2024-05-01

Understanding The Limits Of Deep Packet Inspection For Network Traffic Classification

Herman Ramey
University of Texas at El Paso

UNDERSTANDING THE LIMITS OF DEEP PACKET INSPECTION FOR NETWORK TRAFFIC CLASSIFICATION

HERMAN FOSTON RAMEY

Master's Program in Computer Engineering

APPROVED:

Michael P. McGarry, Ph.D., Chair

Yuanrui Sang, Ph.D.

Eric D. Smith, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

UNDERSTANDING THE LIMITS OF DEEP PACKET INSPECTION FOR NETWORK TRAFFIC CLASSIFICATION

by

HERMAN FOSTON RAMEY, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2024

Acknowledgements

I would like to begin by thanking my Thesis advisor, Dr. McGarry, for his continued support throughout this research process. Through his mentorship, I was able to acquire valuable skills that I hope to apply to all facets of my life moving forward. I also want to thank Luis Vergara-Rodriguez, the initial contributor to the Human App Labeling System. Without his efforts, none of this work would likely be possible. Lastly, I would like to thank the committee members for taking an interest in the research presented in this document, and taking the time to oversee my Thesis defense.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Communication Networks

As the competition between the United States and the Soviet Union to develop aerospace capabilities and conduct explorative missions into space began to intensify during the 1960s, the government's attention turned towards advancing the field of science. This demand for technological advances involved developing communication networks that were more efficient and less costly on resources compared to the circuit switched networks at the time. During this time, communications data was being sent over circuit switched networks, which are networks in which dedicated paths must be made between the endpoints for the entire duration of the connection if data was to be sent or received [1]. To address the inefficiencies of circuit switched networks, researchers in the United States and United Kingdom began working towards an idea that we know as packet switching today.

In the case of packet switched networks, the data that is to be sent over the network is broken up into packets and sent to its destination, one at a time. Along the way, routers and switches make forwarding decisions for these packets based on fields in the packets header to ensure the data is transferred in the most efficient manner possible [1]. Packet switching ensures that the transmission media which are connecting two communicating nodes are not required to be occupied during the entire length of the connection. The groundbreaking work conducted by these researchers thus laid the foundation for packet switched networks, and ultimately led to the creation of the Advanced Research Projects Agency Network (ARPANET). ARPANET played a pivotal role in the evolution of the Internet because it was the first packet switching network that was capable of interconnecting computers, and

people, together over vast distances [2].

Packet switching was further developed by many other scientists and researchers after those that worked on ARPANET. However, the research conducted in the field of packet switching ultimately contributed to the birth of the Internet, and consequently transformed a "network of networks" which was primarily used by a select demographic of individuals across the country, to a widely used, influential entity in today's society which affects various aspects of everyday life, including communication and commerce. The increase in the amount of people choosing to do their shopping online due to the level of convenience shopping websites have to offer led to the e-commerce industry that in the United States alone accounts for $285.2 billion in fourth quarter 2023 [3]. Furthermore, as the Internet becomes a tool commonly used for every day tasks, the Internet has made remote work more feasible for companies to offer their employees opportunities to have better work-life balances, as well as empowering employers to have a wider range of potential hiring candidates to select from by using popular social media websites like LinkedIn and Glassdoor.

Today, over 60% of the world's population are users of the Internet [4], being used to conduct many tasks that may be completely unrelated to scientific research. That is to say, the Internet has become an extremely influential aspect of modern life that can be used to carry out everyday tasks with little to no work on the users end, by virtually anybody that has access to these resources. With its growing popularity, the Internet is also no longer simply communication networks being interconnected; the emergence of devices which leverage the benefits of the Internet helped coin a new term, Internet of Things (IoT), referring to a new plethora of devices which can connect to the Internet. Smart watches, smart refrigerators, gaming consoles and smart thermostats are just a few examples of the types of IoT devices that exist today. As of 2023, an estimated 15.4 billion devices exist that are capable of connecting to the Internet [5].

1.1.1 Managing Networks

Within the structure of the Internet's interconnected system, each element requires oversight from one or more individuals to maintain operation and peak efficiency. Network management can be summarized by the FCAPS model [6], which consists of five distinct areas:

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management

Detecting, isolating and fixing faults in the network is an aspect of fault management. Network Traffic Classification (NTC) can play a role in aiding fault management by organizing network traffic according to the applications that produced it. For instance, utilizing an application label as a feature can help identify network faults when using machine learning algorithms. Managing configurations of a network involves setting up devices and making sure they function as intended. NTC plays a role in configuration management by enabling network administrators to enforce rules and automated reactions customized for applications. For example, a network can be set up to restrict the amount of bandwidth allocated for video streaming, to a percentage like limiting it to 25%. Account management, in the realm of network management, encompasses observing network usage trends. This task involves overseeing data consumption, monitoring resource distribution and overseeing expenses linked to network operations. In performance management, the main goal is to keep track of and enhance performance standards. This involves evaluating metrics such as network latency, throughput and packet loss. Additionally, performance management includes pinpointing and addressing issues that may be slowing down the network to make

it more efficient from the users perspective. Lastly, managing the security of a network is focused on establishing security measures to safeguard against uninvited mainframe accesses and cyber attacks. This includes designing firewalls, intrusion detection systems and encryption protocols. Network managers are responsible for managing these aspects, and by categorizing network traffic based on the applications that generated it, automation tools can simplify the management of these areas and ease the workload of the network management organization or individual. This categorization allows for policies and automated responses specific to applications to be implemented, thus increasing effectiveness and decreasing manual intervention required in managing networks.

1.2 Network Traffic Classification

Systems which utilize Network Traffic Classification observe raw network data and sort the data into categories, sometimes also referred to as classes, in real-time using criteria specific to the NTC system being utilized. Various factors, like source and destination IP addresses, port numbers, transport protocols and payload content can influence the classification process and be used as filtering criteria. NTC has had significant impact on the networking and communications community over the years, showcasing its versatility across a broad range of use cases. NTC can prove to be highly beneficial for network managers that wish to add an extra layer of security to their network. In this context, NTC can be used in content filters to block unwanted traffic flowing through a network, a layer which ISPs have been steered towards implementing over the years due to the high volume of Internet traffic today[7]. For example, if a company wishes to ensure users of their local area network (LAN) cannot access certain websites, NTC and the benefits it has to offer complement this network traffic management process. In addition to filtering content on a network, NTC can also be used to identify patterns in application behavior to deliver personalized Quality of Service (QoS) features to the user by prioritizing certain types of traffic over others [8]. Essentially, by having a system set in place that can

classify applications based on the traffic they're generating, an order of precedence can be implemented. By leveraging the benefits of these QoS features, applications that necessitate some level of latency and bandwidth guarantees (video streaming, voice over IP (VoIP), etc.) can take priority in the network during high congestion events [9]. Lastly, NTC can also be used in anti-malware products to detect malicious behavior. Thus, in the case of botnet attacks on the network, the anti-malware product can use NTC to identify the traffic these attackers are generating, and plan accordingly [10].

1.2.1 Current Methods for NTC and their Limitations

Current literature in the field of NTC identifies three main approaches to classifying network traffic: transport layer port conventions, deep packet inspection (DPI), and the utilization of machine learning algorithms on network traffic features ([7],[11], [12], [13], [14]). Port based, which is the earliest and arguably most simple approach to NTC, classifies by looking into the packet header for TCP/ UDP port number and uses a database of port numbers maintained by the Internet Assigned Numbers Authority (IANA) to classify the traffic [15]. As a simple example, if the packet header contains port number 443, this would map to the HTTPS protocol, and port 80 to HTTP. However, this approach is only accurate when standard port conventions are followed, which is not always the case. Applications can hide their identities by masquerading under well-known port numbers such as port 443 or 80 or take advantage of the dynamic range of ports by randomly assuming port numbers so the identity of the application can essentially be different at any given time. Because of the inaccuracies that the port based approach presents, a more refined approach was eventually introduced, DPI. As opposed to the port based approach which merely uses the packet header for information extraction, the DPI approach looks even deeper into the data and inspects the payload itself to find a byte signature that matches a signature within an external DPI-database and classifies according to these signatures [13]. It is due to the nature of this approach that it is also referred to as signature based or payload based approach.

Although DPI has proven to be more accurate compared to the port based approach, there are shortcomings to the approach that must be acknowledged [7]. The first is that DPI is resource intensive because the data within the packet has to first be accessed, then processed, stored, and of course compared to match signatures in the database. Depending on the size of the dataset that must be classified, this can be very demanding on whatever computing resources are available. Second, DPI may present significant legal concerns due to looking at user's private network traffic data. Lastly, a very significant shortfall to DPI is that the approach does not work on encrypted traffic. The purpose of encryption is to protect private data being transferred within a network, or between networks, from being intercepted and breached. Because of this, even by analyzing the payload itself, DPI will fail to identify the causal application generating the encrypted traffic.

Machine learning (ML) algorithms can also be applied for NTC use cases and have shown promise in recent years when compared to the port and signature based approaches [13]. This particular approach involves utilizing flow features such as source IP, destination IP, source port, destination port, number of packets and bytes, the duration of the flow, or any other combination of features that can be acquired either through explicit flow-level metadata extraction or feature engineering, to train an ML classifier to generalize on never-before-seen traffic data. ML classifiers can be supervised, unsupervised, or semi-supervised, depending on whether or not the labels that represent the ground truth are readily available. In unsupervised and semi-supervised problems, some or all of the dataset does not contain labels, and is regarded as a more exploratory approach to NTC. This is because it is the model's responsibility to categorize the traffic based on the inherent structure of the data, and if the model is able to exploit any hidden patterns in the data without being explicitly told what each sample in the dataset should be attributed to. On the other hand, when following the supervised learning approach, the ground truth labels for the training dataset must be known beforehand. To put it another way, in the supervised learning model, input values along with the features and expected output are first given to the classifier to train it to make predictions, or generalize, on new data.

1.3 Thesis Contribution

Many authors that have conducted research in the field of NTC claim that when utilizing the ML supervised learning approach, acquiring a training dataset that contains labels which are both accurate and representative of real network environments is crucial to yielding satisfactory results ([16], [17], [14]). However, acquiring enough labeled training data to build a high-performing classifier has proven to have its own challenges. This may involve tasking a subject matter expert to manually label all traffic in a controlled network environment, which is potentially very time consuming when trying to build large enough datasets for classification purposes [18]. Additionally, the inherent tendency for humans to make errors [17] compared to automated techniques consequently jeopardizes accuracy of the labeling process, ultimately degrading the overall performance of the classifier.

An alternative approach to manually labeling traffic is using DPI tools that automate the labeling process, which mitigates both the shortfalls to manually labeling highlighted previously. Open-source libraries such as OpenDPI [19] and nDPI [20] offer software to classify traffic based on the deep packet inspection method, which can then be integrated to other network monitoring tools like pmacct [21] (discussed in further detail in 3.0.2). However, if one were to use DPI to generate labels for the training dataset, unfortunately there is no guarantee that these labels are 100% "accurate" or even the absolute ground truth. Thus, the questions that the research presented in this document addresses are: How accurate are labels generated using DPI compared to the ground truth, and are there any viable solutions to acquiring labeled network data?

