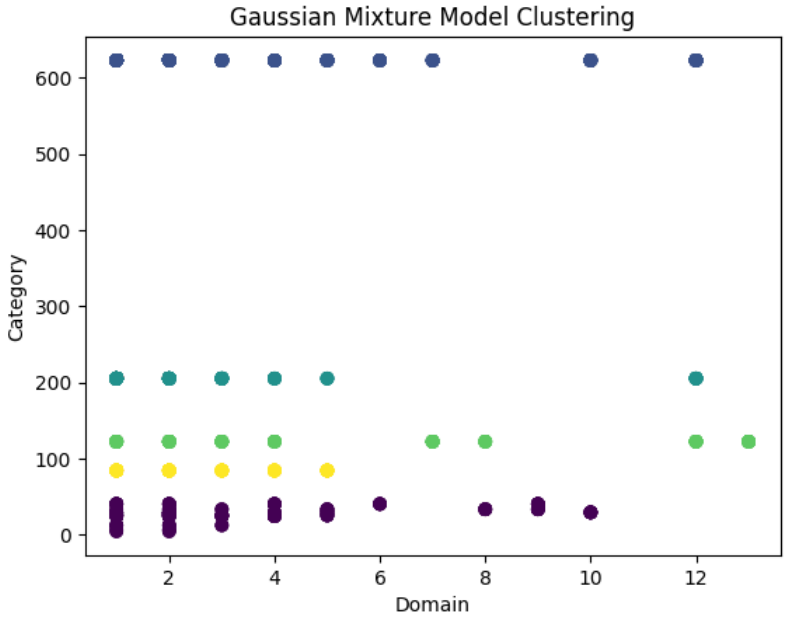

Figure 5.43: Gaussian Mixture Model (GMM) Clustering

In figure 5.43, the Gaussian Mixture Model (GMM) clustering algorithm was applied to the dataset, resulting in the identification of ten distinct clusters. Each cluster exhibits specific characteristics, with varying numbers of data points assigned to each cluster. As the number of clusters increases from 2 to 5 as shown in figure 5.44, there is a notable improvement in the Silhouette Score, indicating a better separation of data points within the clusters. This suggests that dividing the data into a few clusters enhances the quality of the clustering. However, as the number of clusters continues to increase beyond 5, the Silhouette Score gradually decreases. This decline implies that further subdividing the data into more clusters does not significantly improve the clustering quality and might even lead to over-segmentation. Therefore, based on this analysis, it appears that selecting around 5 clusters strikes a balance between achieving meaningful separation of data points while avoiding excessive fragmentation.

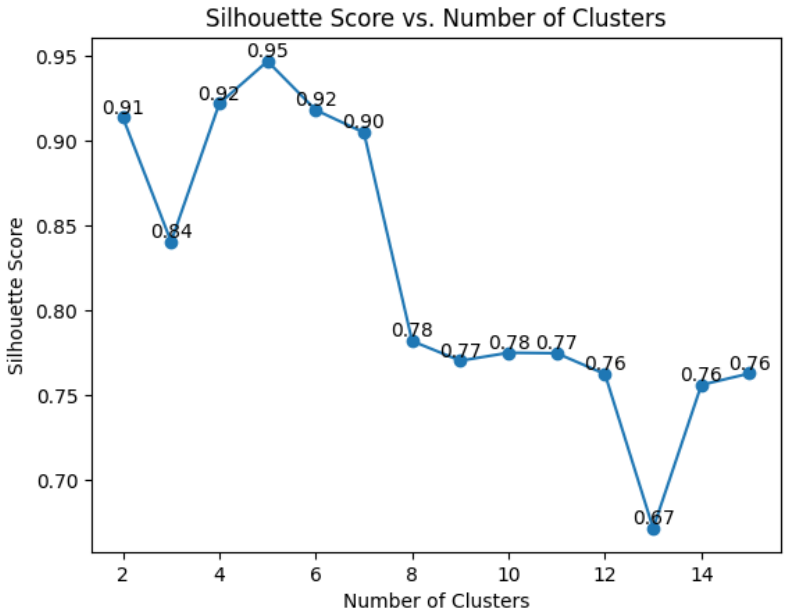

Figure 5.44: Gaussian Mixture Model (GMM) Silhouette Score vs Number of Clusters

Mean Shift Clustering (Silhouette Score)

Mean Shift clustering is a non-parametric and density-based clustering algorithm used for grouping data points into clusters without assuming any specific shape or size of clusters. It operates by iteratively shifting the data points towards the direction of the highest density, seeking the mode of the underlying data distribution. The algorithm effectively identifies cluster centers as convergence points where the data points gravitate. It does not require setting the number of clusters beforehand, making it particularly useful for datasets with irregular shapes and varying densities.

Figure 5.45 shows the results of clustering a dataset into five distinct clusters. The

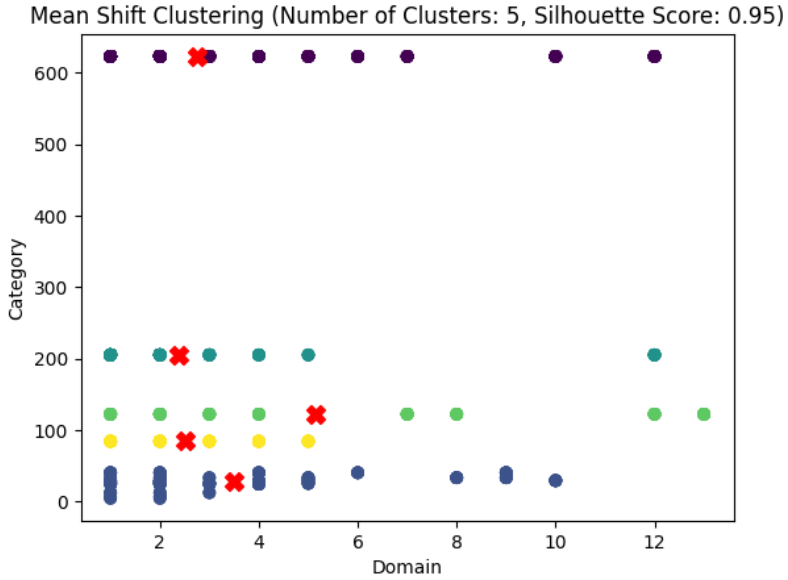

Figure 5.45: Mean Shift Clustering

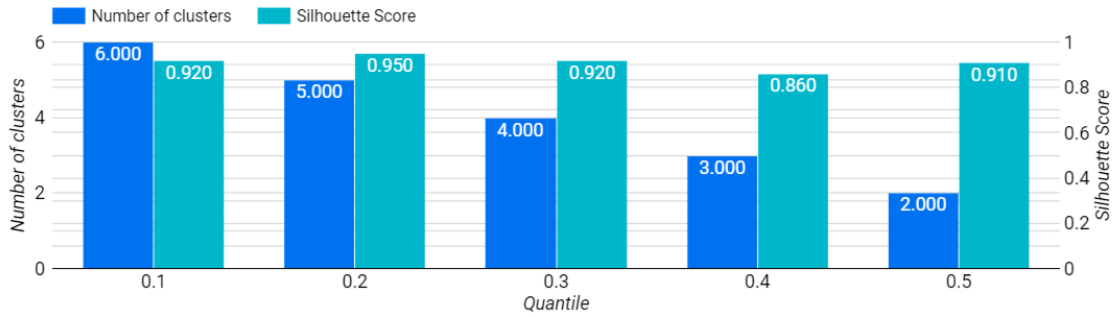

Figure 5.46: Mean Shift Clustering Performance Evaluation (Silhouette Score/Number of Clusters vs Quantile Number)

