On Improved Generalization of 5-State Hidden Markov Model-based Internet Traffic Classifiers

by

Grant H. R. Bartnik

A Thesis
presented to
The University of Guelph

In partial fulfilment of requirements
for the degree of
Master of Science
in
Computer Science

Guelph, Ontario, Canada

© Grant H. R. Bartnik, May, 2013
ABSTRACT

On Improved Generalization of 5-State Hidden Markov Model-based Internet Traffic Classifiers

Grant H. R. Bartnik
University of Guelph, 2013

Advisors:
Dr. Stefan C. Kremer

The multitude of services delivered over the Internet would have been difficult to fathom 40 years ago when much of the initial design was being undertaken. As a consequence, the resulting architecture did not make provisions for differentiating between, and managing the potentially conflicting requirements of different types of services such as real-time voice communication and peer-to-peer file sharing. This shortcoming has resulted in a situation whereby services with conflicting requirements often interfere with each other and ultimately decrease the effectiveness of the Internet as an enabler of new and transformative services. The ability to passively identify different types of Internet traffic then would address this shortcoming and enable effective management of conflicting types of services, in addition to facilitating a better understanding of how the Internet is used in general. Recent attempts at developing such techniques have shown promising results in simulation environments but perform considerably worse when deployed into real-world scenarios. One possible reason for this discrepancy can be attributed to the implicit assumption shared
by recent approaches regarding the degree of similarity between the many networks which comprise the Internet. This thesis quantifies the degradation in performance which can be expected when such an assumption is violated as well as demonstrating alternative classification techniques which are less sensitive to such violations.
Acknowledgments

To all of my family, friends and colleagues for their continued support over the years. A special thanks to my advisor, Stefan Kremer for his considerable patience, encouragement and timely, pragmatic advice.

And finally, to Jennifer for helping me maintain perspective, reminding me of what is truly important and ultimately providing all of the support and accommodation one could ask for.
Table of Contents

List of Tables viii

List of Figures x

1 Introduction 1
   1.1 Contributions .................................................. 4
   1.2 Conclusion ..................................................... 6

2 State of the Art 7
   2.1 Introduction ..................................................... 7
   2.2 Literature Survey ................................................ 10
       2.2.1 Classification Level Considered .......................... 13
       2.2.2 Machine-Learning Algorithms Employed .................... 17
       2.2.3 Data Sources and Ground Truth ............................. 22
       2.2.4 Features ................................................ 27
       2.2.5 Experimental Design ...................................... 32
       2.2.6 Beyond the State of the Art ............................... 38
   2.3 Conclusion ..................................................... 40

3 Preliminaries 42
   3.1 Introduction ..................................................... 42
   3.2 Network Independent vs. Dependent Features and Poor Generalization Performance ......................................................... 42
   3.3 Network Configuration As a Source of Bias .......................... 44
       3.3.1 Relationships between MTU, Packet Size .................... 45
       3.3.2 MTU and Packet Inter-arrival Time ........................ 48
   3.4 Traffic Traces ................................................... 51
       3.4.1 Definition of A Traffic Trace ................................ 52
       3.4.2 Feature Definitions ......................................... 58
       3.4.2.1 Packet Size ....................................... 58
       3.4.2.2 Inter-Arrival Time .................................... 59
       3.4.2.3 Message Size ........................................ 59
       3.4.2.4 Serialization ......................................... 60
   3.5 Hidden Markov Models .......................................... 63
       3.5.1 Introduction ............................................. 64
       3.5.2 Three Canonical HMM Problems ............................. 67
List of Tables

2.1 Summary of surveyed literature and comparison to approach taken in this thesis. .................................................. 39

3.1 Five most prevalent maximum packet sizes observed per trace. ........ 48
3.2 A summary of the maximum number of concurrent sockets utilized by traffic type. ............................................. 51
3.3 A human-readable representation of an Internet traffic trace. ............ 53
3.4 The first trace instance, corresponding to communication between sockets 37504/80. ................................................. 55
3.5 The second trace instance, corresponding to communication between sockets 37505/80. ................................................. 56
3.6 The second trace instance, corresponding to communication between sockets 37506/80. ................................................. 56
3.7 The second trace instance, corresponding to communication between sockets 37507/80. ................................................. 56
3.8 An example of a human-readable traffic trace consisting of multiple concurrent sockets. .......................................... 57
3.9 An example of a human-readable traffic trace employing the message size feature and multiple concurrent sockets. ............... 58

3.10 The Analysis of Variance (ANOVA) table structure and associated calculations. .................................................. 92

4.1 Applications chosen to simulate different classes of traffic and the respective sources. ............................................... 103
4.2 Summary of all 45 experimental conditions. .................................. 111
4.3 Univariate linear model incorporating all 45 experimental conditions. 120
4.4 ANOVA table corresponding to the model described in Table 4.3. ... 121

5.1 Results of applying Tukey’s HSD to Hypothesis 2. ........................... 128
5.2 Results of applying Tukey’s HSD to Hypothesis 3. ........................... 130
5.3 Results of applying Tukey’s HSD to Hypothesis 4. ........................... 131
5.4 Results of applying Tukey’s HSD to Hypothesis 5. ........................... 133
5.5 Results of applying Tukey’s HSD to Hypothesis 6. ........................... 134
5.6 Results of applying Tukey’s HSD to Hypothesis 7. ........................... 136
5.7 Substantiated conclusions to the seven hypotheses introduced in Chapter 4 concerning the relative generalization performance of network independent vs. dependent features in the context of 5-state, fully-connected, HMM-based Internet traffic classifiers.
List of Figures

2.1 Three different HMM topologies described in recent Internet traffic classification literature. .................................................. 18
2.2 A transaction diagram of the packet size feature employed in [15]. 23
2.3 A transaction diagram relating the concept of a “message” to that of a “packet” feature. .................................................... 29
2.4 A transaction diagram depicting the usage of stochastic message padding as a means of obfuscating the packet-size traffic signature. 30
2.5 A transaction diagram describing the concept of processing time as a traffic classification feature. ...................................... 31
3.1 Histogram of maximum packet size, “Web” traffic class. .......... 47
3.2 ECDF of packet inter-arrival time, small (576 bytes) and large (1500 bytes) MTU conditions. .............................................. 49
3.3 A graphical representation of the trace described in Table 3.3. .... 54
3.4 A contrast of different Internet traffic serialization mechanisms. 62
3.5 A graphical representation of a fully connected 5-state Hidden Markov Model. ................................................................. 66
3.6 Plots of various parameterizations of the 2-parameter Γ probability density function. ......................................................... 74
3.7 A HMM-based Internet-traffic classifier architecture. .......... 77
3.8 Probability density function of Student’s t distribution. .......... 89
3.9 A figure of the empirical $F$ distribution for various combinations of $P$ and $N$. ................................................................. 94
4.1 Network topology employed during data set generation. .......... 98
4.2 Transmission characteristics retained by Tcpflow. .................. 101
4.3 10-fold cross-validation yielding 10 sets of aggregate $F$-scores. 108
4.4 Superimposed plots of standardized aggregate $F$-score per experimental condition. ......................................................... 115
4.5 Aggregate $F$-score vs. Factor Level without weighting. ......... 117
4.6 Aggregate $F$-score vs. Factor Level with weighting. ............. 118
5.1 Mean aggregate $F$-score for each experimental conditional. Note that bold typeface does not indicate a statistically significant difference exists between any of the observations. ......................... 124
6.1 A WireShark view of a real YouTube traffic trace. .................. 145
Chapter 1

Introduction

The maturation of widely-available broadband Internet access infrastructure is proving to have a similarly transformative effect on how the world communicates in the 21st century as the widespread availability of reliable voice communication did a century ago. During a significant portion of the 20th century, voice communications was the primary driver of global telecommunications research, providing a vast improvement over the telegraph and postage-based mediums which preceded it. Coinciding with the digital revolution and the advent of the personal computer, a need to communicate data (in addition to voice) began to surface. This shift away from voice to data coincided with a shift in research focus and technological development, eventually leading to the creation of the modern Internet. A medium which today cannot adequately be described as a mere communications network, but rather as an enabler of new technologies like remotely-operated surgical procedures [43], efficient dissemination of vast scientific knowledge [28] and greatly improved audio and visual communications [12]. As a global network, it is also of tremendous political importance, spurring philosophical debate regarding issues as diverse as network neutrality, government censorship and even human rights as it pertains to access to information and services available (sometimes exclusively) on the Internet. Unfortunately, the
multitude of different types of services which are made possible by the Internet and the various environments it operates in (both political and physical) pose a significant difficulty to the network’s operation, as the needs of such vastly different usage scenarios often conflict. Management of this problem is only hampered by the rapid rate at which new Internet-based technologies and services are developed. If the Internet is going to maintain its torrential pace of technological evolution (and society, as a consequence, continue to reap the benefits) solutions to these problems will need to be developed. The first step in that direction is to develop techniques which can be used to accurately and unobtrusively differentiate between the multitude of services available.