The work presented in this thesis document presents a human application labeling system which addresses the shortcomings of the existing literature by very significantly improving the distinction between the network traffic that should be labeled with the human supplied label and which traffic should not be. The labeling system introduced in this research operates in a way to allow the human network user to label traffic during their normal every day use of the network rather than as part of a contrived experiment.

Lastly, this thesis provides an analysis of deep packet inspection, specifically that provided by the nDPI library. Genuine ground truth labels provided by a human user are used to facilitate this analysis that reveals and/or confirms the shortcomings of deep-packet inspection beyond the privacy concerns it invokes.

1.4 Thesis Outline

Chapter 2 aims to provide brief tutorials on relevant concepts, as well as present a survey of the existing literature on NTC. Chapter 3 introduces the system developed to refine the labeling process and outlines the experimental plan implemented for this research. Chapter 4 will present and analyze the experimental results. Finally, Chapter 5 will conclude this document.

Chapter 2

Background and Related Works

Obtaining labelled datasets to train a supervised learning classifier is a major obstacle in the field of NTC today, and there have been several authors that touch on the subject in the hopes that the state of the art may be advanced ([17], [10], [9], [22], [23], [24]). This chapter aims to provide background knowledge on specific concepts integral to understanding the research presented in this document, as well as conduct a formal survey of the existing literature as it pertains to the approaches that have been employed in the field of NTC. Through the work discussed in this chapter, the techniques employed for NTC and the labelling process itself have been refined, but like all other research endeavors, there are of course shortfalls to each work. An additional objective to this chapter is to highlight any disadvantages researchers may have found in their work, and generally present their findings before moving onto Chapter 3 where specific solutions to the problem this thesis is trying to solve are discussed.

2.1 Network Flow Monitoring

Network flow monitoring, a concept that has its roots in the 1990's, is the process of aggregating packets being sent over a network based on distinctive data that can be extracted from the packet header such as destination IP, source IP, destination port, source port, and so on for the purposes of analysis or evaluation. Network flow monitoring can be broken down into two subcategories: active and passive network monitoring. Active network monitoring is the act of injecting traffic into a network in the hopes that this injected traffic will shed light on any issues that the network may be having, or to simply observe

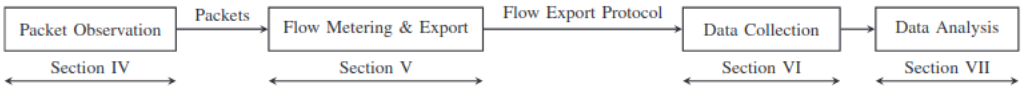


Figure 2.1: Architecture of typical flow monitoring configuration. Taken from [25].

how the network will behave under certain conditions. Passive monitoring, on the other hand, does not involve any traffic being intentionally injected into the network, but rather it involves using an observation point to observe traffic that already exists to measure or evaluate different aspects of the network. The resultant aggregation of packets, bound by a particular time interval, can then be referred to as a flow. To put it another way, if 60 packets are captured at the observation point, and 40 of them share the same properties and fall within the same time interval, this stream of data will collectively be combined into a flow because they share the same properties. The approach used in the experiments presented in this thesis uses the passive approach.

2.1.1 General Architecture

In [25], the general architecture for flow monitoring is outlined (see Figure 2.1). The architecture begins with an observation point, a point in the network that serves as the point where packets are captured. Next, during the flow monitoring and exporting stage, the packets are aggregated into their respective flows based on characteristics of the data such as IP numbers, port numbers and a specified time interval during the flow monitoring stage. During the exporting stage, the data is compiled into a single datagram to be transported using a particular transport protocol, UDP being the most widely used in the export process. The Metering and Exporting processes are closely related to one another, which is why the two stages may be combined into one. After the data has been exported to the collector, the data can then be processed and analyzed using either manual or automated techniques.

2.1.2 NetFlow and IPFIX

In [25], the authors cover two popular protocols that have been developed for network flow monitoring: NetFlow and IP Flow Information eXport (IPFIX). Introduced by Cisco in 1996, NetFlow is a proprietary technology used in flow monitoring and export that revolves around flow-based switching. In flow-based switching, tables referred to as flow caches are built that contain information about the network traffic being observed, organized into key fields (flow keys) and non-key fields. These keys assist in placing the packets into their respective flow entries based on the key. The key is essentially used to determine whether or not that flow is a new flow or if it should be combined with an existing flow. Using the flow cache, forwarding decisions are then made using the first packet in a flow, then all other packets that belong to the same flow simply follow suit. These flows, however, do expire, specifically during the Metering stage of the flow export workflow. Upon expiration, the flows are considered to be ready for export. Common reasons for flows to expire are related to timeout issues, such as no flows being reported over a specified time interval, or even if one single flow has been prolonged for an extended amount of time, usually 120 seconds to 30 minutes.

Alternatively, IPFIX, which is based on the more flexible NetFlow v9, is the standardized flow export protocol, introduced by the IETF in 2013. The authors of [25] elaborate on the features implemented by IPFIX compared to NetFlow, one of which is flow records being organized as IPFIX messages and templates. These messages may follow the simplified format illustrated in Figure 2.2. Common elements in an IPFIX message may be destinationIPv4Address, sourceIPv4Address, protocolIdentifier, and flowStartMilliseconds. These information elements (IEs) as they are referred to as, are maintained by the IANA, although new IEs may be created to accomplish application-specific tasks. Much like NetFlow, IPFIX utilizes flow caches for its flow exportation, following flow cache entry expiration guidelines as with NetFlow. Once the flows are deemed expired, they are ready for export. Transport of the flows can be implemented using UDP, same as NetFlow, except IPFIX has the options to also use transport protocols such as Stream Control Transmis-

sion Protocol (SCTP) or Transmission Control Protocol (TCP), one significant difference between IPFIX and NetFlow. Although, it is important to note that most of the time, UDP is used as the transport protocol in IPFIX.

ID	Name	Description
152	flowStartMilliseconds	Timestamp of the flow's first packet.
153	flowEndMilliseconds	Timestamp of the flow's last packet.
8	sourceIPv4Address	IPv4 source address in the packet header.
12	destinationIPv4Address	IPv4 destination address in the packet header.
7	sourceTransportPort	Source port in the transport header.
11	destinationTransportPort	Destination port in the transport header.
4	protocolIdentifier	IP protocol number in the packet header.
2	packetDeltaCount	Number of packets for the flow.
1	octetDeltaCount	Number of octets for the flow.

Figure 2.2: An example subset of common IEs. Taken from [25].

2.2 ML Algorithms: A Tutorial

Machine learning algorithms have become a powerful tool for researchers worldwide, offering a variety of applications that have contributed to the advancement of several industries:

- The healthcare industry can be enhanced with machine learning techniques to be used in cases like designing systems to streamline diagnoses, identify patterns in patient parameters to predict a patient's risk for a particular disease, improve the quality of medical imaging, and in genetic engineering [26].

- In the finance industry, factor models may be supplemented with machine learning models to provide portfolio pricing and asset management, and to predict the future

prices of assets based on time series data [27].

- Using data analytics combined with machine learning algorithms, companies can deliver personalized advertisements based on customers past behavior and aid in the marketing process by making use of the significant amount of data provided by their customers [28].

These are just a few examples of the industries that stand to benefit from machine learning algorithms, with many more today using these techniques to improve their respective areas.

Machine learning can be divided into 3 main categories: Supervised, unsupervised, and semi-supervised (see Figure 2.3). However, it is important to note that there are more categories of machine learning such as reinforcement learning, deep learning and transfer learning. The three approaches outlined in Figure 2.3 are those most relevant to the research presented in this document, so only these approaches are covered.

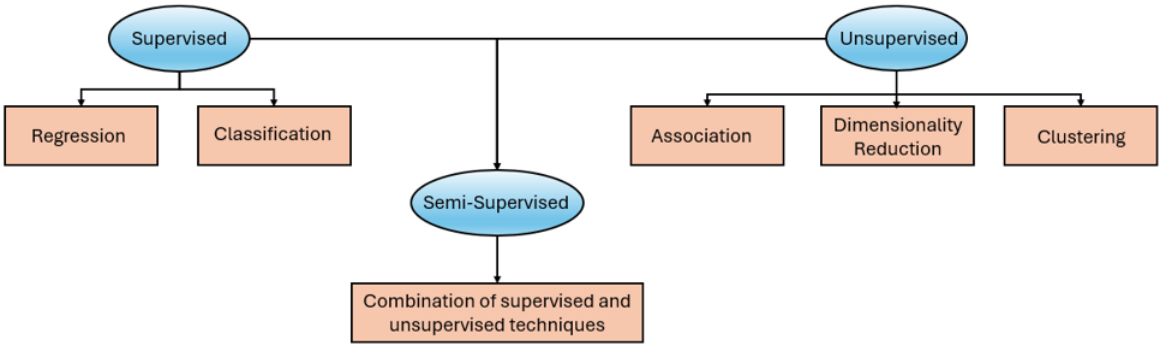


Figure 2.3: Breakdown of approaches to ML.

2.2.1 Supervised Learning

Supervised learning is a technique used in machine learning algorithms that is mainly used for classification and regression problems. In a classification problem, the output is expected to be a discrete value which represents a class [29]. This can be a simple problem that only

involves predicting "positive" or "negative", such as whether a patient is infected with a viral infection (binary classification), or multiple classes, such as types of dogs or plants (multi-class classification). On the other hand, a regression problem yields a continuous value to predict things like stock market prices or the amount of energy used in a household. The main difference between the two algorithms is that the training data for classification use cases must contain categorical target variables, whereas the training data for regression use cases must contain numerical target variables. This document will mainly focus on the classification algorithm because it is most relevant to the research presented.

To implement a machine learning model which uses supervised learning, the first step is acquiring a training dataset that contains a label for every example that will be shown to the classifier during training. Next, relevant features from the dataset that are determined to have significant influence on the model accuracy are chosen, and feature reduction techniques are typically also employed to reduce the number of redundant or insignificant features in the dataset. For example, if a model is trying to predict whether a patient is positive or negative for a disease, some features that may be relevant might be age, weight, or family history; some irrelevant features would be things like favorite color or other unrelated demographic information such as geographic location. The dataset is then split into a training set, testing set, and sometimes a validation set depending on the application scenario. Once the data has been split, the model can then be trained, evaluated, and optimized through hyper-parameter tuning and potentially further feature engineering.