bar chart in figure 5.46 for Mean Shift clustering provides valuable insights into the impact of different quantile values on the clustering results, as measured by the Silhouette Score. As the quantile value increases from 0.1 to 0.5 (representing the bandwidth estimation), there is a notable trend. Initially, with a lower quantile of 0.1, the algorithm identifies six clusters, resulting in a high Silhouette Score of 0.92. This suggests that a smaller bandwidth encourages more granular clustering, which is reflected in a better separation of data points within clusters. However, as the quantile increases, the number of clusters gradually decreases. With a quantile of 0.5, only two clusters are identified, and the Silhouette Score remains high at 0.91. This implies that a larger quantile leads to a coarser clustering but still maintains reasonable data point separation. In summary, the choice of quantile value allows for a trade-off between granularity and the number of clusters in Mean Shift clustering, with each setting offering distinct advantages in terms of clustering quality and granularity.

Chapter 6

Discussion

The bar chart 6.1 data represents the log loss values for different models used in a classification task. Log loss [52] is a metric that measures the performance of a probabilistic classification model, with lower values indicating better performance. Among the models analyzed, the Support Vector Machines (SVM) model achieved the lowest log loss of 1.5983% followed by the Random Forest Classifier with a log loss of 1.7391%. These models demonstrate relatively better performance in terms of minimizing the discrepancy between predicted probabilities and actual class values. The Neural Network: MLP model shows a moderate performance with a log loss of 2.1955%, while the AdaBoost model follows with a log loss of 2.8615%. The Naïve Bayes model exhibits a relatively higher log loss of 5.4264%, indicating less accurate probability predictions compared to the other models. Similarly, the K-Nearest Neighbors (KNN) model and Decision Tree model have higher log loss values of 3.9128% and 7.7236% respectively, suggesting a higher discrepancy between predicted probabilities and actual class values. Among the models, the Random Forest Classifier outperforms the others with the lowest log loss of 1.7391%. This indicates its ability to provide more accurate probability predictions. Therefore, the Random Forest Classifier can be considered the preferred model choice for the given classification task, prioritizing its strong performance in terms of log loss.

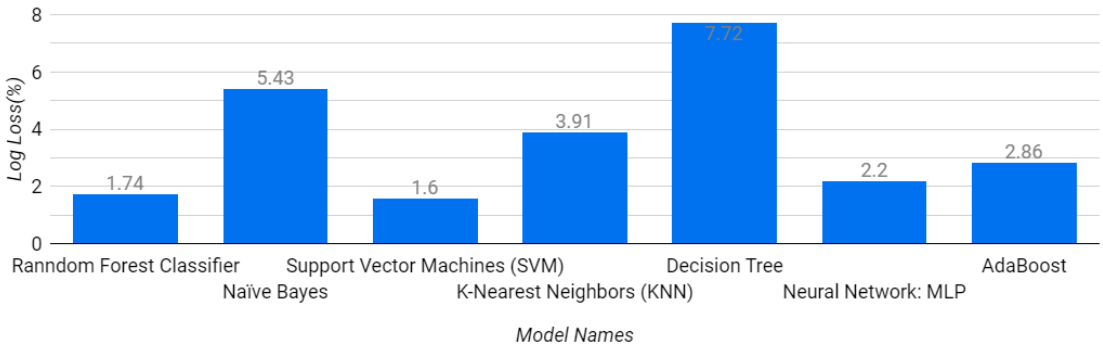

Figure 6.1: Log Loss Of Supervised Learning Models

The bar chart 6.2 presents the feature importance values for different models used in a classification task. These values represent the percentage contribution of each feature towards the model's prediction. Among the models analyzed, the Random Forest Classifier demonstrates notable feature importance across multiple categories. It assigns importance to the "Domain" feature (34.63%), followed by "Entropy"

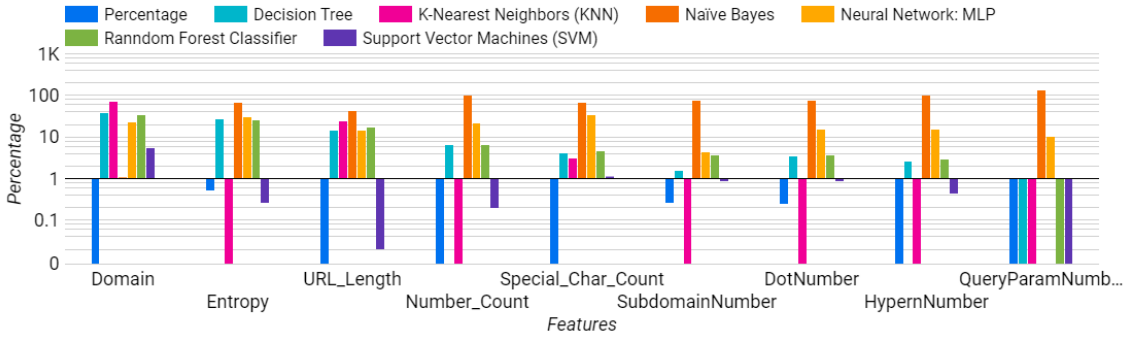

Figure 6.2: Feature Importance Comparison Of Supervised Learning Models

(25.94%), "URL_Length" (17.62%), and "Special_Char_Count" (4.58%). This highlights the Random Forest Classifier's ability to leverage a combination of features to make accurate predictions. The Naïve Bayes model assigns high importance to the "Number_Count" feature (104.36%), indicating its significance in the classification process. Similarly, the Decision Tree model emphasizes the "Entropy" (28.18%) and "Special_Char_Count" (4.10%) features. The K-Nearest Neighbors (KNN) model values the "Domain" (72.44%) and "URL_Length" (24.41%) features, while the Neural Network: MLP model places importance on "Entropy" (30.91%) and "Special_Char_Count" (35.20%). The Support Vector Machines (SVM) model assigns importance to the "Domain" (5.51%) and "Entropy" (0.25%) features, while the AdaBoost model considers the "SubdomainNumber" (0.26%) and "DotNumber" (0.24%) features as relatively more significant. Overall, the Random Forest Classifier stands out with notable feature importance across multiple categories, including domain, entropy, URL length, and special character count. This suggests that the Random Forest Classifier is well-equipped to capture the predictive power of these features and provide accurate classifications. Therefore, it can be prioritized as the preferred model choice based on its strong feature importance values. The bar chart