The situation is analogous to that experienced on motorways during peak hours of travel where congestion results due to the inability of the underlying infrastructure to cope with demand. Such a scenario shares many of the same traits as the performance degradation that results from TCP/IP-based networks struggling to mitigate the conflicting requirements of different types of services. Namely, unexpected delays/stoppages throughout the network, reduced average throughput and overall customer dissatisfaction with both the utility of the infrastructure as well as any services which rely upon it. For these reasons, the research domain most concerned with facilitating understanding of these phenomena refers to such scenarios as ”Internet traffic” and much like its transportation infrastructure counterpart endeavours to better understand the cause of, and as a result devise solutions to, the resulting problems. This thesis is primarily concerned with assessing whether certain performance characteristics of a well understood Internet traffic classifier can be
improved via utilization of a novel set of Internet traffic properties.

Advancing the state-of-the-art in Internet traffic classification facilitates better utilization of existing network resources, more advanced network analytics leading to more efficient, secure and robust network architectures as well as an increased awareness of how the Internet enables, accelerates or otherwise affects social development. It is a particularly challenging problem, in part, because of the entrenched nature of the technology which supports the Internet, which places constraints on the nature of the types of solutions which could be readily deployed. To be considered a viable solution, a given approach must respect the real-time processing constraints of digital communication systems, the architecture of the technologies employed within modern Internet infrastructure, as well as the inherent difficulty of the classification problem itself. Despite these challenges, there is still tremendous value in advancing the state-of-the-art [16] as the benefits of pushing the frontier of understanding in the field have the ability to both improve the capabilities as well as advance the understanding of one of the greatest technological inventions in recent history.

This thesis extends the state-of-the-art by investigating the relationship between a specific class of Internet traffic features and classifier performance (specifically generalization capability) across different data sources. Possible interactions between said features and the operating environment are also explored in an attempt to address observations regarding degraded real-world performance of Internet traffic classifiers [8]. A statistically rigorous experimental design and accompanying analysis procedure are employed, both of which also constitute novel contributions to the field and serve to further distinguish this work from the state-of-the-art.
1.1 Contributions

This thesis will determine whether a novel combination of features can be used to improve upon the generalization performance of 5-state, Hidden-Markov Model-based Internet traffic classifiers. With respect to the granularity of the classification to be performed, the goal will be to differentiate between specific sources of Internet traffic, going as far as to attempt to classify example traces by their generating application. Emphasis will be placed upon utilizing features which are thought to be independent of and thus likely unaffected by changes in the underlying network. Such features are worthy of further consideration/study as it unknown whether their usage will lead to improved performance. Given the goal to identify the generating application of a given trace, if the features utilized are truly reflective of the application behaviour, the underlying network configuration should exert no influence over the classification task. However, empirical evidence suggests otherwise and it is this discrepancy which warrants further investigation.

It is important to note that significant emphasis is placed on assessing generalization performance which has been specifically identified as a particular weakness of existing classifiers [54]. A somewhat unique definition of generalization performance is employed which encompasses an additional aspect of classification accuracy above and beyond that which is present in the state-of-the-art [51]. In this work, generalization performance then is defined as classification accuracy with respect to both how independent observations are classified as well as how said performance is maintained across different network configurations. These two facets are manifested as a number
of different experimental conditions wherein the same classifier architecture is tested against observations drawn from fixed sets of traffic types sampled across different network configurations. Superior generalization performance is characterized by both highly accurate classification between traffic types within a single network configuration as well consistent performance across different network configurations, whereas poor performance can be attributed to any combination of poor accuracy or highly variable results across network configurations or between traffic types.

Six different types of Internet traffic as generated by six different applications will be used to represent classes of traffic with conflicting requirements which would likely benefit from reliable disambiguation. Generalization performance will be measured via testing of classifiers utilizing three different sets of features across 45 different experimental conditions meant to mimic different real-world scenarios. Relative generalization performance will be assessed using a statistically valid experimental design and accompanying analysis in conjunction with a well-understood metric. Statistically valid conclusions regarding the relative performance of said feature sets (including those referenced in contemporary publications [8, 15, 46]) will be achieved via generation of performance metrics. All data utilized in this study was generated specifically to address the research questions outlined above and as such does not suffer from the same degree of uncertainty present in other studies [3–5, 7, 8, 10, 13, 15, 17, 22, 23, 29, 41, 46, 47, 60, 64, 66]
1.2 Conclusion

The flexibility of the Internet to deliver such a vast array of services is both a testament to the foresight of it’s compartmentalized design as well as continual challenge to manage. A victim of it’s own success, it has become so integrated into many facets of the everyday lives of so many that there are significant real-world implications for research into unobtrusive means that can be employed to enhance it’s capabilities. Internet traffic classification research is an example of such a pursuit as the techniques and knowledge gained through study lead to improved methods for traffic classification as well as a greater understanding of how the Internet is being utilized in general. The research described herein addresses open questions in the field as well as serving as an example of a more structured and statistically valid approach. As such, this work can serve as an example of how to both maintain the momentum often found in relatively young fields while also incorporating a more rigorous methodology often associated with more mature research domains.
Chapter 2

State of the Art

2.1 Introduction

By design, the underlying architectural components of the Internet (the TCP/IP protocol stack) lack a universal mechanism for differentiating or describing the nature of the data they are employed to communicate. As alluded to in the previous chapter, this deficiency is a result of a conscious decision whereby the protocol stack was designed to be agnostic of the types of information communicated. Such an approach has two advantages. First, it serves as a means of enforcing the interface between adjacent layers in the protocol stack, thus facilitating tremendous flexibility with respect to accommodating different protocols in the future. Second, it greatly simplifies the scope of the design by way of explicitly excluding all aspects of packet priority, which is only an issue when the underlying infrastructure cannot meet the needs of the services being delivered. Had this concept not been considered during the design, it is difficult to envision the architecture scaling as well as it has thus far. The need for Internet traffic classification then is not a result of any gross design deficiencies but rather a product of the longevity of the core architecture, the explosive growth of the Internet and the failure of the underlying network infrastruc-
ture to keep pace with the demands of services and applications. As a result, the discrepancy between the design philosophy of the TCP/IP protocol stack, the usage scenarios of modern Internet users and the capabilities of deployed infrastructure result in the sub-par performance of use cases like Voice-Over-Internet-Protocol (VOIP) communication, web browsing, Peer-to-Peer file sharing and media streaming.

More specifically, sub-par performance arises when the demands of different types of Internet traffic fail to be met by the protocol stack in conjunction with the underlying network infrastructure. While it is tempting to assign blame for such performance to a lack of capacity, such an assignment would be overly simplistic [9, 52]. Rather, the conflicting nature of different types of traffic also plays a significant role. As an example, consider the conflicting requirements of BitTorrent (a peer-to-peer (P2P) file sharing application) and Skype (a VOIP service). Skype is sensitive to network jitter and latency as it relies on the timely delivery of small packets containing digitized speech, while the throughput requirements are relatively easily satisfied by modern broadband networks. In contrast, BitTorrent is quite tolerant of fluctuations in the rate at which packets are transmitted, though it has the potential to consume significant bandwidth if configured accordingly. When the two applications are run simultaneously, the capacity consumption of BitTorrent impairs the ability of the network to meet the latency/jitter requirements of Skype, resulting in unacceptably poor voice call quality. One potential solution is to throttle or otherwise prioritize Skype traffic over BitTorrent traffic. However, such an approach hinges on the ability to reliably identify the two respective traffic types in real-time, such that relative priorities can be assigned. This seemingly simple task is actually quite difficult due
to the very nature of the underlying protocols which comprise the modern Internet. Addressing this shortcoming is one of the primary motivators of contemporary Internet traffic classification research, which attempts to find ways of identifying Internet traffic as early in the transmission lifetime as possible to support priority assignment schemes which appear to operate instantaneously from an end-user’s perspective.