Training a classifier using the supervised learning approach can generally be split into two phases: training and testing. The training phase is comprised of allowing the classifier to know which input values correlate to which label, then the model will infer on the dataset to generalize on new data. During training, the model will attempt to make predictions, then a loss function is applied to the model so that it can adjust its predictions to reduce the amount of prediction errors as much as possible. Once this loss has been minimized as much as possible, the testing phase ensues. This phase exists to evaluate how close the classifiers output is to the ground truth if tasked with generalizing on data it has never seen before;

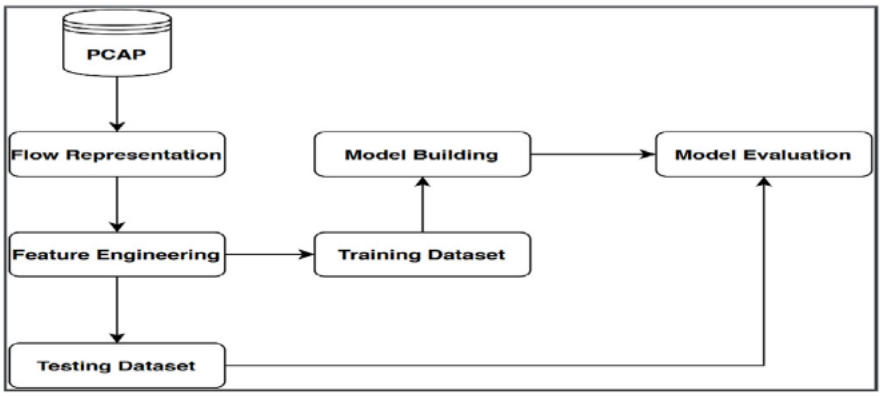


Figure 2.4: Supervised learning approach to training classifier on flow features. Taken from [7]

this process of evaluating the generalization capabilities of the classifier using the ground truth labels is what makes the key distinction between the supervised learning process and the unsupervised learning process. After the model has been thoroughly trained and tested, it is ready for deployment and can generalize on new data. The process is illustrated in the context of network flows in Figure 2.4 .

To evaluate a supervised learning model after it has been trained, 4 evaluation metrics exist: accuracy, recall, precision, and F-Measure, also referred to as F1-score. Accuracy, represented by the formula in 2.1, measures the number of correct classifications made by the model out of all of the attempts made by the model. This metric is rarely used on its own, as it is not representative of imbalanced datasets. Therefore, precision, recall and F1-score (2.2, 2.3 and 2.4) can be used to complement the evaluation process.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{2.1}$$

$$Precision = (TP)/(TP + FP) \tag{2.2}$$

$$Recall = (TP)/(TP + FN) \tag{2.3}$$

$$F1 - Score = 2 * (Precision * Recall)/(Precision + Recall) \tag{2.4}$$

Precision and recall are measured by class in multiclass classification problems to keep the process simple. Thus, when constructing a confusion matrix like that in Figure 2.5, true positives (TP) indicate when the classifier predicts a class that matches the ground truth label for the class that you are evaluating, whereas false positives (FP) are when the classifier predicts the class that you are evaluating, but the ground truth for that sample is something other than the class you are evaluating. On the other hand, true negatives (TN) are when the classifier predicts something other than the class that you are evaluating when the ground truth for that sample is, in fact, not the class label you are evaluating. False negatives (FN) represent the predictions made by the classifier where the class label predicted is something other than the class you are evaluating, but the ground truth is the class you are evaluating. F1-score is simply the harmonic mean of both precision and recall.

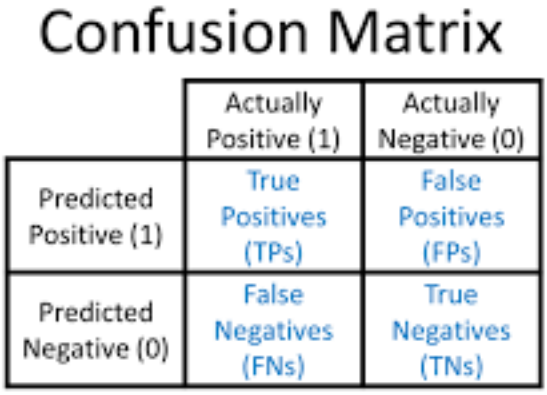


Figure 2.5: Example confusion matrix

2.2.2 Unsupervised Learning

As opposed to the supervised learning approach, the unsupervised learning approach begins with an unlabeled dataset, and the output is generally unknown at the start of the training process. Due to the nature of this approach, unsupervised learning does not follow the same steps during the training process as supervised because there are no labels for the model to use. The purpose of this approach is to find hidden patterns in the data, hence why it is referred to as exploratory analysis. In other words, given a dataset with no labels, an unsupervised learning approach will generate its own interpretation of the data, all with minimal human intervention. As observed in Figure 2.3, the main uses of unsupervised learning are for association rule mining, dimensionality reduction, and clustering problems.

Association rule mining, also referred to as pattern mining, is used when the goal is creating rules that describe associations within the data. For instance, a rule that a pattern mining model may identify is that customers at a grocery store who buy milk usually buy cookies as well. If this were a real grocery store, this company can use this information to potentially make more money by placing the cookies near the milk and exploiting this buying behavior. Dimensionality reduction is a preprocessing technique developed to reduce noise in the training dataset. In dimensionality reduction, the dimensions of the input dataset are reduced by capturing only the most important features, while maintaining as much of the original information as possible. Lastly, clustering problems arise when one wishes to find categorical groups in the data by grouping together data points based on similarity or distance metrics. Regardless of the problem, because there is no ground truth to evaluate the models' predictions against, it is not as straightforward to measure the accuracy of the models' output. Nonetheless, this approach can be very valuable in shedding light on the underlying structure of the given dataset.

2.2.3 Semi-Supervised Learning

By observing Figure 2.3, it can be seen that a third learning approach exists, the semi-supervised approach. Semi-supervised learning is very similar to supervised learning; however, the key difference is that this approach can handle unlabeled data as well. This can be a very useful technique when only a small portion of the dataset is labeled due to restrictions in the labeling process, or if it is too costly on resources to generate labels on the rest of the dataset, common disadvantages to supervised learning. In either case, semi-supervised learning is powerful in that it leverages the benefits of both supervised and unsupervised learning, while also mitigating these disadvantages.

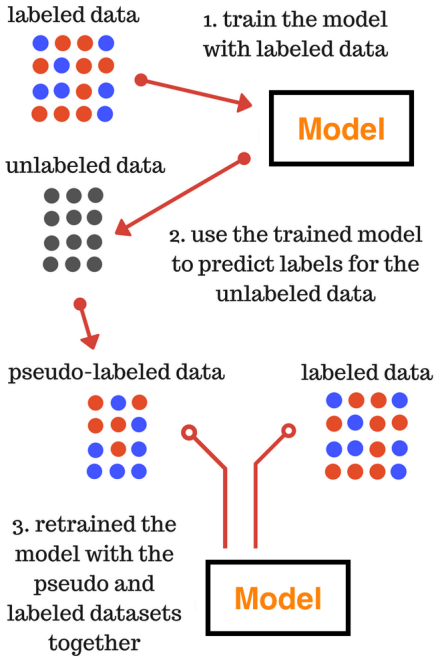


Figure 2.6: Illustration of how pseudo labels are generated. Taken from [30]

One popular example of semi-supervised learning in action is pseudo labeling. Pseudo labeling involves training a model with labeled data, then using this trained model to predict pseudo labels for a larger, unlabeled dataset. The newly generated pseudo labels with the highest confidence are then added to the labelled dataset, and this new dataset is

used to train a new model. This process can be repeated as many times as necessary until the model reaches some convergence threshold to yield the highest performance possible from the new model (See Figure 2.6). Clearly, this pseudo labeling technique has high utility in problems like image and text classification.

2.2.4 Concept Drift

Concept drift is a significant obstacle in the field of machine learning today and remains an active area of research. Concept drift refers to the idea that as time progresses, the effectiveness of the patterns learned by the model will decay as a result. This means that a model may perform rather well initially, but then if the data that it is being asked to infer on is evolving into something that the model cannot reconcile with the training it received, then the performance of the model may suffer. A simple example that illustrates concept drift is in the stock market. In this particular sector, financial trends and behaviors can change very quickly over time, which is why this is such a difficult field to perform data analytics in. These changes could be because of factors such as what consumers are buying, how companies are doing, or how investors are feeling. When confronted with these changes in market behavior, a machine learning model that has been trained on stock data from two or three years ago might find it challenging to accurately infer on stock data today.

2.3 Network Traffic Classification (NTC)

2.3.1 Port Based Approach

Briefly discussed in Chapter 1, the port based approach has been deemed the simplest and earliest approach in the field of NTC, however, this approach has been found to have significant drawbacks pertaining to accuracy and reliability. Port numbers can be split into 3 ranges: well-known or system ports, registered or user ports, and dynamic ports. This is summarized in Figure 2.7. Randomization, or the act of applications using the

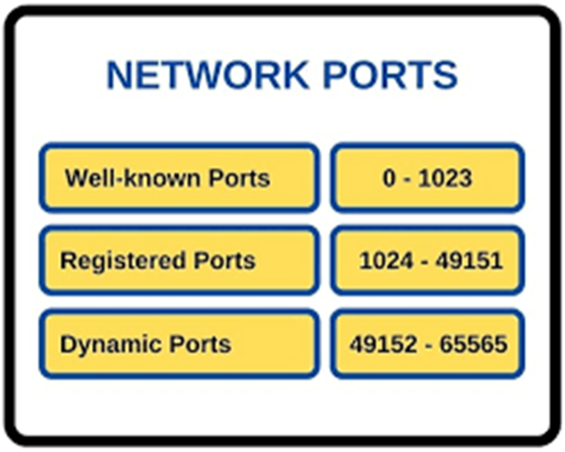


Figure 2.7: Distribution of ports as maintained by the IANA.

dynamic or non-standard port ranges, is one reason there are concerns when using the port based approach. Another is due to masquerading, which is when applications hide their identity by using standard port numbers that are usually reserved for specific applications and protocols. One example of masquerading is a botnet attack using ports reserved for DNS or HTTP. Although these significant drawbacks have been identified, the port based approach is not completely useless. When port conventions are followed, that is to say, when legacy applications are encountered which abide by the rules set in place by the IANA, port based does rather well.

In [31] they found that when deploying the port based approach for NTC, less than 70% of flows were correctly classified. Additionally, in [13] they identified that the proportion of correctly classified flows was dependent on the types of traffic present, particularly in the case of traffic sets that contain higher volumes of P2P flows, overall accuracy was the lowest. On the other hand, traffic sets containing WWW, DND, Mail, News, SNMP, NTP, Chat and SSH flows had the highest precision and recall with over 90%. To evaluate the accuracy of the port-based approach, common metrics such as accuracy, precision, recall and F1-score were used, and the CoralReef software suite.

2.3.2 Payload Based Approach

To avoid the common pitfalls associated with the port based approach, deep packet inspection was introduced, with popular libraries such as OpenDPI and nDPI emerging to offer researchers the ability to classify network data. This approach inspects the actual payload of the data being observed on the network, avoiding issues like masquerading and randomization. This is accomplished through the use of a signature library, where byte signatures within the payload are matched within the library, and classified according to the value that maps to that specific signature. This matching process can either be done on a single packet, or on an aggregation of packets using one or more signatures.

Although this approach does outperform the port based approach, particularly when applications are not abiding by standard port conventions, there are pit falls to this approach. These pitfalls may be that there are potential legal concerns associated with accessing private data, the approach is resource intensive, and lastly, deep packet inspection lacks visibility in determining the causal application when the traffic is encrypted. Even with these shortcomings, DPI can still prove to be very effective; in (Slimming down deep packet inspection systems), they introduced a DPI technique with the goal of reducing the computational complexity of the classification process by only looking at part of the byte signature or setting limits on the number of packets to be used in the matching process. Through their experiments, they were able to achieve up to 99% accuracy for P2P and Mail traffic, overshadowing the results found in the port based approach.