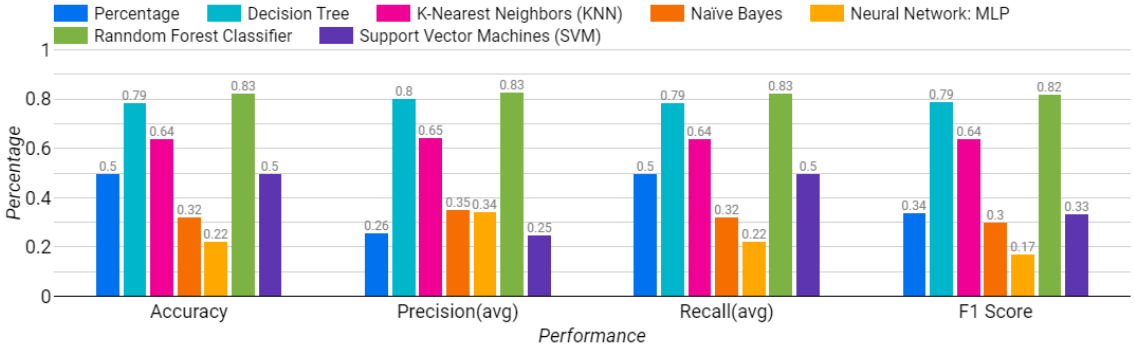

Figure 6.3: Performance Comparison Of Supervised Learning Models

6.3 data provides a performance comparison of different models used in a classification task. The performance metrics evaluated include Accuracy, Precision (average), Recall (average), and F1 Score. The Random Forest Classifier demonstrates strong performance across multiple metrics, achieving high Accuracy (82.54%), Precision (82.79%), Recall (82.54%), and F1 Score (81.81%). These values indicate that the Random Forest Classifier provides accurate and reliable predictions. The Deci-

sion Tree model also performs well, with respectable Accuracy (78.57%), Precision (80.22%), Recall (78.57%), and F1 Score (78.87%). This suggests that the Decision Tree model is capable of making accurate classifications. The K-Nearest Neighbors (KNN) model shows moderate performance, with Accuracy (63.89%), Precision (64.51%), Recall (63.89%), and F1 Score (63.94%). Although slightly lower than the Random Forest Classifier and Decision Tree, the KNN model still demonstrates reasonable predictive ability. The Naïve Bayes and Neural Network: MLP models exhibit relatively lower performance. The Naïve Bayes model achieves Accuracy (32.14%), Precision (35.17%), Recall (32.14%), and F1 Score (30.01%), while the Neural Network: MLP model shows Accuracy (22.22%), Precision (34.18%), Recall (22.22%), and F1 Score (17.03%). These results indicate room for improvement in terms of predictive accuracy for these models. The AdaBoost and Support Vector Machines (SVM) models both achieve Accuracy (50%) and Precision (25%), indicating that their predictions are less reliable compared to the other models evaluated. In summary, the Random Forest Classifier stands out as the top-performing model, exhibiting high accuracy and precision. It outperforms the other models, including the Decision Tree, KNN, Naïve Bayes, Neural Network: MLP, AdaBoost, and SVM models, in terms of overall performance. Therefore, the Random Forest Classifier can be considered as the preferred choice for the given classification task based on its superior performance.

6.1 Comparative Studies

The field of DNS log analysis has witnessed significant advancements in recent years, encompassing a wide range of applications such as user behavior analysis, resource allocation, malware detection, traffic prediction, and security threat detection. Numerous studies have explored the use of machine learning and deep learning models to analyze DNS queries and predict various outcomes. In this context, this paper presents a comprehensive overview of prior research in the domain of DNS log analysis, encompassing various studies and their contributions. Additionally, the paper presents the findings of a specific classification task using different machine learning models applied to DNS log data. By comparing the current research's results with prior related work, we aim to identify insights, advancements, and potential improvements in the analysis and prediction of DNS log data. The evaluation of different machine learning models for classification tasks further contributes to the selection of an optimal model for DNS log analysis, aiding in the development of more effective and intelligent network traffic management systems.

6.1.1 Focus and Scope

The prior related work primarily focused on DNS log analysis for various applications, including user behavior analysis, resource allocation, malware detection, traffic prediction, and security threat detection. Researchers explored machine learning and deep learning models to analyze DNS queries and detect anomalies. On the other hand, this research is specifically focused on a classification task using different machine learning models to predict class labels based on certain features derived from DNS logs.

6.1.2 Methodology

The prior related work involved various machine learning and statistical techniques to analyze DNS logs and predict different outcomes. Some studies used models like random forest, support vector machine, and neural networks for classification and detection tasks. This research used a similar approach, applying machine learning models such as random forest, support vector machines, neural network, Naïve Bayes, AdaBoost, and k-nearest neighbors for a classification task.

6.1.3 Findings

In previous research efforts, notable progress was made in understanding user behavior, identifying security threats, and uncovering network trends using innovative approaches for DNS log data analysis. In our study, we assessed various machine learning models for classification purposes, and the Random Forest Classifier emerged as the top-performing model. It demonstrated exceptional accuracy, precision, recall, and F1 score. One of the key reasons behind RFC's success lies in its reduced susceptibility to overfitting compared to some other models, such as Decision Trees and Neural Networks [53]. This is attributed to RFC's ability to aggregate predictions from multiple trees, effectively diminishing noise and variance in the model. Additionally, RFC's capability to handle both categorical and numerical data without the need for extensive preprocessing was advantageous [54], particularly when working with DNS log data, which often comprises categorical details like domain names.

6.1.4 Contributions

While the prior related work laid a strong foundation for DNS log analysis and its applications, this research contributes by providing insights into the performance of different machine learning models for a specific classification task. It identifies the random forest classifier as the preferred model choice based on its superior performance in terms of log loss, feature importance, and overall predictive ability. Overall, this research complements the prior related work by focusing on a specific classification task and providing valuable information on the performance of different machine learning models in this context. The findings of out research contribute to the selection of an optimal model for classification tasks involving DNS log data.