In addition to addressing the operational issues highlighted above which require solutions that can operate in a near real-time fashion, the ability to identify traffic types in an offline capacity is also of significant value [23]. Such research facilitates greater understanding of how the Internet is being used as a whole and as such can be employed to assist in growth forecasting, performance monitoring and security-related investigations. However, care must be taken to consider the target deployment scenario, as this exercise often uncovers unique constraints which may prevent the utilization of certain techniques. As an example, if the goal is to identify Internet traffic in an online fashion as a means of mitigating the conflicting requirements of different types of services, the ability to identify a given service quickly is of considerable importance. As such, it may be beneficial to trade-off some degree of accuracy for substantial gains in early identification capability. Conversely, if the goal is to identify all of the traffic types which traversed a given network in order to arrive at an understanding of which types of traffic dominate, the same trade-off is not justified as the accuracy of the resulting classification is of higher value than the immediacy with which it was computed. The motivation behind Internet traffic classification research can therefore significantly influence the approach taken and is an important aspect to consider when surveying the field.
2.2 Literature Survey

In a survey of Internet traffic classification techniques which employ some form of machine-learning [51], Nguyen and Armitage utilize five qualities (Machine-Learning Algorithm, Features, Data Traces, Traffic Considered, Classification Level) to qualitatively summarize surveyed works. Inspired by this approach, the following attributes were identified as serving a similar purpose in the context of the survey described in this section.

Classifier Architecture

Classifier Architecture refers to the type of mechanism employed which takes as input a set of features extracted from a traffic trace and outputs a label designating the classifier’s assigned traffic class. Popular approaches include Deep Packet Inspection (DPI), Machine-Learning Algorithm (MLA), Hybrid and Composite (H&C) techniques and more hardware-oriented approaches. Choice of architecture often reflects an intended operating environment as well as the nature of the features to be used [7, 23, 29]. In general, classifiers which are computationally expensive to operate are often suitable for offline operation while efficient classifiers are preferred for real-time deployments. Similarly, DPI can only be utilized if the contents of the packets are present, while many MLA-based approaches assume visibility into both directions of a given traffic trace. In addition, it is important to note that hardware-based solutions can be employed to address some of the limitations computationally expensive algorithms encounter when deployed in real-time scenarios [59]. The survey to
follow will focus on MLA-based approaches, though both DPI and H&C will be discussed briefly. Approaches relying on customized hardware are considered out of scope and as such are not discussed.

Features

Features are characteristics of Internet traffic that can be used for identification or classification purposes. An example of a feature is the IANA assigned port number associated with a given traffic trace. A significant amount of research \[2,11,16,22,42\] has been conducted into assessing the relative merits of different classes of features, resulting in the following major categories:

**Packet-Level** features that are in some way derived from analysis of packets within a given traffic trace, irrespective of the source/destination. Examples include mean packet size and characteristic bit-sequences.

**Flow-Level** features are Packet-Level features that have been altered in some way to reflect directionality of packet flow between a single end to end connection. Examples include ratio of sent to received packets and tuples of packet size, direction.

**Connection-Level** features build upon Flow-level features but extend the concept to include multiple concurrent connections. Examples include tuples of packet size, direction, concurrent connection number and ratio of all incoming packet sizes to outgoing packet sizes across all concurrently active connections within a given trace.

Features can be categorical, discrete or continuous in nature. The choice of
which features to employ is often related to the operating domain of interest as well as the intended classifier architecture. As an example, some packet-level features such as mean packet size (which are computed over an entire traffic trace) are not well suited to identifying traffic early with respect to the lifetime of the connection, while characteristic bit sequences can often be very effective in this scenario if the sequence occurs early in the trace [30].

Data Sources

Data Sources are the sets of traffic traces utilized for testing and training purposes. The generalization capability of a given classifier is often evaluated via assessment of performance across multiple data sources. Popular types of data sources are Academic, Commercial and Residential. Data Sources play a role in defining what information is available for feature extraction, which is largely dependent upon from where the traces comprising a Data Source were collected with respect to the network topology.

Traffic Considered and Classification Level

The traffic considered and classification level are inter-related concepts. The traffic considered is simply the targeted types of traffic in which a given classifier is meant to operate. Sometimes this set includes an “Unknown” class whereby a given classifier should identify traffic traces that are unrecognized. Classification level refers to whether a single type, multiple individual types or aggregations of specific types are to be classified.
These attributes serve as a solid framework from which to further examine recent research with respect to the stated goals of this thesis. Given the focus of this work on classifying different sources of Internet traffic traces by their generating application, all relevant works surveyed are similar enough in terms of “Classification Level” and “Traffic Considered” to permit combining these two characteristics. In addition, given this work’s emphasis on utilization of a statistically sound experimental design, an additional category will be added which will attempt to summarize how classifier performance is evaluated and how conclusions regarding performance are substantiated. In subsequent sections, each qualitative aspect will be discussed in more detail, citing representative works which anchor the breadth of the category under discussion. After each category has been introduced and thoroughly discussed, it will be possible to distinguish the approach taken in this thesis with respect to the state-of-the-art.

2.2.1 Classification Level Considered

Within the domain of Internet traffic classification research, there exists three major themes with respect to the types of traffic targeted for classification, henceforth referred to as “single”, “many” and “aggregate”. The “single” case focuses on identifying a specific application and is typically motivated by two similar yet often polarized pursuits; a desire to increase the ease in which the targeted traffic type can be detected to retard (i.e., in the case of P2P file sharing) or enhance (i.e., in the case of VOIP applications like Skype) the performance of the generating application. The “many” and “aggregate” classes are similar in that they both involve the
pursuit of classifiers which can identify multiple classes of traffic, the key difference between the two being the composition of the different classes. The “many” variety targets identification of specific generating applications or protocols (e.g., HTTP, DNS, eDonkey), while the “aggregate” theme is comprised of classifiers which attempt to identify groups of traffic sharing similar characteristics (e.g., Chat, E-Mail, File Transfer). Although this thesis aligns most closely with the “many” approach, examples of all three classes are described below to provide a sense of how each one manifests itself in the state-of-the-art.

The ability to identify a specific class of traffic is of great importance to network operators as they begin to make greater use of the Internet as a means of delivering services traditionally served via traditional, dedicated infrastructure (e.g., telephone, television). As a form of VOIP service, traffic generated by Skype is an example of such a class. In [10], Bonfiglio et al. describe and compare three classification approaches for Skype traffic: one which leverages the randomness introduced by ciphers within the protocol, another which relies on traffic features like packet size and inter-arrival time and a third based on Deep Packet Inspection (DPI). The first approach utilizes a Chi-Square Classifier, which in short is able to test whether the observed distribution of packets matches expectations. Since a significant portion of Skype traffic is encrypted and therefore effectively random, the ability to determine that certain packets in a repeated sequence are sufficiently random can be used to construct features which can be utilized for classification. The second approach utilizes a Naïve Bayes’ Classifier operating over features like packet size and inter-arrival time. The third approach utilizes DPI in conjunction with Skype-specific signatures
to establish ground-truth. Performance evaluation is conducted utilizing two data sets, one from a campus network (the reference condition) and one from a commercial ISP. The campus data set consists primarily of web, mail and bulk file transfers, while in addition to the aforementioned traffic types, the commercial data set contains P2P file sharing and competing VOIP services. Relative performance is reported via tallies of the number of identified Skype flows as well as false positives and negatives for each classifier and data source. Bonfiglio et al. report that, while in isolation, neither the Naïve Bayes classifier or Chi-Square Classifier perform particularly well overall, though each seems to be particularly well suited to identifying specific Skype use cases (CSC=UDP, NBC=TCP). When combined, the resulting composite approach achieves acceptable classification performance on both TCP and UDP-based Skype traffic. Unfortunately, no comparisons to existing approaches were reported so it difficult to place the results in context to the state-of-the-art.