2.3.3 Flow Features Based Approach

In response to the issues related to the port based approach and DPI, using statistical features from a flow to identify network traffic has become widely used. The idea of using flow features to classify network traffic has been gaining significant traction over the years, providing a reliable method to identify flows without jeopardizing accuracy like with the port based approach, or risking legal consequences, computational complexity

and low visibility in scenarios involving encrypted traffic like in DPI. Features of a flow might include duration of the flow, length of data within the packets, number of bytes, and arrival periodicity. These features are thought to be indicative of the application generating the traffic, which ML algorithms can then leverage to generate class labels. These ML algorithms may be supervised, such as Decision Trees, Neural Networks and Support Vector Machines (SVM), unsupervised such as in clustering algorithms, or they can combine supervised techniques with unsupervised techniques in a semi-supervised manner.

In [13], through classification experiments involving network traffic collected in the U.S., Korea, and Japan, it was found that when using SVM to classify network flows, the highest performance was achieved (over 98% accuracy) when the model was trained on more than a few thousand flows. They also found that as the size of the training set decreased, so did the performance of the model, indicating a linear relationship between training set size and performance. Alternatively, NN's have an inverse relationship between training set size and performance, namely the computational speed required to train the model. That is to say, as the training set surpassed 10,000 flows to gain better performance metrics, the NN was excessively slow to train and not practical, whereas the SVM required less than 10,000 flows to achieve the highest performance when compared to other algorithms such as Naïve Bayes, Bayesian Networks, Decision Trees, k-NN, and Neural Networks.

2.4 Strategies to Labeling Network Data

Accurately labeling large datasets of network traffic with the least amount of cost to available resources is one of the most important requirements to building high-performance supervised machine learning models tasked with classifying network traffic. However, with the abundance of data required to train the model, acquiring labeled data proves to have its own challenges. In [16], researchers conducted a study on the challenges presented in the labeling process, identifying two main approaches that exist today: automatic and human guided. Their recent survey on current labeling methods highlights the advantages

and disadvantages of each approach, delving deeper into the specific implementations of both automatic and human-guided techniques, such as injection timing versus behavior profiles for automatic labeling and manual versus assisted for human-guided. According to their findings, using an automated labeling method involves generating a dataset within a structured and predictable network environment. This helps identify abnormal activities amidst regular traffic, thereby removing the need for expert manual labeling.

In 2021, Heng, Chandrasekhar, and Andrews, with the help of six undergraduate students from UT Austin, created an automated platform designed to produce and gather data traffic from numerous popular mobile apps within a controlled setting [24]. The workflow of their system is illustrated in Figures 2.8 and 2.9. Through their system, users designate common web browsing activities for each application, such as scrolling a news feed on Facebook or watching a video on Netflix, enabling the data to be tagged with both the application name and the corresponding activity. Their efforts produced three labeled datasets: one based on deterministic methods, another utilizing randomized techniques, and the third they considered the most authentic as it was generated by humans. Out of the three datasets they generated, the human-generated dataset most closely aligns with the findings presented in this paper. However, based on the findings of our own research, it is reasonable to assume that the human- generated dataset will inadvertently contain unintended traffic that will be mislabeled. This is referred to as label bias, and it occurs when a subset of labels derived are biased or assigned in an unreliable manner, creating noise in the data. This labeling error will ultimately degrade the performance of a classifier if used as a training set and will likely require techniques to handle the noisy labels.

On the other hand, when using human-guided labeling methods to circumvent the disadvantages associated with automated techniques, the network environment lacks control, placing the responsibility entirely on expert users. Guerra et al. go on to discuss a significant challenge that presents itself when implementing the human guided labeling technique, which is the difficulty in labeling the volume of traffic needed for current statistical based Network Intrusion Detection system (SNID) requirements. An example of research that

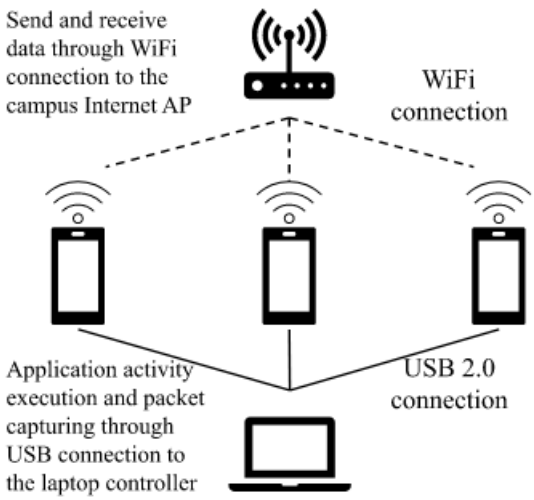


Figure 2.8: Experimental setup of the work presented in [24]

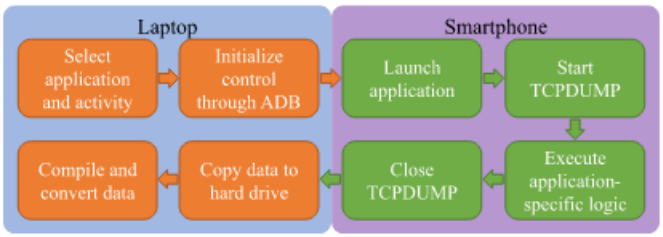


Figure 2.9: Workflow of the data collection in [24]

utilizes the human guided approach is in [17]. Here, they devised a technique to improve the labeling process known as WebClass, a web-based software system that enables users to examine and label potential anomalies on time series of traffic measurements through a graphical user interface (see Figure 2.10). Their research harnessed a community of experts for annotating and continually monitoring labels, ensuring they remain current and highly precise. WebClass provided a platform for researchers to collaborate on the network traffic labeling process, aiming to enhance both its simplicity and reliability. By enabling users to review others' labels, the community could collectively identify and agree upon genuine positive anomalies.

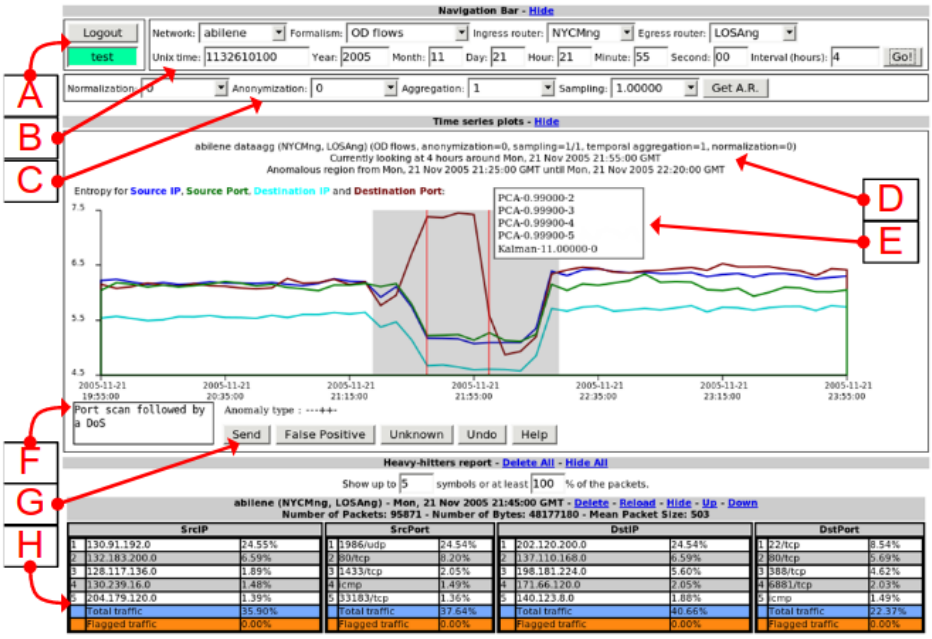


Figure 2.10: WebClass GUI

An alternative approach to human-guided and automated techniques, semi-supervised methods can also be used to generate class labels for network traffic, which is what the work done in [22] was meant to accomplish. In their work, they created a system which combines supervised learning with unsupervised learning to mitigate the challenges associated with labeling large volumes of network data. Their approach involved using multiple representations of the data to gain insight on the different patterns in the data that may present itself through clustering algorithms. Then, once each representation of the data has been run through a clustering algorithm, (K-means), a consensus function is used to essentially combine the output of the initial clustering algorithms. They refer to this stage as the ensemble clustering phase. After the ensemble clustering phase, labels are predicted, and the labeling process is refined through local self-training. This is accomplished by analyzing the decision boundaries of the labeled and unlabeled data from each individual cluster. K-Scores are used to predict the probability that an unlabeled flow belongs to each of the k classes, with the highest probability yielding a class label that maps to that specific

class. Simultaneously, their system utilizes a global training technique which also uses the decision boundary to make the labeling process more robust, using the entire dataset in a semi-supervised manner to produce pseudo labels, as opposed to only using the individual clusters. When evaluated against 4 other semi-supervised labeling techniques, it was found that their SemTra system achieved an average accuracy of 94.96% in a binary classification problem, the third highest of all the other techniques. On the other hand, in a multi-class classification, their system was able to reach 93.84%, which is lower than with the binary classification problem, but still the highest of all the other methods.

Chapter 3

Experimental Plan

The objective of this chapter is to introduce the experimental platform that was used to conduct the research presented in this document, which we call the Human App Labeling System. This system presents as an innovative approach to labeling network traffic by integrating user interaction through a browser extension and providing users with a platform to actively label their Internet activities from a set of options like Email, News, Social media, Video streaming and Information browsing, thus contributing to a unique form of ground truth label acquisition. This approach not only refines accuracy and comprehensiveness of the labeling process but also minimizes reliance on resource-intensive manual efforts and mitigates loss in terms of mislabeling traffic. In addition to our novel labeling approach, we propose a formal comparison of these user-generated labels to those generated by deep packet inspection tools, particularly ntop's open-source product, nDPI. The framework presented in this study reveals a novel perspective on network traffic labeling and holds promise for advancing the field of network traffic analysis that utilizes supervised learning techniques.

3.1 Human App Labeling System

The Human App Labeling System has 4 core elements: Firefox's WebExtensions API, pmacct enabled with the features offered by nDPI, and a post-processing script that utilizes the popular WHOIS protocol and our own flow matching logic; all 4 elements will be further discussed in the subsequent sections. In general, the WebExtensions API is used in this context to develop a Firefox extension that enables users to label their Internet browsing

activity through the use of a popup window that prompts the user to give feedback on the type of applications they are interacting with during a particular browsing session (see Figure 3.1).