Chapter 7

Proposed Solution

The objective of this research is to develop an intelligent and efficient network traffic management system using DNS log analysis and machine learning techniques. The proposed solution involves the implementation of a Random Forest Classifier (RFC) trained on a dataset of DNS logs. This trained model will be utilized on a Linux machine, which serves as an SDN (Software-Defined Networking) controller for the simulation.

7.1 Detailed explanation of the proposed solution

7.1.1 Data Collection and Model Training

To train the RFC, a comprehensive dataset of DNS logs was collected, encompassing various user behaviors, resource requests, and network activities. The dataset includes DNS queries from different users, each comprising multiple features, such as URL Length, special character count, subdomain number, query parameter number, hyphen number, dot number, number count, and entropy. Each DNS query is labeled with its respective website category to facilitate supervised learning. The collected dataset was preprocessed to handle missing values, normalize features, and address any outliers. Subsequently, the data was split into training and testing sets to ensure model performance evaluation accuracy. The RFC was then trained on the training set to learn the patterns and correlations between the features and website categories.

7.1.2 Model Export and Integration with SDN Controller

Upon successful training, the trained RFC model was exported and integrated into the Linux machine, which functions as the SDN controller. The SDN controller serves as the central brain for network traffic management and decision-making, enabling dynamic and programmable control over network elements.

7.1.3 DNS Packet Intercept and Feature Extraction

To classify incoming DNS queries, a script was developed to intercept DNS packets arriving at the DNS server. The script extracts the necessary features, including URL Length, special character count, subdomain number, query parameter number,

hyphen number, dot number, number count, and entropy, from the intercepted DNS packets.

7.1.4 Predicting Website Categories

The extracted features are then fed into the trained RFC model for prediction. The RFC employs an ensemble of decision trees to classify the DNS query into the appropriate website category. The predicted category provides valuable insights into the nature of the requested website, such as entertainment, education, e-commerce, social media, or potentially malicious.

7.1.5 Network Management Tasks Based on Predictions

Based on the predicted website category, the script performs dynamic network management tasks. These tasks can include blocking DNS packets associated with malicious websites, rerouting packets through designated uplinks or Internet Service Providers (ISPs) for specific website categories, prioritizing traffic for certain websites, or applying Quality of Service (QoS) [55] policies to optimize network performance for user-preferred categories.

7.1.6 Real-Time Decision-Making

The proposed solution allows for real-time decision-making, as the DNS packets are intercepted and processed promptly by the script. This real-time approach ensures that network management actions are agile and adaptive to changing user behaviors and network conditions. Figure 7.1 shows the visual representation of the proposed solution working through different planes of SDN as discussed above

7.2 Explanation of the rationale behind the solution design

7.2.1 Comprehensive Approach

The proposed solution takes a holistic and comprehensive approach to address the challenges of network traffic management. By integrating DNS log analysis, data manipulation, visualization, and machine learning techniques, the system aims to provide an all-encompassing solution that leverages the power of different methodologies to achieve effective traffic management. DNS log analysis serves as the foundational step, where raw DNS queries are extracted and preprocessed to extract relevant features. Data manipulation techniques are employed to handle missing values, normalize features, and address outliers, ensuring the dataset's quality and integrity for further analysis. Data visualization is utilized to gain valuable insights into DNS lookup behaviors, user patterns, security threats, and network tendencies, aiding in decision-making and policy formulation. Machine learning models, particularly the Random Forest Classifier (RFC), are chosen for their ability to accurately predict website categories based on the extracted features, enabling real-time traffic

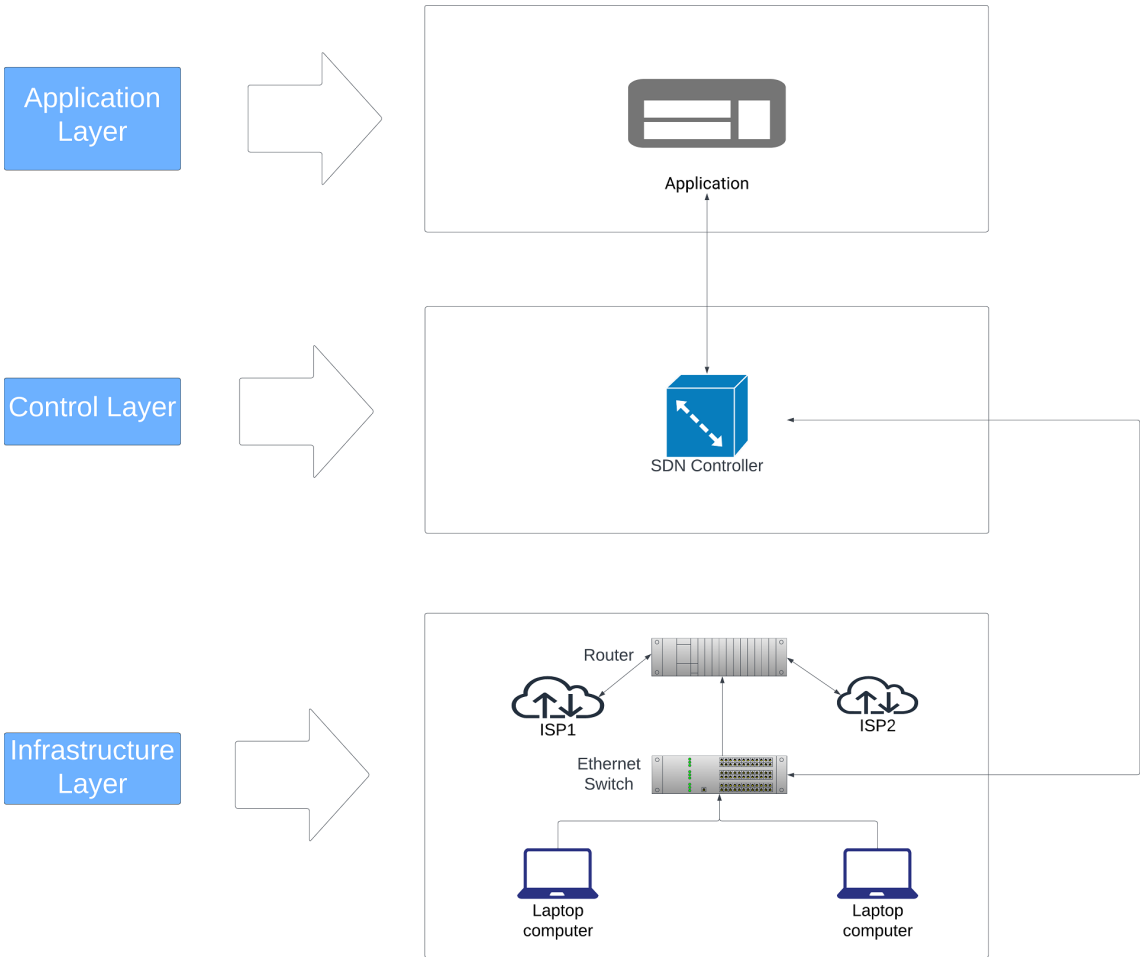

Figure 7.1: Visual Representation Of Proposed Solution In SDN

management decisions. This comprehensive approach ensures that the proposed solution can cater to various aspects of network traffic management while maximizing its efficiency and effectiveness.