The work of Cheng and Wang [13] is an example of both a classifier occupying the “many” class as described above as well as one employing a novel feature set. In [13], Cheng and Wang describe a classifier which operates using features derived from the connection patterns exhibited by hosts engaged in communicating application-specific traffic. They cite a deficiency of previous classifiers which utilize connectivity graphs between servers as features wherein traffic generated by servers hosting multiple applications was not able to be classified. By analyzing both the connectivity between servers as well the number of concurrent connections between hosts per application, servers hosting multiple types of applications could be identified. Two different data sources were employed corresponding to different locations
within the network topology. The first source was from the network backbone perspective while the second from the perspective of an edge node. Via analysis of the connection patterns for various services, Cheng and Wang found success in developing a relatively efficient means of distinguishing SMTP, POP3, Web and BitTorrent traffic, though no numerical summary regarding classification performance was presented. Like [10], no comparisons to existing approaches other than citations were presented rendering any metric-based comparison to the state-of-the-art impossible.

In contrast to the limitations cited above, in [36], Kim et al. analyze the relative performance of nine classification architectures and associated features sets across seven different data sources. Of the nine classification architectures studied, two are freely available software packages, six are previously studied machine-learning-based algorithms and one is a new machine-learning approach utilizing a Support Vector Machine (SVM). As opposed to the identification of a single traffic type as in the case of [10] or multiple traffic types in the case of [13], Kim et al. target classification of “aggregate” groups of related applications/protocols like Mail, Chat, Streaming and P2P. The analysis is broken down into three sections, with the first two sections targeting the existing classification packages CoralReef and BLINC, while the third section concentrates on machine-learning-based classification algorithms. While both CoralReef and BLINC have a predefined set of features they utilize for traffic classification, the remaining seven classifiers require a set of features to be specified. Kim et al. determine these features via application of the CFS algorithm, arriving at a set of six to ten traffic features from an original set of over 35. Results are presented via plots of precision, recall and F-Measure (defined in Section 4.3.1.2)
for the different experimental conditions of interest. Of all the machine-learning algorithms employed, the SVM approach was found to be superior, with average accuracy of greater than 98% reported across all traffic groups when one data source is used and 94.2% when multiple data sources are employed. The analysis concludes with an examination of generalization performance, where Kim et al. note a significant decrease (i.e., between 10%-30%) in average accuracy when the test/training sets used to evaluate the SVM-based classifier are independent, thus reaffirming the importance of representative test/training sets as well the detrimental affects of over-fitting on generalization capability of machine-learning-based classifiers [19].

Each of the three previously presented works is representative of the three most prevalent approaches regarding the granularity of traffic classes considered in state-of-the-art research. While they were intentionally introduced above in order to move from one extreme to the other, they similarly serve as good baselines with respect to subsequent sections. Of particular interest is the utilization of different classes of traffic features, machine-learning algorithms, data sources and experimental designs and performance metrics. Themes such as reliance on DPI to establish ground-truth as well as the absence of statistically valid experimental designs will also resurface in subsequent sections. Via exploration of these themes in recent literature it is possible to gain an appreciation of how to improve upon the state-of-the-art.

2.2.2 Machine-Learning Algorithms Employed

Over the last decade, investigation into the relative merits of a myriad of different machine-learning techniques, and by extension classifier architectures, has
dominated the Internet traffic classification research literature [16,51]. Although this study exclusively employs a HMM-based classifier architecture, the motivation behind such a decision was more so to eliminate extraneous variables from the experimental design rather than to endorse or advocate that such a model is ideal. Further motivating this decision was consideration of the types of traffic to be classified, the features to be employed as well as the goals of the study. Evidence of this same evaluation can be found via the following review of recent literature on the subject and as such reinforces the validity of categorizing contributions to the field in this way.

![HMM topologies](image)

Figure 2.1: Three different HMM topologies described in recent Internet traffic classification literature.

As an example of a study also utilizing HMM-based classifiers, in [46], Maia et al. employed a 2-stage, HMM-based architecture as a means of capturing both
the stateful correlation structure between features during the early or “handshaking” stages of a given trace as well as the “steady-state” statistical aspects that emerge later. Bidirectional traffic traces were analyzed in order to produce serialized vectors of observations of the form \( \text{direction, packetsize, inter-arrivaltime} \). The 2-stage HMM topology combines the approaches taken by Wright and Dainotti [15, 66] into a single model, as depicted in Figure 2.1.

Multiple types of traffic including HTTP, Telnet and that generated by the online game Counter Strike were studied. Results were compared against several other classifier architectures, all utilizing the same feature sets. Coincidentally, the strongest classifiers reported were all HMM-based. The results indicated a relationship between the number of packets considered and the average accuracy, resulting in over a 15% improvement from analyzing sequences of five packets (62.5%) to 13 packets (80.0%) when the 2-stage architecture is employed. When the entire sequence is analyzed, the 2-stage architecture classifies 95.5% of all traces correctly, while the two other HMM-based architectures (profile and fully-connected topologies) classify 87.5% and 85.4% correctly respectively. As a result, Maia et al. claim that the 2-stage HMM topology improves over the state-of-the-art.

In contrast to the HMM-based architectures described above, in [41], Li et al. employ a SVM-based, semi-supervised approach which addresses a few shortcomings of the supervised, HMM-based techniques employed above. Li et al. cite the difficulty in obtaining reliably labelled datasets as well as the inability of supervised learning-based architectures to classify novel types of traffic as primary motivators for their research, while acknowledging that unsupervised learning approaches generally ex-
hibit weaker classification accuracy overall. Two different feature sets were employed depending upon the directionality of the features within each set (e.g., unidirectional vs. bidirectional). In accordance with the definition of an “aggregate” classification level given in the preceding section, Li et al. defined traffic types in a general way, resulting in classes like “Web”, “Mail”, “Game” and “P2P”, each of which was composed of traffic from multiple applications. In contrast to the approach taken in [46], due in part to the semi-supervised “co-training” approach, the resulting SVM-based classifier produced comparable results while also possessing the capability of identifying traces which did not belong to any of the predefined traffic classes. Average classification accuracy of the resulting classifier of the two feature sets was determined via a 10-fold cross-validation procedure, resulting in scores of 95.7% (bidirectional) and 94.9% (unidirectional) respectively.  

Li et al. reinforce the importance of their findings by citing the comparable performance of their SVM-based classifier to other contemporary architectures while noting that the constraint of a completely labelled data source is eliminated.

Given the plethora of classifier architectures and machine-learning-based methodologies employed in the field over the last decade, research into the merits of composite architectures has started to appear [4, 17]. Composite classifier architectures are comprised of several independent classifiers which may or may not be utilizing the same features and whose output is encapsulated in a decision rule. Thus, from an evaluation perspective the methodology is very similar to that em-

---

1 For consistency and as a result of independent evaluation in [37], the use of 10-fold cross-validation is employed in this thesis as well.
ployed with traditional classifier architectures. In addition, significant effort has been placed into developing reference algorithm implementations [32], which has facilitated meta-studies incorporating larger numbers of classifier architectures and associated learning methodologies. In [17], Dainotti et al. examined the relative merits of different methods for reaching consensus amongst the output of many classifiers. As in the case of [46], emphasis is placed on early identification of a given traffic trace, and as such, (at most) features are extracted from the first 10 TCP packets. Two distinct feature sets were employed depending upon the architecture of the constituent classifiers (i.e., full vs. partial trace observation). For classifier architectures that operated over partial sequences of observations, like the HMM-based architectures employed by Maia et al. in [46], packet size and inter-arrival time features were employed. The other set consisted of classifiers which could accommodate both partial sequences of observations in addition to statistics derived from the entirety of the trace. A mixture of application-specific and protocol-specific traffic types including BitTorrent, DNS, and HTTP were used for assessing classification accuracy. Danoitti et al. report that the best performing composite classifiers result in a 21% improvement in classification accuracy when the number of TCP packets considered is one and 4% when two packets are considered. When the number of packets is greater than two, performance is slightly better than that reported in [41,46], with average accuracy of 98.4%.