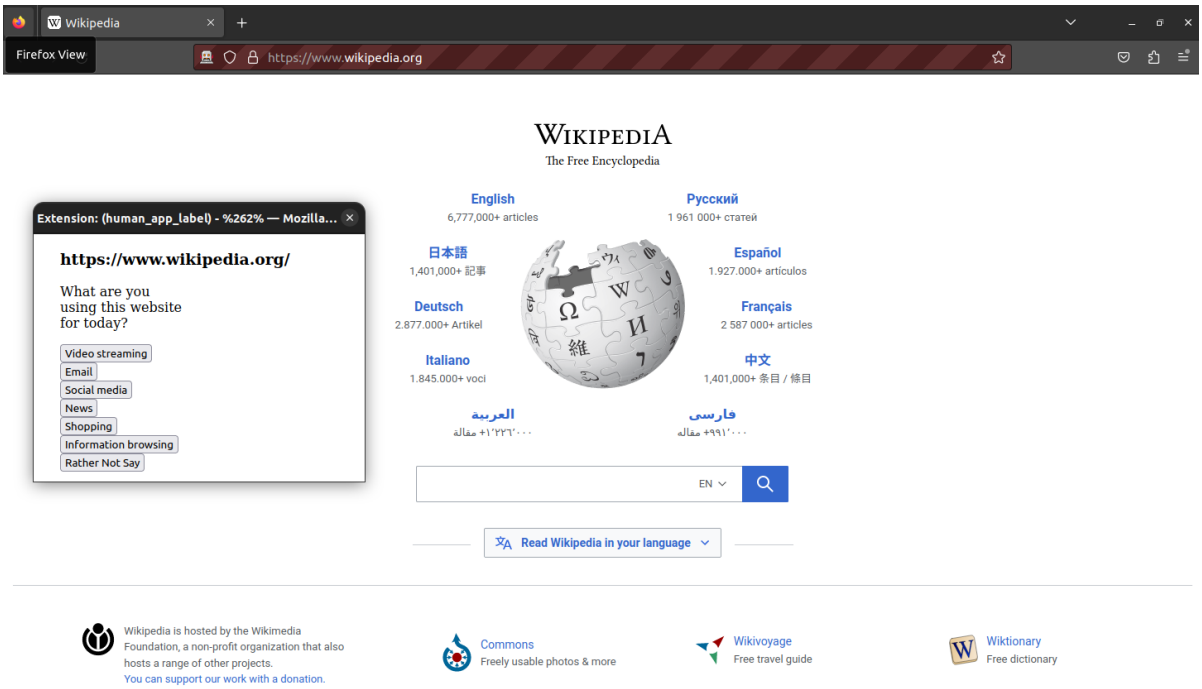


Figure 3.1: An example that demonstrates the Human App Labeling extension running in Mozilla Firefox

Simultaneously, pmacct is running in the background to provide more insight on the traffic users are generating during their session and produce class labels for individual flows using nDPI, an open-source software kit used for DPI. Once the user ends their browsing session, post-processing automatically begins on both the output data provided by pmacct and the output data generated by the browser to ensure that the final output dataset provided by the system contains accurate labels. This post-processing is done through the use of a Python script which merges the pmacct output with the WebExtensions output based on specific parameters and makes use of the WHOIS protocol to discern intended traffic that should be labeled from unintended traffic that should not be labeled, all based

on WHOIS metadata. Refer to Figure 3.2 for a general overview of the system design.

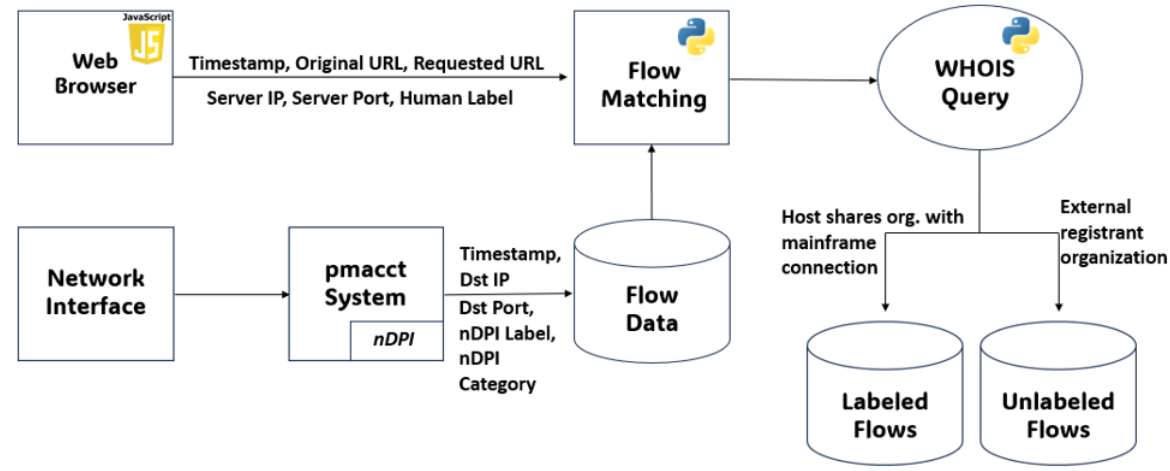


Figure 3.2: Overall design of the Human App Labeling System

3.1.1 WebExtensions API

Firefox's WebExtensions API allows developers to create web extensions that complement the browsing experience and are compatible with multiple browsers. For this stage of the development of the Human App Labeling system, the extension was designed to run on Mozilla Firefox Developer edition on Ubuntu 22.04 LTS. Furthermore, the WebExtensions API contains JavaScript API's which are used in the browser's background scripts where developers of the background logic decide how the system handles certain events such as when a host initiates an HTTP request, or when a request is completed. In the case of the Human App Labeling System, the popup which prompts the user for input is only instantiated when the browser detects a *GET* request for the mainframe of a particular web page. The popup window will remain visible until the user chooses one of the following options discussed in beginning of this chapter, in addition to an option "Rather not say." In the case the user chooses Rather not say, their traffic will not be recorded until prompted by another *GET* request. Once the system receives the users label associated with this *GET*

request, both the label and the host that made the request are logged to be later applied to future requests for the same mainframe. This helps to avoid multiple popup windows overwhelming the users browsing experience when the same host is making several HTTP requests.

If a request made by a host is not a *GET* request for a web page's mainframe, such as a *GET* script request or a *POST* script request, the background script contains logic to handle these requests in a unique manner. If the host that is making the request contains the original mainframe host name string anywhere in its own host name, the request is labeled with the same human label that maps to the mainframe request. An example is if a user navigates to cnn.com, chooses News from the set of options, then a *GET* request from host cnn.io is triggered, this request will also contain the News label because they share the string 'cnn'. Otherwise, the request is not labeled. This technique which helps to separate the requests that should be labeled from those which should not be assists in distinguishing the auxiliary traffic that is meant to supplement the intended host from the unintended traffic that is present for other reasons, such as advertisement delivery.

While the extension is running, the background script is communicating with another programming interface within the system which is used to capture and log the information for all requests extracted by the extension. This data is available in an output file urlstream.csv, providing vital information about each request such as the human label that was given to that request, host name, epoch time, origin URL, server IP, and server port. Throughout this document, the output urlstream.csv file refers to the output file which contains relevant information about the streams of URLs that are triggered upon navigating to a web page. The output of this module is represented as the 'Web Browser' portion of Figure 3.2. This output data is later used as matching criteria in the post-processing stage, further outlined in Section 3.0.3.

3.1.2 Pmacct and nDPI

Pmacct, short for Promiscuous Mode Accounting, is an open-source project developed to help monitor and analyze IPV4 and IPV6 network traffic by capturing packets and aggregating them into flows. These flows can be exported to a comma-separated values (CSV) file to further analyze each individual flow. For the purposes of this study, the fields which we were most concerned with were destination and source port, destination and source IP, number of packets/ bytes, and flow start/ end times. Pmacct can also be configured to enable features offered by nDPI, a toolkit for traffic monitoring offered by the company ntop. As observed in Figure 3.2, when pmacct is configured to use these nDPI features, pmacct's flow capture output files are appended with a class label for each individual flow based on flow signatures detected by nDPI. nDPI is capable of classifying up to 300 protocols including but not limited to popular protocols such as DNS, HTTP, SMTP, Google, and several others, each of which falls under a category like Web, Social media, Email to name a few. These class label categories will be the main focus of our comparison to our human labels in the hopes of achieving a robust analysis of our Human App Labeling system versus nDPI.

3.1.3 Post-Processing Stage

After the user has ended their browsing session and the extension has been successfully terminated, the system will enter a state of post-processing, represented in Figure 3.2 as the 'Flow Matching' and 'WHOIS Query' modules. First, the system goes into a wait stage for 5 minutes to allow pmacct to finish writing the flow information it was able to obtain to the output flow CSV file. Once the wait period has expired, the post-processing comes in the form of a Python script, which expects 2 files as input: the pmacct output file provided by pmacct and the URL stream file containing relevant data for each request provided by the extension. The post-processing stage's first objective is to find matches from the URL stream file in the pmacct flow file based on IP and port numbers, and time

difference between the epoch time denoted in the URL stream file and the flow start time from the pmacct flows file. If a request from the URL stream file has an IP that matches a destination or source IP in the pmacct flows file, in addition to the destination or source ports being the same, and the time difference between the entries is within 30 seconds, the system considers this to be a match. When a match is found, the corresponding data for that request is appended with the data for that matching entry from the pmacct flows file.

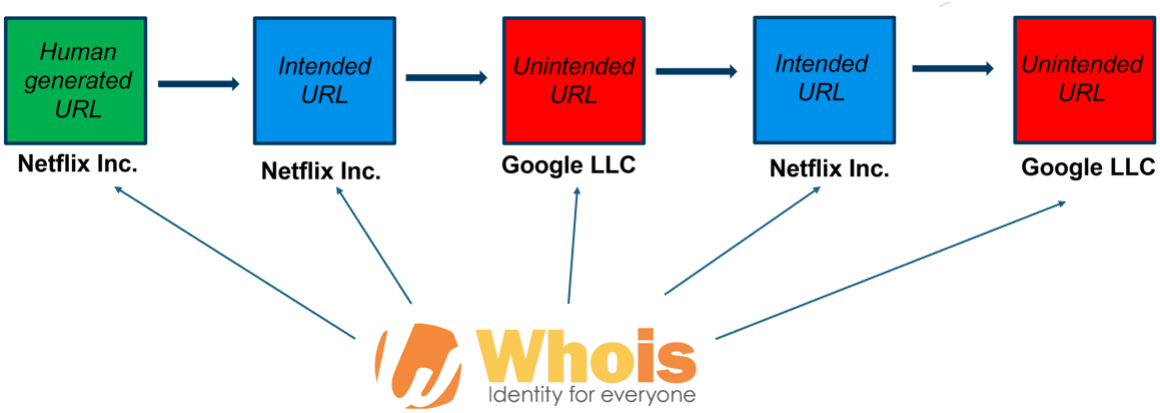


Figure 3.3: Illustrating the usage of the WHOIS protocol in the Human App Labeling System

The next step in the post-processing stage is to conduct WHOIS queries for all unlabeled flows to determine whether or not the request is in fact unintended traffic or if it was inadvertently not labeled by the system. This occurs when a host is using an alias which does not immediately appear as intended traffic, so it is handled by the system as unintended traffic and thus does not receive a human generated label. For example, if a user is video streaming on 'netflix.com', and a request has been made by the host 'nflxext.com', one might be able to assume that 'nflxext.com' is associated with 'netflix.com' based on the host name, but the system will not make the same assumption. Because 'nflxext.com' does not contain the string 'netflix' anywhere in the host name, the request will not receive the users human label which was chosen after navigating to 'netflix.com'. To mitigate this issue, WHOIS queries can be made for these unlabeled flows to obtain metadata on the

host that can shed more insight into who that particular organization is affiliated with.

For the purposes of our study, we chose to use Registrant Organization as a deciding factor in whether or not two hosts are affiliated with one another. See Figure 3.4 for an example of a WHOIS record for 'nflxext.com.' Additionally, refer to Figure 3.3 for an illustrative figure depicting how the WHOIS protocol is implemented in our Human App Labeling System. In Figure 3.3, the WHOIS protocol is used to extract the Registrant Organization of the human generated URL, which is found to be Netflix Inc. Then, a stream of URLs is produced, and the WHOIS protocol can again be used in a similar fashion. Because 2 of the 4 URLs are associated with Netflix Inc., these requests are deemed to be intended; those that belong to Google LLC are ultimately found to be unintended traffic and remain unlabeled.