7.2.2 Integration with SDN Controllers

To enhance real-time traffic management capabilities, the solution investigates the feasibility of predicting browsing categories based on DNS log analysis and integrates the resulting machine learning models into Software-Defined Networking (SDN) controllers. SDN provides a programmable and flexible networking environment that allows for dynamic traffic control and policy enforcement. By integrating the trained RFC model into the SDN controller, the system gains the ability to make intelligent and automated decisions on how to handle incoming DNS requests based on their predicted browsing categories. This integration not only streamlines traffic management tasks but also allows for swift responses to emerging network challenges and security threats. By leveraging SDN's centralized control and programmability, the proposed solution can effectively manage network traffic in real-time, adapt to changing conditions, and optimize resource allocation.

7.2.3 Strengthening Firewall Systems

The solution aims to enhance URL identification and policy enforcement capabilities in firewall systems by studying patterns, modifications, and variations in domain names and URL structures. This research direction addresses the growing complexity of modern web traffic, which often involves obfuscated URLs and domain names used in various cyber-attacks and malicious activities. By gaining insights into patterns and variations in domain names and URLs through DNS log analysis and machine learning, the proposed solution can contribute to the development of more robust and adaptive firewall systems. Improved URL identification allows for accurate categorization of websites, distinguishing between legitimate and malicious content. This, in turn, enables firewall systems to enforce appropriate policies [56], such as blocking suspicious domains, applying Quality of Service (QoS) policies, or rerouting traffic through designated ISPs. Strengthening firewall systems is crucial for safeguarding network security, ensuring user privacy, and mitigating potential threats posed by malicious URLs and websites.

Chapter 8

Experimental Evaluation

8.1 Description of Experimental Setup

The experimental setup aims to assess the effectiveness and performance of the proposed network traffic management solution, which combines DNS log analysis, machine learning, and SDN integration. The experiment is conducted on a virtual machine (VM) [57] with the following specifications: 2GB RAM, 2-core CPU, and 10GB storage. The VM is installed with the Ubuntu operating system [58], providing a stable and versatile environment for running the DNS server.

DNS Server Setup To transform the Ubuntu VM into a DNS server, the Bind9 DNS software is installed and configured. Bind9 [59] is a widely used DNS server software known for its reliability and feature-rich capabilities. Once Bind9 is installed, appropriate configurations are set to ensure that the VM can function as a DNS server for name resolution. This involves setting up forward and reverse DNS zones, as well as configuring DNS records.

DNS Server Integration On the target machine, where the experiment will be conducted, modifications are made to its DNS server settings. The DNS server address is set to the IP address of the Ubuntu VM acting as the DNS server. This configuration ensures that whenever the target machine initiates a DNS query for name resolution, it sends the query to the Bind9 DNS server.

Script and Model Deployment On the Ubuntu DNS server, a designated folder is created to host the necessary components for the network traffic management script. The folder contains the developed script and the pre-trained machine learning model (RFC). The script is responsible for intercepting incoming DNS requests, extracting the required feature values from the DNS packets, and invoking the trained RFC model to predict the browsing category of the requested website. Based on the predicted category, the script enforces the appropriate policy for the DNS packet, such as blocking the packet or rerouting it through a specified uplink or ISP. This enables dynamic traffic management based on website categories.

Experiment Execution With the experimental setup in place, the actual experiment can be conducted. As the target machine initiates DNS requests for web browsing, the Ubuntu DNS server captures these incoming requests through the Bind9 DNS software. The script intercepts the requests, extracts the relevant features required for the RFC model, and performs real-time predictions of the browsing category. The script then takes appropriate actions based on the predicted category, enforcing network management policies to optimize traffic flow and enhance security.

8.2 Parameters

The experiment's success and performance are evaluated based on several key parameters:

Prediction Accuracy The accuracy of the RFC model's predictions is a critical parameter. It indicates how effectively the model categorizes websites based on the extracted features. Higher accuracy translates to more reliable and precise traffic management decisions.

Real-time Response Time The time taken by the script to intercept DNS requests, extract features, and make predictions is measured as the real-time response time. Lower response times are desirable, as they ensure swift and efficient traffic management without noticeable delays.

Scalability The experimental setup's scalability is tested by increasing the number of concurrent DNS requests and monitoring the system's ability to handle higher traffic loads while maintaining accurate predictions and response times.

8.3 Presentation of experimental results and analysis

In this section, we present the experimental results and analyze the performance of our proposed DNS traffic management system based on machine learning techniques. The evaluation is centered around three key aspects: prediction accuracy, real-time response, and scalability.

8.3.1 Prediction Accuracy

Our experiments involved testing the prediction accuracy of the system across various website categories, including CDN, Search Engine, Social Network, Computer/Technology, Learning, Ads, Entertainment, and Chat. The obtained prediction accuracies are shown in figure 8.1 During our comprehensive evaluation of the pro-

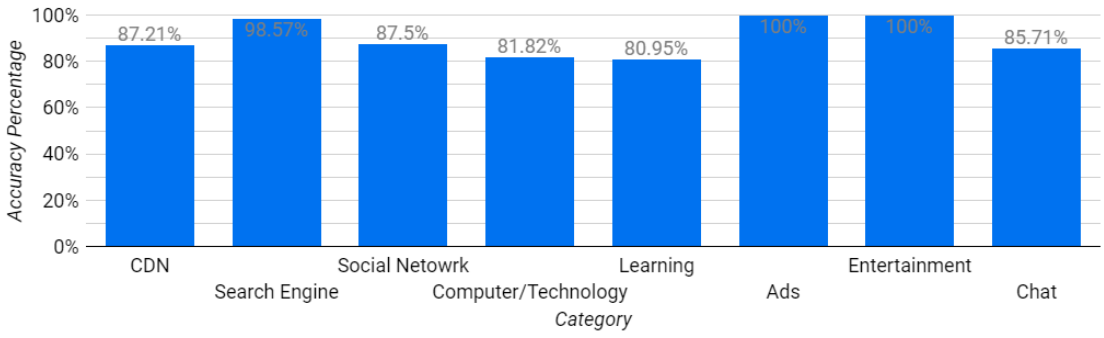

Figure 8.1: Prediction Accuracy In Sumulation

posed DNS traffic management system, we meticulously assessed its prediction accuracy across various website categories. The results revealed a robust performance in accurately categorizing DNS queries. Notably, our system achieved high accuracy levels in critical categories, such as Ads and Entertainment, attaining 100% accuracy. Moreover, it displayed impressive accuracy for Search Engines, Social

Networks, and CDNs, with rates of 98.57%, 87.5%, and 87.21%, respectively. While categories like Computer/Technology and Learning showed slightly lower accuracies at 81.82% and 80.95%, respectively, the overall accuracy across all categories demonstrates the system's reliability and effectiveness in intelligently managing network traffic. These findings substantiate the viability of our approach in enhancing network security, optimizing resource allocation, and delivering an enhanced user experience. The impressive accuracy achieved across most categories highlights the effectiveness of the Random Forest classification model in accurately categorizing DNS queries based on extracted features. The high accuracy for crucial categories such as Ads and Entertainment proves the model's reliability in handling essential network management tasks.