The aforementioned works are only a small fraction of the state-of-the-art of machine-learning techniques for Internet traffic classification. They serve as examples of three broad classes of studies each employing different types of classifier architectures. The work by Maia et al. exemplifies supervised classifier architectures meant
to operate on an observation vector based on some time unit (typically, but not necessarily packet-aligned) and as such can classify traces via examination of as few as the first four packets. The work of Li et al. is an example of a semi-supervised classifier which has distinct training benefits but requires observation of the entire trace before producing a classification. In [17], Danoitti et al. explore different techniques for combining multiple classifier architectures to improve upon the state-of-the-art. All three approaches reported reasonably good classification accuracy (e.g., ≥ 90%), though little consistency existed with respect to the types of traffic targeted and the sources of the datasets themselves, thus making direct comparisons difficult. Despite these differences, each work surveyed reported similar performance while also employing similar strategies for establishing ground-truth.

2.2.3 Data Sources and Ground Truth

A significant obstacle to overcome when conducting Internet traffic classification research is the lack of standardized datasets. Though there are publicly available repositories of Internet traffic [14, 38], these repositories contain unlabelled traffic traces only and as such must be further processed to be used to train and test supervised or semi-supervised MLA-based classifiers. The same is true for private datasets which are collected solely for research purposes, as is the case in [15, 17, 46] where traces taken from academic networks were utilized. This preprocessing step has become known as a establishing “Ground-Truth” as it requires extracting and classifying individual traces from large, unlabelled datasets. This is achieved by manual means [46], DPI [10, 15, 17, 24, 40, 42, 54, 60] or through utilization of another classi-
tier [13] or some combination thereof [3]. To eliminate the uncertainty that arises due to using one classification mechanism to establish ground truth for the purposes of assessing another, some researchers have opted to generate their own traces using the generating applications themselves [3]. To further add to the complexity of the matter, there are further sub-categories of data sources with respect to the availability of bi-directional traces, as well the point at which traffic was intercepted (e.g., the edge or core of a given network). If any sort of consensus amongst published works is to be reached, the choice of dataset, the means of establishing ground-truth and any consequences which stem from those choices must be well understood.

In [15], Dainotti and Rossi employ a HMM-based classifier architecture in conjunction with two packet-level features: packet size and inter-arrival time. Traffic

![Diagram](image)

**Packet Size Observation Vector = [1000, 1500, 1500, 1000]**

Figure 2.2: A transaction diagram of the packet size feature employed in [15].
classes targeted include a mixture of online games (Age of Mythology, Counter Strike), P2P file sharing and media streaming (eDonkey, PPlive), messaging (MSN, SMTP) and web browsing protocol traffic in the form of HTTP. Training and testing trace generation depended upon the specific traffic type. With the exception of the Age of Mythology dataset, which was generated at the Worcester Polytechnique Institute, the remaining datasets were generated via monitoring of the University of Naples, Frederico II campus network. A mixture of port-based and IP-based heuristics were used to establish ground-truth. Like the other HMM-based architecture described in Section 2.2.2, classification performance was measured via application of a cross-validation procedure, resulting in the construction of a confusion matrix. Classification accuracy ranged between 90.24% (eDonkey) to 100% (Age of Mythology). Since there was only a single source of traces for each traffic type, no analysis regarding generalization performance across network configurations could be performed.

In contrast, in [3] Alshammari and Zincir-Heywood investigate the relative generalization performance of two machine-learning-based classifier architectures utilizing two sets of traffic features on two (nearly identical) sets of traffic traces across two independent network configurations. Although the focus of the study was on assessing the classifiability of one particular class of traffic (specifically, SSH encrypted traffic) the experimental design employed serves as an example of greater emphasis being placed on the possible influence of a given source on the resulting generalization performance. Like [15], ground-truth had to be established, except a commercial tool utilizing DPI was used. Consistently strong classification (e.g., greater than 97%) accuracy was achieved across all experimental conditions as determined by a 10-fold
cross-validation procedure. The relatively consistent results observed between the two independent sources is highlighted as an avenue for future research, acknowledging the potential role a given network configuration may play in influencing classification accuracy.

In [8] Bernaille et al. squarely target the question of generalization performance across network configurations. The study involves examining the classification performance of three different types of clustering-based algorithms (k-Means Probabilistic, Gaussian Mixture Model (GMM), HMM), across six different sources of traffic traces. Held constant throughout the study were the feature set (sizes of the first four packets) and traffic types. Prior to presenting the results of the their experiment, Bernaille et al. assess the distribution of their features across each source and traffic type and conclude there is little evidence of source-specific variation, thus justifying the utilization of their feature set in the experiment to follow. As is the case with traditional machine-learning-based classifiers, a calibration (or training) phase was conducted to iteratively refine the clustering parameters using a subset of the available dataset. Although classification accuracy was generally quite good (e.g., greater than 90%), there was evidence of source-specific performance degradation which affected the k-Means and HMM-based clustering algorithms exclusively. The “M2C ADSL” data set proved particularly problematic, resulting in performance degradation on the order of 25% in some cases. Bernaille et al. utilize this result as justification for the superiority of the GMM-based classifier, as it exhibited relatively consistent performance across all data sources (read: network configurations) tested. One interesting aspect of the study was the approach taken to assess generalization
performance with respect to classification of novel application traces that were not utilized during calibration. The traces employed for this purpose were manually generated, though details regarding the underlying network configuration and how it relates to the other data sources utilized was not published. In this case, the HMM-based clustering algorithm proved to correctly identify the novel application traces as "unknown" more consistently than the GMM-based algorithm.

The three studies introduced in this section demonstrate reasonably consistent approaches employed in order to generate data sets suitable for research into machine-learning-based Internet traffic classification. All three studies rely (at least in part) on establishing ground-truth via utilization of DPI, port-based analysis, heuristics or some combination thereof and as such are somewhat constrained by the types of traffic they can accommodate. In [8], Bernaille et al. attempted to address this shortcoming via utilization of manually generated traces, however said traces were only incorporated into the test set and as such only used to gauge each clustering algorithm's ability to detect genuinely unknown traffic types. Despite these shortcomings, the works surveyed in this section reflect a spectrum which is fairly representative of the field as a whole. A spectrum that is anchored by the desire to construct a composite dataset with as many traffic types as possible, centred on a balanced design where target traffic types are equally represented across multiple sources and ending with a scenario where, in addition to a balanced design, additional sources (and traffic types) are utilized in order to assess generalization performance.
2.2.4 Features

Along with classifier architecture, investigation into the relative strengths of various traffic features constitutes a significant portion of contemporary research in the field. Rather than simply enumerate a selection of the most well-studied or utilized features found in recent literature, this section will focus on establishing criteria which may be used to categorize said features into sensible classes. Of particular interest are characteristics which impose restrictions on the types of traffic which can be classified, possible deployment scenarios within a given network as well as the type of classifier architecture which can be used. Due to the inconsistent nature of the experimental design methodologies employed and considering that most feature sets are reported as exhibiting similar classification accuracy, there is little value in attempting to categorize by performance. As such, it can be assumed that the features discussed have been reported to exhibit comparable classification accuracy under a given experimental condition.

Early identification of a traffic trace is an important motivating factor behind many recent studies in the field. In [7], Bernaille et al. utilize the size of the first five “data packets” communicated via TCP between and two end points as features for classification purposes. In this context, “data packets” refers to packets which contain valid user data which are typically observed after the 3-way TCP handshake procedure [55]. Given the requirement to contain valid user data, subsequent TCP signalling packets are also excluded from the feature set. In addition, both directions of traffic contribute to the first five “data packets” criterion. An unsupervised clas-
sifier architecture is employed, utilizing the $k$-means clustering technique to create clusters from training data. Classification is performed via computing the Euclidean distance of a given observation set to the nearest cluster, which is then interrogated to determine the most common traffic type within, in order to determine the classification. As such, the inherent sequence of the first five packets is not utilized for classifications purposes. Preliminary performance results indicate average classification accuracy well below other contemporary approaches (82%), though it is important to note that the classifier was unable to properly classify “pop3” traffic at all, thus heavily influencing the measure. Competitive performance was observed for HTTP (99%), Kazaa (95.25%), NNTP (99.6%) and SSH (96.92%) traffic. Though the preliminary performance is worrisome, the relatively space-efficient feature set and corresponding classifier operational efficiency demonstrate the potential of TCP payloads to be used to identify corresponding applications. Bernaille et al. highlight these qualities as motivating their approach for an alternative to classical port or DPI-based classification.