Raw Whois Data

```
Domain Name: nflxext.com
Registry Domain ID: 1639716970_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2024-02-07T02:55:22+0000
Creation Date: 2011-02-11T18:09:52+0000
Registrar Registration Expiration Date: 2025-02-11T00:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registrant Organization: Netflix, Inc.
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at https://domains.markmonitor.com/whois/nflxext.com
Admin Organization: Netflix, Inc.
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at https://domains.markmonitor.com/whois/nflxext.com
Tech Organization: Netflix, Inc.
```

Figure 3.4: Sample WHOIS record for nflxext.com

After each request from the URL stream file has been appended with flow information

provided by pmacct based on key parameters outlined earlier, and each flow that was initially not labeled has been thoroughly analyzed and updated with correct human labels when applicable, the system will produce one, single output file in CSV file format. The file will contain pmacct flow information for each request, including the nDPI label and category, as well as the human label that is provided by the extension. See Figure 3.5 for a detailed example of what one should expect to see in the output of the system.

This output file provides an excellent platform to evaluate the accuracy and effectiveness of nDPI's class labels when compared to the human labels provided by the user because it offers an opportunity to visually observe how often the nDPI category matches with the ground truth human label.

nDPI Label	SRC_IP	DST_IP	SRC_PORT	DST_PORT	PROTOCOL	TIMESTAMP	TIMESTAMP	TIMESTAMP	PACKETS	BYTES	Human Label	Host	Origin URL	Category	Organization
TLS	3.137.95.4	10.0.2.15	443	58666	tcp	27:01.4	37:15.9	40:01.6	916	1111899	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.137.95.4	58666	443	tcp	27:01.4	37:15.9	40:01.6	633	620603	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.141.216.	10.0.2.15	443	53616	tcp	27:06.6	28:27.1	34:08.7	20	6953	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
TLS	10.0.2.15	3.141.216.	53616	443	tcp	27:06.6	28:27.1	34:08.7	21	4012	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
TLS	3.141.216.	10.0.2.15	443	53618	tcp	27:06.6	28:27.1	34:08.7	20	7179	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
TLS	3.131.250.	10.0.2.15	443	34384	tcp	27:25.9	27:26.7	30:01.9	13	3956	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.131.250.	34384	443	tcp	27:25.9	27:26.7	30:01.9	13	1332	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.141.216.	53618	443	tcp	27:06.6	28:27.1	34:08.7	21	2590	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
TLS	3.131.250.	10.0.2.15	443	34372	tcp	27:25.9	30:17.8	33:01.1	25	6768	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.131.250.	34372	443	tcp	27:25.9	30:17.8	33:01.1	24	4153	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.19.205.1	10.0.2.15	443	49596	tcp	27:35.5	39:05.8	42:01.1	99	36952	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.19.205.1	49596	443	tcp	27:35.5	39:05.8	42:01.1	99	14245	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.19.205.1	10.0.2.15	443	49586	tcp	27:35.5	39:07.0	42:01.1	557	105654	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.19.205.1	49586	443	tcp	27:35.5	39:07.0	42:01.1	199	677531	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.137.95.4	10.0.2.15	443	33980	tcp	34:29.0	37:26.8	40:01.6	40	43113	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.137.95.4	33980	443	tcp	34:29.0	37:26.8	40:01.6	38	7920	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.132.231.	10.0.2.15	443	38290	tcp	35:04.1	35:05.2	38:01.8	1	44	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.132.231.	38290	443	tcp	35:04.1	35:05.2	38:01.8	2	100	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.132.231.	10.0.2.15	443	38276	tcp	35:04.1	38:06.3	41:01.7	24	4416	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	10.0.2.15	3.132.231.	38276	443	tcp	35:04.1	38:06.3	41:01.7	22	4440	Video streaming	netflix.com	https://www.net	Web	Netflix, Inc.
TLS	3.131.26.1	10.0.2.15	443	60064	tcp	35:15.9	36:18.0	42:01.1	17	5528	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
TLS	10.0.2.15	3.131.26.1	60064	443	tcp	35:15.9	36:18.0	42:01.1	18	2640	Video streaming	netflix.com	https://ae.nflxin	Web	Netflix, Inc.
NetFlix	10.0.2.15	45.57.40.1	37240	443	tcp	35:16.7	35:24.8	41:01.7	16	2241	Video streaming	netflix.com	https://www.net	Video streaming	Netflix, Inc.

Figure 3.5: Example output file generated by Human App Labeling System

3.2 Experimental Methodology

3.2.1 Objectives and Hypotheses

The primary objective of the experiments conducted during this research process was to compare the labels generated by our human generated labels to those generated by nDPI. To do this, we aimed to devise a system which produces labeled network flow data that

contains both nDPI class labels/ categories and a human generated ground truth label for each HTTP request made during an Internet browsing session. This system should be operated by researchers and those outside of academia alike, offering a seamless user experience that demands little effort from the user. These human generated ground truth labels can then be used to train a machine learning classifier, providing an opportunity to compare its performance to a separate model trained using nDPI labels instead. It was hypothesized that the human generated labels would yield the best performing classifier because the human generated labels are being considered the ground truth in this context, and they are believed to be representative of true application behavior. Furthermore, due to the lack of visibility nDPI has in the case of encrypted traffic, there is potential for the human generated labels to provide better transparency compared to the deep packet inspection counterpart. Finally, it was hypothesized that nDPI could not be considered the absolute ground truth because there is no guarantee that the output of nDPI is 100% accurate.

3.2.2 Experimental Setup and Data Collection

The experimental environment for this research involved browsing the web on the University of Texas at El Paso's private network, particularly on Ubuntu machines in UTEP's Communication Networks Laboratory (NetLab). In addition to this location, the data generated from browsing the web was produced on a laptop on a home network in El Paso, Texas, also on an Ubuntu operating system. At both locations, the browser that was being used to collect data is Firefox Developer edition, which offers more tools for developing software when compared to the standard version of Firefox.

Experiments consisted of browsing the web and observing the nDPI labels that pmacct would generate for each flow to get an idea of which labels might be prevalent in a typical browsing experience, as well as determining the quantity of traffic generated by different applications. In any case, experiments were tailored to expect some quantity of traffic to be encrypted and classified as encrypted by nDPI because it was eventually observed that

encrypted traffic is almost always present during a browsing session long enough to generate significant amounts of traffic. The inclusion of encrypted traffic has proven advantageous for our research, as the Human App Labeling system demonstrates the ability to optimize the labeling process compared to nDPI by identifying network flows that would otherwise be classified as Transport Layer Security (TLS) flows or Cloudflare. Note: Although Cloudflare is not solely used for encrypting traffic, one feature this web service implements is Secure Sockets Layer (SSL), which is the first implementation of TLS. Therefore, throughout this document, encrypted traffic will be referred to those flows which are labeled by nDPI as TLS and Cloudflare and categorized as Web, per nDPI's class to category mappings.

3.2.3 Experimental Plan

The Human App Labeling System is capable of producing six ground truth labels: Video streaming, Social media, News, Information browsing, Shopping and Email. These options were chosen because they were believed to be options that encompass all activities that a typical web user might participate in. However, out of those human labels, nDPI would only be capable of producing Video streaming, Email and Social media labels. Although there are only three nDPI categories that could potentially overlap in the system output (human label and nDPI label for a given flow match), we aimed to ensure that the generated network traffic data presents as many nDPI categories and protocols as possible so a proper evaluation of nDPI's full capabilities can be achieved. In a similar fashion, to demonstrate the capabilities of the Human App Labeling System, it was necessary to ensure traffic from all six possible options was also generated.

In addition to generating traffic that demonstrates the capabilities of both nDPI and the Human App Labeling System, it was important that the browsing activities performed during the experiments were unbiased and unconstrained. This means that the typical browsing activities that we anticipated from users must be present in the datasets, and the activities should not be contrived, as was seen in [24]. The application activities conducted in their work involved explicitly prescribing actions such as scrolling, clicking, or remaining

on one web page for x amount of unit time. In other contrast to their work, a primary objective of the experiments presented in this work is to ensure that the datasets are produced as naturally as possible, and represents the natural behaviors of the Internet.

Once a significant amount of flows which have been labeled with both the nDPI labels and human labels has been generated, a comparative analysis of the labels will follow. As mentioned in Chapters 1 and 2, DPI is a powerful tool that can be used to identify traffic or generate labels for training a supervised machine learning classifier. However, to get the best results from the trained classifier, it is imperative to using accurate, representative labels that present little to no noise in the training data. Having acknowledged this, evaluating the performance of nDPI will shed insight on the accuracy and performance of DPI as it compares to the ground truth, a much needed area of research considering the many applications of NTC today.

Chapter 4

Experimental Results

The Human App Labeling System discussed in Chapter 3 was used to label network traffic during normal network usage. Browsing activities included popular websites such as Netflix, Twitter and CNN. For a complete list of the browsing activities that were performed during our dataset generation, refer to Table 4.1. As a result, the system yielded network traffic datasets with two labels: the human ground truth label and the label provided by the nDPI library. Individual datasets were produced for each separate browsing session, then these datasets were combined to generate a dataset with approximately 5.5k labeled network flows. Refer to Figure 4.1 for a heat map visualization of the distribution of classes and nDPI categories generated by the system. This heat map outlines the number of occurrences of each human generated label overlapping with an nDPI class and category. For instance, the heat map shows that 194 flows that were labeled as Video streaming by the human user were ultimately labeled by nDPI as Amazon, and categorized as Web.

4.1 nDPI Labels versus Human Labels

Evaluation of the human-generated labels compared with those produced by nDPI begins with a straightforward comparative analysis of the combined system output. Three distinct possibilities warrant consideration and will be discussed in this chapter: first, the frequency and infrequency of the nDPI classification matching that of the Human App Labeling System, and the specific causes of this scenario. Next, times where the nDPI label is too vague, or not specific enough. Lastly, any instances where nDPI falls short in categorizing encrypted traffic, whereas the human label identifies the true nature of the

Table 4.1: Experimental browsing activities. The table describes the sites that were visited to generate the dataset, split into categories which correspond to the human label options.

Video Streaming	**Shopping**	**Email**	**Social Media**	**Information Browsing**	**News**
• Netflix • YouTube • Hulu • IFLIX • Twitch	• eBay • Best Buy • WalMart • Target • Wayfair • Wish • Costco • Etsy • Amazon	• Outlook • GMail • Yahoo Mail	• LinkedIn • Instagram • Twitter • Facebook • Reddit	• Google • Wikipedia • Wikimedia • Wikidata	• CNN • Washington Post

browsing activity. This final scenario may be the most significant of them all because it is a reminder of a critical drawback to the nDPI labeling process: nDPI cannot extract enough information from encrypted traffic to attach a meaningful label to those flows.