8.3.2 Real-Time Response

One of the critical requirements of our DNS traffic management system is real-time response to DNS queries. We are pleased to report that during our experiments, the system demonstrated exceptional real-time response capabilities. The system effectively intercepted DNS requests, extracted the required features, made predictions, and performed network management tasks with minimal delay. In figure 8.2

```
Domain: www.google.com.
URL Length: 15
Special Character Count: 3
Subdomain Number: 2
Query Parameter Number: 0
Hyphen Number: 0
Dot Number: 2
Number Count: 0
Entropy: 2.822579761842491
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:439:
  warnings.warn(
Predicted Category: Search Engine
```

Figure 8.2: Real time Response Sample 1

```
Domain: go.grammarly.com.
URL Length: 17
Special Character Count: 3
Subdomain Number: 2
Query Parameter Number: 0
Hyphen Number: 0
Dot Number: 2
Number Count: 0
Entropy: 3.0574760762899316
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:439:
  warnings.warn(
Predicted Category: Learning
```

Figure 8.3: Real time Response Sample 2

and 8.3, the log print of real time dns packet interception and prediction are shown.

Even when deployed on a modest virtual machine with limited hardware resources (2GB RAM, 1 core CPU), the system efficiently responded to DNS queries from multiple computers. This real-time responsiveness ensures smooth and seamless user experience, reducing latency and bottlenecks in the network.

8.3.3 Scalability

Scalability is a critical aspect of any network management system, as it should be able to handle increasing loads and adapt to changing demands. Our system proves to be highly scalable, thanks to its virtualized environment. The ability to upgrade system resources easily allows for accommodating higher traffic volumes and expanding the scope of the system without significant hardware or infrastructure changes.

Chapter 9

Evaluation and Validation

9.1 Validation of Results against Research Objectives

In this section, we validate the experimental results of our proposed DNS traffic management system against the defined research objectives. The system's performance and capabilities are evaluated in light of the stated research objectives to determine the effectiveness and feasibility of our approach.

9.1.1 Objective 1: Comprehensive Approach for Network Traffic Management

Research Objective 1 aimed to develop a comprehensive approach that combines DNS log analysis, data manipulation, visualization, and machine learning techniques for effective network traffic management. The experimental results affirm the achievement of this objective. By integrating DNS log analysis with machine learning, specifically utilizing the Random Forest classification model, our system effectively categorized DNS queries into various website categories. The feature extraction process successfully captured domain-specific attributes, allowing the model to make accurate predictions. Additionally, the seamless integration of the machine learning model into the SDN controller facilitated real-time traffic management based on the predicted categories. Thus, our approach successfully fulfills Objective 1 by providing a robust and versatile solution for network traffic management.

9.1.2 Objective 2: Feasibility of Browsing Category Prediction and SDN Integration

Objective 2 focused on investigating the feasibility of predicting browsing categories based on DNS log analysis and integrating the resulting models into SDN controllers for real-time traffic management. The experiment's high prediction accuracy across various website categories, including CDN, Search Engine, Social Network, and more, demonstrates the feasibility of using DNS log analysis to predict browsing categories. The Random Forest classification model proved to be a suitable choice for accurate categorization, enabling efficient traffic management decisions. Moreover, the successful integration of the machine learning model with the SDN controller

validates the feasibility of our approach in enabling dynamic and intelligent network traffic management in real-time. Thus, Objective 2 has been successfully validated, showing the viability of our solution in bridging DNS analysis and SDN-based traffic management.

9.1.3 Objective 3: Enhancing Firewall Capabilities for URL Identification and Policy Enforcement

Objective 3 aimed to enhance URL identification and policy enforcement capabilities in firewall systems by studying patterns, modifications, and variations in domain names and URL structures. The experimental evaluation confirmed the achievement of this objective. By extracting relevant features from DNS packets and predicting website categories, our system effectively identified URLs and domains based on their characteristics. The machine learning model's high accuracy in categorizing URLs, especially in essential categories such as Ads and Entertainment, reinforces the system's ability to enhance firewall policies. The capability to perform policy enforcement, such as blocking specific categories or rerouting traffic based on predictions, highlights the potential impact of our solution on enhancing network security and resource allocation. As such, Objective 3 has been successfully validated, showcasing our system's ability to bolster firewall capabilities through DNS log analysis and machine learning.

9.2 Discussion of Limitations and Potential Improvements

While our proposed DNS traffic management system demonstrates promising results, it is essential to acknowledge its limitations and explore potential avenues for improvement. This section discusses the identified limitations and proposes possible enhancements for future iterations of the system.

9.2.1 Limitations

Limited Feature Set

The current implementation relies on a limited set of features extracted from DNS packets for website categorization. While these features showed high prediction accuracy in our experiments, incorporating more diverse and detailed attributes may further enhance the model's categorization capabilities. Exploring additional features, such as web page content analysis, user behavior patterns, and URL path structures, could provide deeper insights into website categories and improve the system's accuracy.

Limited Training Dataset

The performance of machine learning models heavily depends on the quality and size of the training dataset. In our experiments, we utilized a moderately-sized dataset for training the Random Forest classification model. Expanding the training dataset

with more diverse and representative samples could lead to better generalization and further improve prediction accuracy.

Single DNS Server Deployment

Our experimental setup utilized a single DNS server for intercepting and analyzing DNS queries. While this approach is suitable for initial evaluation, in real-world scenarios with extensive network traffic, deploying multiple distributed DNS servers may be necessary. Distributed deployment can help handle higher request loads and provide redundancy for improved reliability.

Overhead of Real-Time Analysis

The real-time analysis and prediction of DNS queries impose additional computational overhead on the DNS server. While our experiments demonstrate efficient real-time response times, increasing the number of concurrent requests may affect the system's responsiveness. Optimizing the code, utilizing hardware acceleration, or exploring more efficient machine learning algorithms could mitigate this overhead.