Building upon the work of Bernaille et al., Li and Moore use a feature selection process to extract the 12 best features from a superset of over 248 features concerning the first five TCP packets observed within a trace [40]. The resulting feature set is composed of traditional port-based identifiers, TCP connection based statistics like the number of packets sent with the PSH flag set, the median sum of bytes in all IP packets and variance of all bytes sent from the client to the server. Like [7], the sequence of observations is not utilized as the eventual features used for classification are scalar quantities as opposed to vectors corresponding to the first
$N$ packets observed. A C4.5 decision-tree-based classifier is employed and validated using 10-fold cross-validation. With respect to work already studied, comparable classification performance is observed with consistently high classification accuracy scores (greater than 95%). Another example of the utilization of C4.5 decision trees can be found in [60] where Shen and Fan extract features not from TCP packets, but from consecutive and uninterrupted aggregations of packets in a given direction. Referred to “messages” (see Figure 2.3), these features are hypothesized to be better reflect the protocol employed by an application than other features studied thus far.

Figure 2.3: A transaction diagram relating the concept of a “message” to that of a “packet” feature.

Shen and Fan used Singular Value Decomposition to extract a set of nine features from a set of over 246 inspired by the work of Auld et al. [5]. Unlike the approach taken above by Li and Moore, where only the first five packets were considered, fea-
tures were extracted from the entirety of trace. In total, six different applications are targeted (HTTP, BitTorrent, QQ, TELNET, SMTP, FTP). Performance was assessed via cross-validation, with very promising average case classification performance of 99%. It is important to note that such a classifier would be of little practical use in an early classification scenario due to the necessity to observe the entirety of a trace before rendering a classification.

In addition to the constraints regarding possible deployment scenarios, another weakness of the approaches described above is the susceptibility of such methods to counter-measures employed by an application, such as stochastic message padding (see Figure 2.4) [67].

![Figure 2.4](image)

Figure 2.4: A transaction diagram depicting the usage of stochastic message padding as a means of obfuscating the packet-size traffic signature.

By introducing stochastic messaging behaviour into the protocol structure,
the representative nature of derived features is nullified. Waizumi et al. address the
aforementioned shortcomings via studying the viability of using the time taken by an
application to process and respond to messages as features for classification purposes
(see Figure 2.5), while only observing the first $N$ messages of a given trace [64], with $N$ varying from one to seven depending upon target application.

![Diagram](image-url)

**Figure 2.5:** A transaction diagram describing the concept of processing time as a
traffic classification feature.

With respect to classification architecture, a type of artificial neural network approach called Learning Vector Quantization is employed to iteratively refine
a decision function through the presentation of training samples. The sequence of
observations with a given trace persists through the feature extraction process and as
such plays a role in the resulting classifier. In all, 10 different applications are studied, including FTP, SMTP, HTTP and RTSP. Performance is assessed via comparison
against three other feature set vectors, consisting of:

1. The number of packets within each message.

2. A combination of (1) above and the proposed “processing time” features.

3. Inverse packet size.

For all but one traffic type, both the proposed “processing time” and “combination” features were outperformed by the inverse packet size feature. The single exception is in the case of the RTSP traffic type which is most accurately classified via the combination approach. Although performance was reported as being inferior to other established methods, the resulting classifier is more versatile with respect to deployment scenarios due to the ability to operate on the first $N$ messages within a trace.

The features described above represent a small but important fraction of those employed in contemporary Internet traffic classification research studies. They demonstrate the relationship between the features under consideration, classifier architecture employed, targeted traffic types and restrictions regarding suitable deployment scenarios. Given the complexity of the relationship, the proportion of state-of-the-art research devoted to furthering understanding of the topic is not surprising.

2.2.5 Experimental Design

In general, all of the works cited in the preceding sections possessed the same basic elements:

- A classifier and accompanying training procedure.
• A set of traffic features.

• A source of traffic traces and some accompanying mechanism of establishing ground truth.

• An assessment of the resulting classifier performance.

This section is centred on surveying the state-of-the-art with respect to the experimental designs employed. Emphasis is placed on assessing which key variables have been targeted, what metrics are used to measure classifier performance and how conclusions are presented and substantiated. Three different works are examined, each serving as representative examples of the different ways in which the aforementioned factors manifest themselves in state-of-the-art Internet traffic classification research.

The most common type of study in the field attempts to ascertain the performance of either a novel classifier architecture or set of traffic features or both utilizing traffic traces extracted and preprocessed from one or more data sources. In [47], Maiolini et al. propose using a \( k \)-Means clustering approach in conjunction with the TCP packet size, inter-arrival time, direction and presence of certain TCP control packets to generate a functional traffic classifier. They evaluate their proposed classifier against a combination of generated and observed traffic traces from academic, commercial and domestic sources. The main focus of the work is centred on assessing performance on both observable and encrypted traffic. The number of TCP packets considered was determined empirically, with sequences of length six deemed optimal. Two meta-classes of traffic were considered, a normal class con-
taining types similar to those discussed in previous sections and an encrypted class containing different types of applications communicating via an SSH tunnel. Two different data sources were analyzed, one with complete visibility into both directions of traffic flow, while the other is restricted to just a single direction and containing only IP header information. Confusion matrices and tables consisting of classifier accuracy by application are used to present results. Four different sets of results are presented, two concerning classifier performance under the two different sources of traffic, one concerning the effect of removing TCP control packets from the traffic traces and another concerning performance on encrypted traffic flows. For the data source where both directions of traffic were visible, average classification accuracy of 94.53% was achieved, while only 84% was achieved when one direction of the flow was visible. Similarly strong results were reported concerning the accuracy in which HTTP over SSH, SFTP and SCP were identified. Further, contrary to the researchers assumptions, removal of TCP control signalling was found to be detrimental to the detection of certain applications.

Like in [47], in [24], Este et al. evaluate the entropy of packet-level features with respect to the classification of Internet traffic. More specifically, they attempt to determine if the usefulness of packet-level features is affected by the point of observation within the network, or if the utility of said features has changed significantly over the span of eight years. Two sets of packet-level features are considered: the first set is focused on describing characteristics of the 3-way TCP handshake that occurs during the initiation of a TCP socket connection while the second set targets the first $N$ packets of a TCP connection to be observed after completion of the 3-way
handshake. The first set contains two features, the round-trip time between receipt of the SYN and SYN-ACK packets by two respective end points and a binary feature indicating whether the two end points reside within the same subnet. The second set contains more commonly studied features like packet size, inter-arrival time and packet direction. Entropy is assessed by way of two metrics: mutual information measure and conditional mutual information measure. Further, three different data sets are utilized, spanning eight years and occupying different locations within the network. DPI was used to establish ground-truth, resulting in unique sets of applications per source. In addition to assessing entropy across sources, three different classifier architectures were also evaluated, resulting in a total of twenty seven different experimental conditions.

Two sets of results are presented. The first concerns investigation into the entropy of features as a function of the number of packets considered across different data sources, while the second examined the relationship between data source, classifier architecture, feature and classification accuracy. In the first set of results, Este et al. utilize plots of mutual information by packet number to demonstrate a gradual decrease in mutual information from the first to the sixth packet across all data sources. With respect to the second set, results are presented by three plots of the average classification accuracy by feature (one plot for each classifier architecture), with multiple series within each plot corresponding to each data source. Conclusions regarding the entropy of each feature and the stability over time are substantiated in reference to the relative classification accuracy captured in each plot. Of all features studied, packet size and inter-arrival time are shown to be superior, with average clas-
sification scores as high as 95%, leading to the conclusion that packet-level features possess sufficient entropy and temporal stability to justify their utilization in traffic classifiers.