4.1.1 When Do Labels Match?

Out of the 5.5k flows in the final training dataset, 74.55% of the flows were given labels by nDPI which differed from the human label for that respective flow. This is a significant portion of the dataset, indicating that much of the application activity that nDPI classifies is potentially being mislabeled. This can be seen in Figure 4.1 where News flows are being categorized by nDPI as Web, in Social media flows that are being categorized as Web and Cloud, and in Video streaming flows that are being categorized as Web by nDPI. According to ntop's documentation, nDPI should be capable of identifying flows from CNN, assigning a category of Media. As illustrated in Table 4.1, CNN was indeed one of the web sites visited, and ideally nDPI would have classified these flows as Media, but this is not what happened. Instead, flows that the human user labeled as News by the human user after

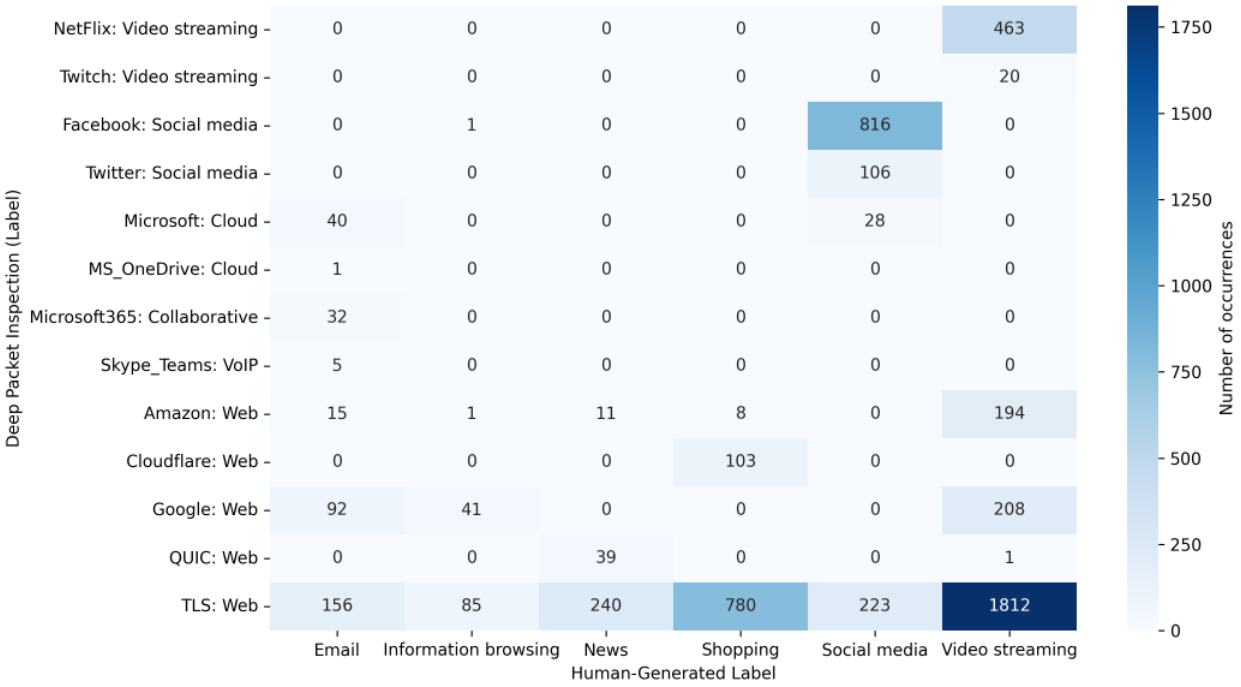


Figure 4.1: Distribution of class labels present in final training dataset. The heat map demonstrates the number of flows for each human label category (x-axis) along with its respective nDPI label (y-axis). Here, the y-axis labels are organized as nDPI Class: Category.

visiting cnn.com and washingtonpost.com were categorized by nDPI as Web, with the majority of the flows presenting as encrypted to nDPI. Because we are considering the human generated label as the ground truth, these mismatches between the human label and the nDPI label suggests that nDPI fails to identify true application behavior in some cases, and may lead to issues when using these nDPI labels to train a machine learning classifier to learn patterns in network traffic.

The remaining 25.45% of the flows in the dataset which had matching nDPI and human generated labels were ultimately found to be generated from Video streaming and Social media, particularly Twitter, Netflix, Twitch and Facebook. This means that for some of the traffic generated from these web sites, nDPI successfully labeled these flows according to

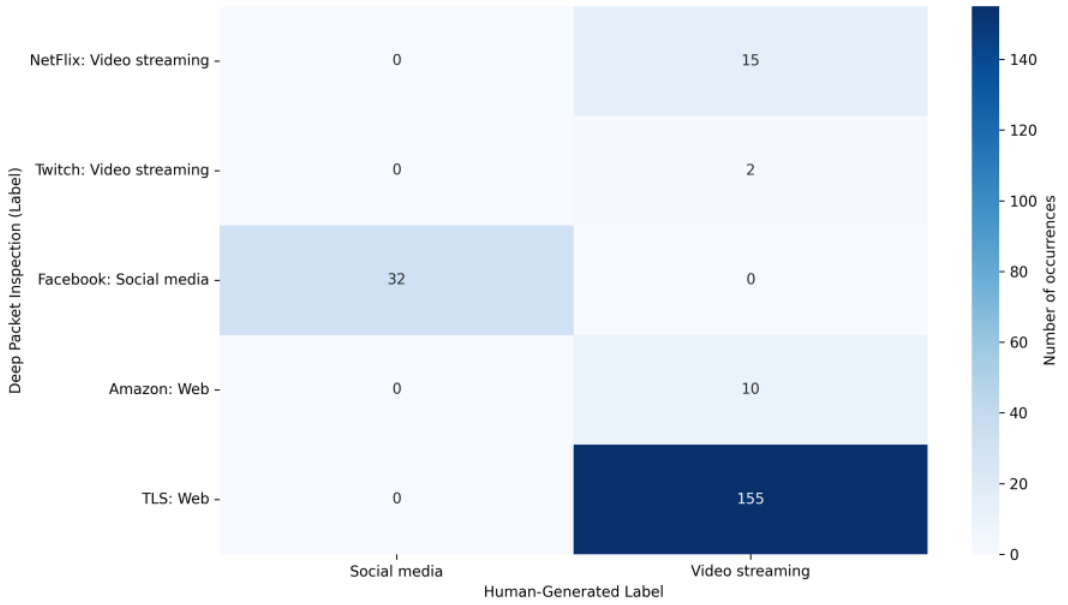


Figure 4.2: Heat map which demonstrates human label and nDPI label distribution when visiting Netflix, Twitch and Facebook. The human user labeled their web browsing activities as 'Social media' and 'Video streaming' (found on the x-axis), however, nDPI provided class labels that map to Video streaming, Social media, and Web (found on y-axis).

their respective ground truth counterpart. Refer to Figure 4.1 for the exact amount of these specific flows, and which nDPI classes coincide with each ground truth label. Additionally, to demonstrate nDPI's ability to identify Facebook and Netflix flows, refer to Figure 4.2 for a heat map which illustrates the flows generated from watching a movie on Netflix, streaming some videos on Twitch, then visiting Facebook. It can be seen in this heat map that all of the Social media flows which were labeled as Social media by the Human App Labeling System were, in fact, also labeled as Facebook and categorized as Social media by nDPI. Furthermore, although a large portion of the flows labeled as Video streaming by the human user were classified as encrypted by nDPI, 15 of the Video streaming flows were labeled as Netflix, and 2 were labeled as Twitch.

4.1.2 Vague nDPI Labels

Vague nDPI labels can lead to noise in the dataset and potentially harm the performance of the model if being used to train a machine learning classifier. In this context, the term "vague" refers to the situation in which the nDPI label does not match the human generated label, but with some intuition it can be seen that they are almost correct. One class this scenario applies to is Email flows generated from visiting GMail and Outlook; according to their documentation, nDPI should classify flows generated from GMail and Outlook (both categorized as Email), however, we can see in Figure 4.1 that out of the flows labeled as Email by the human user, none of them were categorized as Email by nDPI. Instead, the majority of them were categorized as encrypted traffic, while the rest received nDPI labels for Cloud, Collaborative, VoIP and Web. The Email flows that received an nDPI label of Google were those from GMail, and the Email flows that were labeled Cloud and Collaborative were those from visiting Outlook. These discrepancies are a clear display of nDPI's incapability of producing accurate labels, because they are simply not specific enough.

Another class where the nDPI labels could be considered too vague is in Social media traffic being categorized as Social media by nDPI. A portion of these flows are being classified as Facebook, when it can be seen in Table 4.1 that one of the web sites visited to generate Social media traffic was Instagram. nDPI is supposed to be capable of identifying and labeling Instagram flows, however, the Instagram traffic being generated and labeled with the ground truth by the user is being classified as Facebook by nDPI. This was further confirmed by visually analyzing the data and referring to the origin URL that corresponds to these Social media flows. This inaccuracy may be because Instagram is owned by Meta Platforms, Inc., the parent company of Facebook. If this is the case, that may suggest that nDPI is not specific enough and may lead to issues when training a machine learning classifier on nDPI labels.

4.1.3 Encrypted Traffic

Based on observations of the dataset, the inconsistencies between nDPI labels and human generated labels can be largely attributed to nDPI labeling a significant number of the flows as encrypted traffic, approximately 3.4k or 61.56% of the total dataset. Encrypted traffic is categorized simply as 'Web' by nDPI, and since the comparison being made for this study is between the human ground truth label and the nDPI category, this will yield many mismatches considering 'Web' is not an application activity option being offered to the user.

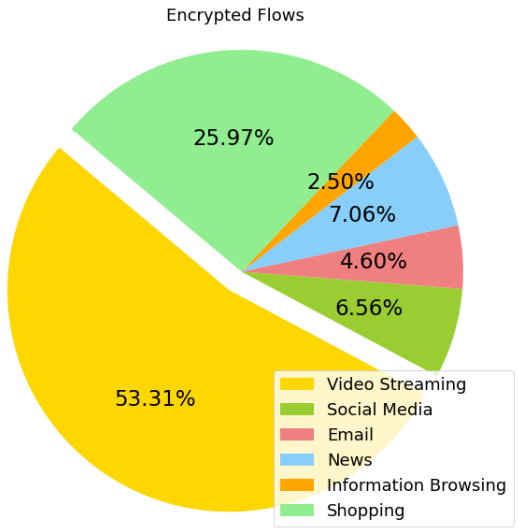
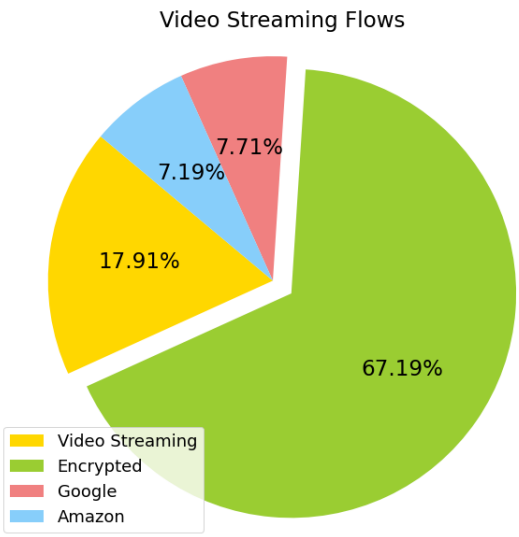


Figure 4.3: Pie chart which illustrates the percentage of flows in the final dataset that were given meaningful ground truth labels by Human App Labeling System when classified as encrypted by nDPI.