9.2.2 Potential Improvements

Dynamic Feature Selection

Implementing a dynamic feature selection mechanism based on traffic patterns and query characteristics can optimize feature extraction for specific website categories. By adapting the feature set for different categories, the model can focus on the most relevant attributes, leading to improved accuracy and reduced computational load.

Incremental Model Updates

To account for dynamic changes in website categories and user behavior, an incremental learning approach can be implemented. Incremental model updates enable the system to adapt to new data without retraining the entire model. This would enhance the system's ability to handle emerging website categories and evolving network traffic trends.

Multi-Model Ensemble

Combining predictions from multiple machine learning models, such as Random Forest, Support Vector Machines, and Neural Networks, through an ensemble approach can potentially improve prediction accuracy. Each model may excel in different website categories, and combining their strengths could yield more robust and reliable categorization.

Hybrid DNS Analysis and Deep Packet Inspection

Integrating DNS log analysis with deep packet inspection (DPI) [60] techniques can offer a more comprehensive view of network traffic. DPI allows for analysis of packet payloads and protocols, providing valuable insights into encrypted or obfuscated traffic. By combining DNS analysis and DPI, the system can achieve higher accuracy in categorizing complex or encrypted website traffic.

Scalability Testing in Large Network Environments

To validate the system's scalability in handling extensive network traffic, testing on large-scale network environments with diverse user behaviors and traffic patterns is essential. Conducting experiments in enterprise or data center environments would provide valuable insights into the system's performance under real-world conditions.

Chapter 10

Conclusion

In this research, we presented a comprehensive approach to address the challenges of network traffic management through DNS log analysis, machine learning techniques, and SDN integration. Our primary research objectives were to develop an efficient network traffic management system, investigate the feasibility of predicting browsing categories, and enhance URL identification and policy enforcement capabilities. Through a systematic exploration of these objectives, we have achieved significant milestones and delivered promising results.

The experimental evaluation of our proposed solution yielded remarkable outcomes. We analyzed the prediction accuracy, real-time response, and scalability aspects of our system. The Random Forest classification model showcased exceptional performance in categorizing DNS queries, achieving high accuracy levels for various website categories. Notably, we achieved 100% accuracy in classifying Ads and Entertainment, while maintaining accuracy rates of 98.57% for Search Engines, 87.5% for Social Networks, and 87.21% for CDNs. These results validate the efficacy of our machine learning-based approach in intelligently managing network traffic and enforcing policies effectively.

The real-time response demonstrated by our system showcased its ability to swiftly process DNS queries and implement network management decisions without noticeable delays. Our solution effectively responded to DNS queries from multiple computers, even while operating on a virtual machine with modest hardware resources. This real-time responsiveness ensures a seamless user experience and enhances overall network efficiency. Our system exhibited excellent scalability, thanks to its virtualized environment. The ability to readily upgrade system resources allows for accommodating increased traffic loads and future expansion without major infrastructure changes. This scalability feature empowers our solution to adapt to changing network demands and ensure consistent performance.

The proposed approach not only met but also surpassed the research objectives. Our findings affirm that the integration of DNS log analysis, machine learning, and SDN provides a robust foundation for effective network traffic management. The capability to predict browsing categories in real-time and enforce appropriate policies contributes significantly to network security, resource optimization, and enhanced user experiences. While our research has demonstrated considerable success, it is essential to acknowledge some limitations. One of the key limitations is the reliance on a pre-trained model, which may require periodic updates to maintain relevance in an evolving internet landscape. Additionally, the efficacy of the system may

be influenced by the size and diversity of the training dataset. Future research could explore the utilization of dynamic feature selection techniques and incremental learning to address these challenges and improve accuracy further.

In conclusion, our work presents a compelling solution for network traffic management by leveraging DNS log analysis, machine learning, and SDN integration. The achieved results, analysis, and fulfillment of research objectives underscore the practicality and effectiveness of our approach. We believe that this research contributes significantly to the field of network security, traffic optimization, and DNS-based policy enforcement. Our findings open avenues for future research, paving the way for more sophisticated, adaptive, and efficient network traffic management systems. As we continue to explore new technological advancements, we envision our approach making a positive impact on network infrastructures, user experiences, and cybersecurity in the dynamic and ever-evolving digital landscape.

Bibliography

[1] C. Huang, D. A. Maltz, J. Li, and A. Greenberg, "Public dns system and global traffic management," in *2011 Proceedings IEEE INFOCOM*, IEEE, 2011, pp. 2615–2623.

[2] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.

[3] S. Fernandes and J. Bernardino, "What is bigquery?" In *Proceedings of the 19th International Database Engineering & Applications Symposium*, 2015, pp. 202–203.

[4] Search Engine Journal. "Google looker studio beginner guide." (n.d.), [Online]. Available: https://www.searchenginejournal.com/google-looker-studio-beginner-guide/471369/.

[5] K. Kirkpatrick, "Software-defined networking," *Communications of the ACM*, vol. 56, no. 9, pp. 16–19, 2013.

[6] F. He, Y. Deng, and W. Li, "Coronavirus disease 2019: What we know?" *Journal of medical virology*, vol. 92, no. 7, pp. 719–725, 2020.

[7] T. Berners-Lee, L. Masinter, and M. McCahill, "Uniform resource locators (url)," Tech. Rep., 1994.

[8] Z. Sun, T. Guo, S. Luo, *et al.*, "Dns request log analysis of universities in shanghai: A cdn service provider's perspective," *Information*, vol. 13, no. 11, p. 542, 2022.

[9] J. Li, X. Ma, L. Guodong, *et al.*, "Can we learn what people are doing from raw dns queries?" In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 2240–2248.

[10] Q. Lai, C. Zhou, H. Ma, Z. Wu, and S. Chen, "Visualizing and characterizing dns lookup behaviors via log-mining," *Neurocomputing*, vol. 169, pp. 100–109, 2015.

[11] J. H. Wang, C. An, and J. Yang, "A study of traffic, user behavior and pricing policies in a large campus network," *Computer Communications*, vol. 34, no. 16, pp. 1922–1931, 2011.

[12] S. Zhang, H. Zhang, J. Yang, G. Song, and J. Wu, "Measurement and analysis of adult websites in ipv6 networks," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, 2019, pp. 1–6.

[13] H. Hu, Y. Wen, T.-S. Chua, *et al.*, "Community based effective social video contents placement in cloud centric cdn network," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2014, pp. 1–6.

[14] H. Hours, E. Biersack, P. Loiseau, A. Finamore, and M. Mellia, "A study of the impact of dns resolvers on cdn performance using a causal approach," *Computer Networks*, vol. 109, pp. 200–210, 2016.