In both [47] and [24], no significant affects regarding the influence of different data sources with respect to classifier performance or entropy were identified. In contrast, in [54], Pietrzyk et al. report instances of significant degradation in classification accuracy when the classifier is trained using traces collected from one source and evaluated on traces from another. Utilizing a combination of flow-level and packet-level features, Pietrzyk et al. describe two different lines of enquiry, referred to as “static” and “cross-site”. The “static” line centres on investigating the affects of data source, traffic class, feature set, classifier architecture and the number of packets utilized on classifier performance. The “cross-site” investigation builds upon the “static” case, placing emphasis on examining the affects of training a classifier on sample traces from one data source and evaluating classification performance on another, resulting in 256 distinct experimental conditions. In the ”static” case, classification accuracy, precision and recall metrics are all employed to measure relative performance under the different conditions of interest. In general the results are in agreement with previously referenced work, with average classification accuracy of greater than 90% achieved via utilization of packet and flow-based features extracted from the first four packets of a given traffic trace. Results pertaining to the second line of enquiry are presented using a series of confusion matrices with the training data source on the Y axis and test source on the X axis. Each matrix describes a single scenario of “cross-site” performance under the aforementioned conditions. The most pertinent result is
that while there is evidence of slight accuracy degradation across all conditions in the
“cross-site” case, some specific traffic classes exhibit significant decreases independent
of classifier architecture or feature set. This finding leads to the conclusion that there
is the potential for over-fitting when training traffic classifiers under the conditions
described above and that investigation into features which are more resilient to this
shortcoming should be pursued.

The one unifying theme which binds the three previous works together (and
is equally as prevalent in state-of-the-art literature) is the multi-dimensional nature
of the experimental designs they employ. Experimental conditions consisting of mul-
tiple variables like classifier architecture, feature set, sequence length, test/training
methodology and data source are utilized with the goal of assessing differences in clas-
sification performance, as quantified by some combination of average accuracy, preci-
sion or recall. Conclusions regarding the relationship between variables and classifier
performance are substantiated using plots, confusion matrices and numerical compar-
isons. While such devices are quite effective at summarizing and/or visually realizing
experimental results, when the experimental design is stochastic in nature (as is the
case here), it is often necessary to employ some form of hypothesis testing in order
to disqualify the possibility of observing said results by chance alone. With respect
to state-of-the-art Internet traffic classification research, this possibility has not been
addressed and as such is an opportunity for further advancement of the field.
2.2.6 Beyond the State of the Art

Table 2.2.6 summarizes the works surveyed above. The final row of the table describes how the contributions of this thesis compare. Strong similarities exist between this work and the state-of-the-art in terms of classifier architecture, traffic considered and to a lesser extent features employed. Significant differences with respect to experimental design and data sources/ground-truth establishment are the result of a conscious decision to approach the problem differently than had been done in any of the referenced works. This was done to address some of the shortcomings alluded to above, namely concerning the integrity of Data Sources employed and how Ground-Truth was established, as well as the quality of the experimental design employed.
<table>
<thead>
<tr>
<th>Ref.</th>
<th>Classifier Architecture</th>
<th>Feature Class</th>
<th>Classification Granularity</th>
<th>Data Source Category</th>
<th>Ground-Truth</th>
<th>Performance Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>Chi-Square, Naive Bayes, Composite, DPI</td>
<td>Packet-Level</td>
<td>Single</td>
<td>Academic, Commercial</td>
<td>DPI</td>
<td>Comparison of False-Positives/Negatives</td>
</tr>
<tr>
<td>[36]</td>
<td>BLINC, CoralReef, Naive Bayes, Naive Bayes Kernel Estimation, Bayesian Network, C4.5 Decision Tree, k-Nearest Neighbour, Neural Network, Support Vector Machine</td>
<td>Packet-Level, Flow-Level</td>
<td>Aggregate</td>
<td>Academic, Commercial</td>
<td>DPI</td>
<td>Comparison of Classification Accuracy</td>
</tr>
<tr>
<td>[41]</td>
<td>Support Vector Machine</td>
<td>Packet-Level, Flow-Level</td>
<td>Aggregate</td>
<td>Academic, Residential</td>
<td>Not Specified</td>
<td>Comparison of Precision, Recall and Classification Accuracy</td>
</tr>
<tr>
<td>[17]</td>
<td>Composite (J48 Decision Tree, k-Nearest Neighbour, Random Tree, Ripper, Neural Network, Naive Bayes, PortLoad, Port)</td>
<td>Packet-Level, Flow-Level</td>
<td>Many</td>
<td>Academic</td>
<td>DPI</td>
<td>Comparison of Classification Accuracy</td>
</tr>
<tr>
<td>[3]</td>
<td>Ripper, C4.5 Decision Tree</td>
<td>Packet-Level, Flow-Level</td>
<td>Many</td>
<td>Academic</td>
<td>DPI, Generated</td>
<td>Comparison of Detection Rate, False-Positive Rate</td>
</tr>
<tr>
<td>[7]</td>
<td>k-Means</td>
<td>Packet-Level</td>
<td>Many</td>
<td>Academic</td>
<td>Not Specified</td>
<td>Comparison of Classification Accuracy</td>
</tr>
<tr>
<td>[40]</td>
<td>C4.5 Decision Tree</td>
<td>Packet-Level</td>
<td>Aggregate</td>
<td>Academic</td>
<td>DPI</td>
<td>Comparison of Precision &amp; Recall</td>
</tr>
<tr>
<td>[60]</td>
<td>C4.5 Decision Tree</td>
<td>Flow-Level</td>
<td>Many</td>
<td>Academic</td>
<td>DPI</td>
<td>Comparison of Precision and Recall</td>
</tr>
<tr>
<td>[64]</td>
<td>Neural Network</td>
<td>Packet-Level, Flow-Level</td>
<td>Many</td>
<td>Academic</td>
<td>Not Specified</td>
<td>Comparison of Classification Accuracy</td>
</tr>
<tr>
<td>[54]</td>
<td>Naive Bayes Kernel Estimation, Bayesian Network, C4.5 Decision Tree</td>
<td>Flow-Level</td>
<td>Many</td>
<td>Commercial</td>
<td>DPI</td>
<td>Confusion Matrix</td>
</tr>
<tr>
<td>NA</td>
<td>Hidden Markov Model</td>
<td>Flow-Level, Connection-Level</td>
<td>Many</td>
<td>Residential</td>
<td>Generated</td>
<td>Hypothesis Tests of Aggregate $F$-score</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of surveyed literature and comparison to approach taken in this thesis.
Addressing the aforementioned shortcomings distinguishes this thesis from the state-of-the-art and as a consequence, represents novel contributions to the field. While the specifics regarding how these shortcomings are addressed are described in subsequent sections, the underlying theme of this study is the exploration of new frontiers in the field by a combination of building upon the state-of-the-art and addressing shortcomings with respect to experimental design and data analysis.

2.3 Conclusion

The Internet is in a constant state of evolution as new services, standards and technologies are created, adopted and deployed. The historically open and borderless nature of the system, fuels evolution driven by both lawful and unlawful parties with vested interests in maintaining the largely anonymous and neutral means in which the Internet operates. This type of evolution presents a challenge for Internet traffic research intent on identifying reliable distinguishing characteristics within Internet traffic traces by way of inventing technologies which inadvertently hinder the identification effort. As such, it is desirable to investigate both the degree to which proposed solutions can withstand such foreseeable architectural changes in addition to how effectively they may perform on present architectures.

Looking forward, although underway since 1994, the transition from IPv4 to IPv6 serves as another significant technological development which should serve as such a consideration, as it constitutes a significant architectural change with potentially significant consequences from an Internet traffic classification perspective.
This study attempts to address these concerns by investigating the influence the underlying network configuration exerts over the differentiation task in response to the findings in [54] which point to a possible link between poor generalization performance and classifier sensitivity to underlying network conditions. This is accomplished by analyzing the relationship between the differentiation task and characteristics of the underlying network infrastructure. The most relevant of which would be the Maximum Transmission Unit (MTU) employed for a given connection which, in addition to connection-specific variation due to the variety of different networks comprising the modern Internet (e.g., mobile, broadband, enterprise), is necessarily affected due to the shift from dial-up to broadband Internet access. Further still, such changes are expected in the future as a result of adoption of IPv6 and as such, insight gained in the present will continue to remain relevant in the immediate future.
Chapter 3

Preliminaries

3.1 Introduction

This chapter will discuss the importance of considering the underlying network configuration from which test/training samples are drawn, as well describing in more detail the aforementioned traffic features and classifier architecture which form a significant portion of this thesis’s contribution to the field. The structure of the 5-state HMM and associated operations will also be discussed in general and in reference to the aforementioned features. The utilization of HMMs as Internet traffic classifiers will also be discussed. This chapter will conclude with a topical introduction to linear modelling, Analysis of Variance and hypothesis testing, all of which are employed to analyze the experimental results presented in subsequent chapters.