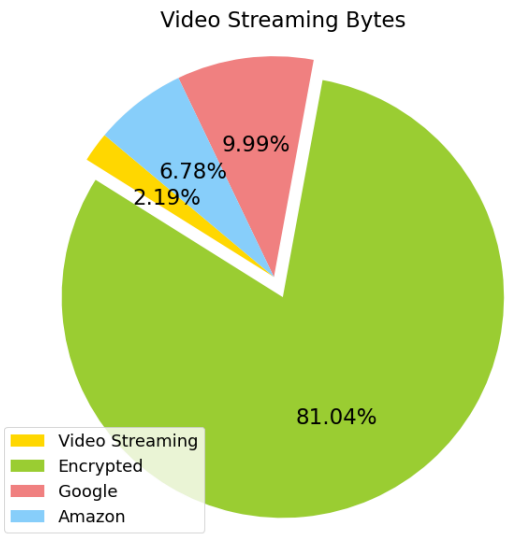
This lack of visibility nDPI suffers from offers an opportunity for the Human App Labeling System to attach labels to flows which cannot be fully categorized due to encryption. This is visualized in Figure 4.3. The encrypted flows represented by this pie chart were

ultimately given meaningful labels due to the robust labeling techniques implemented in the Human App Labeling System, and leads to a higher volume of data labeled with the ground truth label despite nDPI categorizing a large portion of the dataset as encrypted. In addition to quality of training data, the quantity of data provided to the model for training can also be a significant factor in yielding high performance, especially in the field of NTC where acquiring large network traffic data sets for machine learning tasks can be challenging, as discussed in Chapter 2.

Furthermore, it was observed that out of the flows which were labeled Video streaming by the user, more than half of the flows were labeled as encrypted by nDPI, while only 17.91% of the Video streaming flows received a Video streaming label by nDPI. Additionally, out of all the bytes transmitted for Video streaming flows, only a small portion of them were attributed to Video streaming by nDPI, approximately 2.19% of all Video streaming bytes, and 81.04% of these Video streaming bytes were labeled as encrypted by nDPI. This is a significant portion of the Video streaming data, demonstrating once again where nDPI fails to classify traffic effectively. The full distribution of both flows and bytes for Video streaming can be seen below in Figure 4.4.



(a) Distribution of flows for all Video streaming traffic



(b) Distribution of bytes for all Video streaming traffic

Figure 4.4: Pie charts which reveal the distribution of flows (Figure 4.4a) and bytes (Figure 4.4b) for Video streaming activities which were labeled as Video streaming by the human user. The flows and bytes can be seen to predominantly receive encrypted nDPI labels, along with Google, Amazon, and Video streaming.

Chapter 5

Conclusions and Future Works

With the rise in NTC application scenarios today, an accurate and efficient method to classifying network traffic has been in high demand by ISPs and network management organizations. Machine learning has recently become the most popular method to classifying network traffic, however, in the case of supervised models, labeled training data is required. It was discussed in Chapters 1 and 2 that manually labeling network flows is overly laborious due to the fact that expert knowledge is needed, in addition to the time consumed when dealing with large datasets. Automated approaches such as using standard port conventions or DPI can be used in lieu of manually labeling, however, these techniques have been shown to have their own downfalls that were also discussed in this document.

5.1 Final Remarks

The shortcomings of DPI and port-based approaches discussed in Chapter 2 indicate that the future of NTC likely overlaps with that of machine learning classifiers. DPI libraries such as nDPI may be used to generate the labels for training, but it is well known in the machine learning community that the labels must be accurate and contain as little noise as possible to generate satisfactory performance metrics. To mitigate this issue, we introduced a unique method to acquiring ground truth labels for network data, the Human App Labeling System, which leverages the benefits of network monitoring tools like pmacct and nDPI, as well as the implementation of APIs like the Python WHOIS API and Firefox's WebExtensions API. Through this system, we were able to generate 5.5k flows, each with their own nDPI label in addition to a human generated ground truth label. We used

these labels to evaluate the accuracy and performance of DPI by performing a comparative analysis of nDPI labels versus the ground truth label, which led to acquiring insightful results.

We discussed in Chapter 4 that nDPI identifies Social media flows rather well, with 78.60% of all flows which were labeled as Social media being categorized as Social media by nDPI. On the other hand, nDPI completely fails to identify other application activities such as Shopping and Email, highlighting a major flaw in nDPI's performance. Furthermore, we found that some of the labels generated by nDPI may be considered to be too vague, such as Instagram flows being labeled as Facebook, or GMail flows being labeled as Google, and may present as noise when being used to train machine learning classifiers. Lastly, it was found that the presence of encrypted traffic would be advantageous for the purposes of this research because it offers an opportunity for encrypted flows to receive a meaningful label rather than being labeled as encrypted by nDPI. This provides a chance for our Human App Labeling System to produce labeled data on a larger scale, which would ultimately improve the training process.

5.2 Future Directions

For the future of this research, machine learning classifiers may be trained and deployed to evaluate the performance of nDPI versus the ground truth. This may be accomplished by training two separate classifiers: one model which was trained using only nDPI labels, and another which was trained using the human generated labels, which would be considered the ground truth. Here, the classifiers may be asked to infer on data which contains human generated labels, and compare the performance of each. It is hypothesized that the classifier which was trained on human generated labels will provide the best performance, because these labels are considered to be representative of true application traffic behavior.

When training and evaluating the classifiers, addressing certain challenges is inevitable. These challenges may be choosing the right supervised learning model, determining if the

models require more training data, and mitigating any levels of concept drift in the data. Concept drift is briefly covered in Chapter 2. Given the dynamic behavior of the Internet and applications/ protocols it contains, the patterns which an NTC classifier learns may be irrelevant after a certain period of time. For this reason, it is suggested that training the classifiers on the most recent network traffic data may be required.

References

[1] C. BasuMallick, "Packet-Switched Network vs. Circuit-Switched Network: Understanding the 15 Key Differences." [Online]. Available: https://www.spiceworks.com/tech/networking/articles/packet-switched-vs-circuit-switched-network/

[2] K. Featherly, "ARPANET." [Online]. Available: https://www.britannica.com/topic/ARPANET/A-packet-of-data

[3] U. D. of Commerce, "Quarterly Retail E-Commerce Sales." [Online]. Available: https://www.census.gov/retail/mrts/www/data/pdf/ec_current.pdf

[4] S. Kemp, "Digital 2024: Global Overview Report," 2024. [Online]. Available: https://datareportal.com/reports/digital-2024-global-overview-report

[5] L. S. Vailshery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030," 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[6] A. Zola, "FCAPS (fault, configuration, accounting, performance and security)." [Online]. Available: https://www.techtarget.com/searchnetworking/definition/FCAPS#:~:text=FCAPS%20is%20an%20acronym%20for,OSI%2FISO%20network%20management%20model.

[7] A. Azab, M. Khasawneh, S. Alrabaee, K. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digital Communications and Networks*, September 2022.

[8] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust Network Traffic Classification," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 1257–1270, August 2015.

[9] Z. Liu, R. Wang, and D. Tang, "Extending labeled mobile network traffic data by three levels traffic identification fusion," *Future Generation Computer Systems*, vol. 88, pp. 453–466, June 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17309937

[10] J. Guerra, C. Catania, and E. Veas, "Active learning approach to label network traffic datasets," *Journal of Information Security and Applications*, vol. 49, p. 102388, September 2019.

[11] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, pp. 35–40, January 2012.

[12] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, pp. 56–76, September 2008.

[13] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," January 2008, p. 11.

[14] Y. Lim, H. Kim, J. Jeong, C. Kim, T. Kwon, and Y. Choi, "Internet Traffic Classification Demystified: On the Sources of the Discriminative Power," December 2010.

[15] IANA, "IANA Service Name and Transport Protocol Port Number Registry." [Online]. Available: https://www.iana.org/assignments/servicenames-port-numbers/service-names-port-numbers.xhtml

[16] J. Guerra, C. Catania, and E. Veas, "Datasets are not enough: Challenges in labeling network traffic," *Computers and Security*, vol. 120, p. 102810, June 2022.

[17] H. Ringberg, A. Soule, and J. Rexford, "WebClass: adding rigor to manual labeling of traffic anomalies," *SIGCOMM Computer Communication Review*, vol. 38, p. 35–38, January 2008.

[18] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey," *Computer Communications*, vol. 170, pp. 19–41, March 2021.

[19] T. Bhatia, "OpenDPI." [Online]. Available: https://github.com/thomasbhatia/OpenDPI

[20] ntop, "nDPI." [Online]. Available: https://github.com/ntop/nDPI

[21] P. Lucente, "PMACCT: IP traffic accounting." [Online]. Available: https://github.com/pmacct/pmacct

[22] A. Fahad, A. Almalawi, Z. Tari, K. Alharthi, F. Al-Qahtani, and M. Cheriet, "SemTra: A Semi-Supervised Approach to Traffic Flow Labeling with Minimal Human Effort," *Pattern Recognition*, vol. 91, pp. 1–12, February 2019.

[23] M. Kim and I. Lee, "Human-guided auto-labeling for network traffic data: The gelm approach," *Neural Networks*, vol. 152, pp. 510–526, May 2022.

[24] Y. Heng, V. Chandrasekhar, and J. Andrews, "Utmobilenettraffic2021: A labeled public network traffic dataset," *IEEE Networking Letters*, vol. 3, pp. 156–160, September 2021.

[25] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037–2064, May 2014.

[26] H. Habehh and S. Gohel, "Machine Learning in Healthcare," *Current Genomics*, vol. 22, no. 4, December 2014.

[27] M. Dixon and I. Halperin, "The four horsemen of machine learning in finance," *SSRN Electronic Journal*, January 2019.

[28] H. Ji, X. Xu, G. Su, J. Wang, and Y. Wang, "Utilizing Machine Learning for Precise Audience Targeting in Data Science and Targeted Advertising," *Academic Journal of Science and Technology*, vol. 9, no. 2, p. 215–220, February 2024. [Online]. Available: https://drpress.org/ojs/index.php/ajst/article/view/17922

[29] M. Berry, A. Mohamed, and B. Yap, *Supervised and unsupervised learning for data science*. Springer, 2019.

[30] V. Kodzoman, "Pseudo-labeling a simple semi-supervised learning method." [Online]. Available: https://datawhatnow.com/pseudo-labeling-semi-supervised-learning/

[31] A. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," in *Passive and Active Network Measurement*, 2005, pp. 41–54.

[32] IETF, "Internet Engineering Task Force (IETF)." [Online]. Available: https://www.ietf.org/

[33] J. van Engelen and H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, p. 373–440, February 2020.

Curriculum Vitae

Herman Ramey was born October 9, 1995. He graduated from Riverside High School in the year 2013. From there, he joined the United States Navy as a Hospital Corpsman, providing medical care to U.S. Sailors, U.S. Marines and their dependents. After being honorably discharged, he completed his B.S. in Electrical Engineering from the University of Texas at El Paso in May 2023 with Cum Laude honors. Over the last three semesters, he has been a Research Assistant in the Communication Networks Laboratory from the Department of Electrical and Computer Engineering. His research interests include network traffic classification and machine learning.

[0]Email Address: hframey@miners.utep.edu