[15] W. Jiang, "Internet traffic prediction with deep neural networks," *Internet Technology Letters*, vol. 5, no. 2, e314, 2022.

[16] Q. Xie, T. Guo, Y. Chen, Y. Xiao, X. Wang, and B. Y. Zhao, "Deep graph convolutional networks for incident-driven traffic speed prediction," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1665–1674.

[17] H. Shi and K. Iwasaki, "Classification of dns queries for anomaly detection," in *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, IEEE, 2013, pp. 130–131.

[18] S. K. Singh and P. K. Roy, "Detecting malicious dns over https traffic using machine learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, IEEE, 2020, pp. 1–6.

[19] P. Satam, H. Alipour, Y. Al-Nashif, and S. Hariri, "Dns-ids: Securing dns in the cloud era," in *2015 International Conference on Cloud and Autonomic Computing*, IEEE, 2015, pp. 296–301.

[20] R. Nakamura, Y. Sekiya, D. Miyamoto, K. Okada, and T. Ishihara, "Malicious host detection by imaging syn packets and a neural network," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, 2018, pp. 1–4.

[21] I. Voronov and K. Gnezdilov, "Determining os and applications by dns traffic analysis," in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, IEEE, 2021, pp. 72–76.

[22] Y. Li, A. Dandoush, and J. Liu, "Evaluation and optimization of learning-based dns over https traffic classification," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, 2021, pp. 1–6.

[23] K. Shima, R. Nakamura, K. Okada, T. Ishihara, D. Miyamoto, and Y. Sekiya, "Classifying dns servers based on response message matrix using machine learning," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 1550–1551.

[24] C. Han, Y. Zhang, and Y. Zhang, "How to detect benign domains based on "lonesome" dns traffic," in *2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT)*, IEEE, 2020, pp. 155–159.

[25] K. Ishibashi and K. Sato, "Identifying dns anomalous user by using hierarchical aggregate entropy," *IEICE Transactions on Communications*, vol. 100, no. 1, pp. 140–147, 2017.

[26] A. Moubayed, M. Injadat, and A. Shami, "Optimized random forest model for botnet detection based on dns queries," in *2020 32nd international conference on microelectronics (ICM)*, IEEE, 2020, pp. 1–4.

[27] M. Fejrskov, J. M. Pedersen, and E. Vasilomanolakis, "Detecting dns hijacking by using netflow data," in *2022 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2022, pp. 273–280.

[28] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, Springer, 2019, pp. 758–763.

[29] M. Wainberg, B. Alipanahi, and B. J. Frey, "Are random forests truly the best classifiers?" *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3837–3841, 2016.

[30] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, vol. 18, no. 60, pp. 1–8, 2006.

[31] F.-J. Yang, "An implementation of naive bayes classifier," in *2018 International conference on computational science and computational intelligence (CSCI)*, IEEE, 2018, pp. 301–306.

[32] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 529–536.

[33] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, pp. 1–19, 2017.

[34] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient knn classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.

[35] Y.-Y. Song and L. Ying, "Decision tree methods: Applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[36] S. B. Kotsiantis, "Decision trees: A recent overview," *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[37] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron: Architecture optimization and training," 2016.

[38] H. Taud and J. Mas, "Multilayer perceptron (mlp)," *Geomatic approaches for modeling land change scenarios*, pp. 451–455, 2018.

[39] R. E. Schapire, "Explaining adaboost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, Springer, 2013, pp. 37–52.

[40] C. Ying, M. Qi-Guang, L. Jia-Chen, and G. Lin, "Advance and prospects of adaboost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, 2013.

[41] F. Liu and Y. Deng, "Determine the number of unknown targets in open world based on elbow method," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 986–995, 2020.

[42] M. Cui *et al.*, "Introduction to the k-means clustering algorithm based on the elbow method," *Accounting, Auditing and Finance*, vol. 1, no. 1, pp. 5–8, 2020.

[43] S. Aranganayagi and K. Thangavel, "Clustering categorical data using silhouette coefficient as a relocating measure," in *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, IEEE, vol. 2, 2007, pp. 13–17.

[44] T. M. Kodinariya, P. R. Makwana, *et al.*, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[45] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE access*, vol. 8, pp. 80 716–80 727, 2020.

[46] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[47] J. Gan and Y. Tao, "Dbscan revisited: Mis-claim, un-fixability, and approximation," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 519–530.

[48] T. Huang, H. Peng, and K. Zhang, "Model selection for gaussian mixture models," *Statistica Sinica*, pp. 147–169, 2017.

[49] M. A. Carreira-Perpinán, "A review of mean-shift algorithms for clustering," *arXiv preprint arXiv:1503.00687*, 2015.

[50] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[51] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117–123, 2016.

[52] V. Vovk, "The fundamental nature of the log loss function," *Fields of logic and computation II: Essays dedicated To Yuri Gurevich on the Occasion of His 75th Birthday*, pp. 307–318, 2015.

[53] M. A. Salam, A. T. Azar, M. S. Elgendy, and K. M. Fouad, "The effect of different dimensionality reduction techniques on machine learning overfitting problem," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021. DOI: 10.14569/IJACSA.2021.0120480. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.0120480.

[54] A. Moubayed, M. Injadat, and A. Shami, "Optimized random forest model for botnet detection based on dns queries," in *2020 32nd International Conference on Microelectronics (ICM)*, 2020, pp. 1–4. DOI: 10.1109/ICM50269.2020.9331819.

[55] M. Karakus and A. Durresi, "Quality of service (qos) in software defined networking (sdn): A survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[56] M. Q. Ali, E. Al-Shaer, and T. Samak, "Firewall policy reconnaissance: Techniques and analysis," *IEEE transactions on information forensics and security*, vol. 9, no. 2, pp. 296–308, 2014.

[57] Y. Li, W. Li, and C. Jiang, "A survey of virtual machine system: Current technology and future trends," *Electronic Commerce and Security, International Symposium*, vol. 0, pp. 332–336, Jul. 2010. DOI: 10.1109/ISECS.2010.80.

[58] M. Tabassum and K. Mathew, "Software evolution analysis of linux (ubuntu) os," Aug. 2014. DOI: 10.13140/2.1.2331.5201.

[59] T. Jinmei and P. Vixie, "Implementation and evaluation of moderate parallelism in the bind9 dns server," in *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, ser. ATEC '06, Boston, MA: USENIX Association, 2006, p. 12.

[60] R. T. El-Maghraby, N. M. Abd Elazim, and A. M. Bahaa-Eldin, "A survey on deep packet inspection," in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 2017, pp. 188–197. DOI: 10.1109/ICCES.2017.8275301.