3.2 Network Independent vs. Dependent Features and Poor Generalization Performance

The terms “Network Dependent” and “Network Independent” in the context of this thesis differentiate between features which are believed to be sensitive to the
underlying network configuration from those which are not. Sensitivity to changes in the underlying network configuration is an important property of a given feature as it has the potential to limit the features generalization capability, especially when the classifier architecture employed relies on some sort of machine-learning approach which utilizes a training phase and labelled training samples. When a network dependent feature is employed for classification purposes and the resulting classifier performs significantly worse on testing samples generated using a different network configuration than those used for training, the classifier is considered to exhibit poor generalization performance. In the machine-learning community this phenomenon is referred to as “over-fitting” [19] and it occurs when the underlying function approximation employed by the classifier fails to represent the general population and instead becomes reflective of specific observations used for training. Taken to the extreme, an over-fit function approximation can become just a functional representation of the training observations themselves, resulting in a restriction of the domain and range to that of the training observations. Another common result of poor generalization performance occurs when the underlying functional relationship between variables is confounded by the presence of an unknown variable. In this scenario, the presence of the unknown variable within the test sample is not representative of the population and as a consequence, the resulting approximation is similarly biased. When utilized by a classifier, poor generalization performance can be observed depending upon the degree of bias introduced by the confounded variable(s).

Network dependent features then are those which are potentially influenced by the presence of a specific network configuration and as a result, exhibit poor gen-
eralization performance when evaluated using samples drawn from different network configurations. Sensitivity in this context is simply the result of said bias manifesting itself as poor generalization performance in the resulting classifier. Packet size and packet inter-arrival time (two of three features examined in this thesis) are both examples of network dependent features as they can be influenced via changes in network configurations like available throughput or MTU. Network independent features are those which are hypothesized to be unaffected by the presence of specific network configurations during dataset generation and as such are thought to exhibit less sensitivity to the underlying network configuration. Examples of network independent features include TCP port numbers, application-specific payload signatures and message size [64]. Message size is the third feature assessed by this work.

3.3 Network Configuration As a Source of Bias

Though there are many configurable parameters that could introduce significant bias and ultimately lead to degraded generalization performance, this thesis focuses on the potential influence of a biased MTU on the viability of packet size and inter-arrival time traffic features. Employed by both IPv4 [1] and IPv6 [18], the MTU governs the maximum size of a TCP packet/UDP datagram that can be transmitted via the IP protocol. As a configurable parameter, the MTU can have a profound effect on transport layer features that incorporate packet size by effectively truncating the distribution. As a result, any classifier trained using said features would be susceptible to the bias introduced by the unaccounted for influence of the MTU.
size parameter. With respect to inter-arrival time, while there is a well understood theoretical relationship between packet size and inter-arrival time [35], the degree to which this relationship holds empirically in the context of Internet traffic classification research has yet to be determined. Independent of these considerations, there is still tremendous value in exploring the potential impact of MTU as a confounding variable as it will very likely yield additional insight into how existing classifier architectures would fare if MTU sizes greater than 1500 bytes were to become more common [58].

3.3.1 Relationships between MTU, Packet Size

According to RFC 1191 [50], in the majority of cases the minimum MTU employed by an IPv4 connection is 576 bytes, while RFC 1981 raises this minimum to 1280 bytes in IPv6 [48]. Although IPv4 supports a considerable range of MTU sizes, recent research suggests that upwards of 86% of web traffic achieves the maximum supported packet size of 1500 bytes (constrained only by the size of a traditional Ethernet frame [56]) [45]. This statistic underscores the point that while the vast majority of connections fully utilize the IPv4 MTU, a significant portion (14%) do not. Such a discrepancy would not be an issue for Internet traffic classification research if the distribution of MTU size was shown to be consistent across different networks, especially those employed as data sources for Internet traffic classification research. However, to date no investigation of this kind has been conducted and as such it is unclear how MTU size is distributed both within the population as well as the within the more popular datasets utilized for Internet traffic classification research.

While data regarding the distribution of MTU size across networks is not
available, observations regarding the distribution of packet size across networks is [14].
Unfortunately, it is impossible to attribute the differences in packet size distribution
to MTU size employed given the data available. To address this shortcoming, Figure
3.1 is a histogram of the maximum packet size per TCP flow, taken from the auto-
generated dataset employed in this thesis. The “Web” class of traffic was chosen to
align with the observations cited above regarding MTU and Internet traffic which
were observed with a client-side MTU of 1500 bytes. While still not conclusive, the
results seem to reinforce the notion that the MTU size of a given TCP flow can vary
significantly even when the MTU of one of the end-points is fixed to the maximum
the underlying IP protocol version can support.
Figure 3.1: Histogram of maximum packet size, “Web” traffic class.

Although not readily visible in Figure 3.1, despite an MTU of 1500 bytes being employed on the client side, the majority of TCP flows achieve a maximum packet size of only 1488 bytes. Table 3.1 contains a summary of the five most prevalent maximum packet sizes observed. Interestingly, the maximum packet size is achieved in only 8.5% of all flows which would seem to contradict the findings in [45]. However, for the purposes of this study, the observation of considerable variation with respect to the achieved maximum packet size should serve to substantiate further investigation
into the sensitivity of classifiers employing packet size as a feature to changes in MTU.

<table>
<thead>
<tr>
<th>Maximum Packet Size</th>
<th>Percentage of Flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1488</td>
<td>61.25%</td>
</tr>
<tr>
<td>1428</td>
<td>16.95%</td>
</tr>
<tr>
<td>1500</td>
<td>8.55%</td>
</tr>
<tr>
<td>1440</td>
<td>3.97%</td>
</tr>
<tr>
<td>1478</td>
<td>1.82%</td>
</tr>
</tbody>
</table>

Table 3.1: Five most prevalent maximum packet sizes observed per trace.

3.3.2 MTU and Packet Inter-arrival Time

As difficult as it is to try to infer a possible relationship between MTU and packet size based on the data presented thus far, it is even less clear how to do so between MTU and packet inter-arrival time. While Este et al. showed that both packet size and inter-arrival time features remained stable over a period of eight years [24], their research did not address whether the entropy ascribed to each feature persisted across different network configurations. Like the argument above regarding packet size, examination of the packet inter-arrival time statistics corresponding to the generated data employed in this study can shed some light on this relationship.
Figure 3.2: ECDF of packet inter-arrival time, small (576 bytes) and large (1500 bytes) MTU conditions.

Figure 3.2 is plot of the Empirical Cumulative Distribution Function (ECDF) for both MTU conditions studied. The plot is truncated after the third quartile of the observed inter-arrival time in attempt to better portray the more meaningful differences. Unlike in Figure 3.1 where only the “Web” traffic class was used, all six traffic types studied were utilized in order to generate this plot. It is clear that differences in packet inter-arrival time exist between the two MTU conditions, as seen
by the divergence of the red and black lines in the plot. Given the non-normal shape of the ECDF as well as the apparent shift between the two series, utilization of the Mann-Whitney-Wilcoxon Rank-Sum hypothesis test is an appropriate means of testing the hypothesis that there is a significant difference between the two distributions. With samples of size 10,000 taken from each series, after calculation and analysis of the relevant statistics, there is sufficient evidence to conclude there is a statistically significant difference of between 0.00353 and 0.00705ms (99 times out of 100), with \( p << 0.01 \).

The observations described within this section illustrate a clear relationship between MTU and both packet size and inter-arrival time. However, to what degree these observations are representative of the Internet in general is unknown. As such, though the real-world implications of these feature’s sensitivity to MTU variation are not explicitly addressed by this work, a greater understanding of what is possible in terms of classification performance degradation as well as how to overcome certain deficiencies was still achieved. In this context, the modulation of MTU for experimental purposes should not be regarded strictly as an attempt to mimic conditions present across the Internet, but rather as a means of amplifying the effects of MTU bias, the presence of which cannot be readily quantified and would best be served via independent study.
3.4 Traffic Traces

One significant differentiator between this thesis and other contemporary work is what constitutes a single traffic trace. The most widely utilized definition corresponds to a TCP socket anchored by two end points, which can be identified by their respective IP addresses and port numbers, as described in the respective TCP/IP packet headers. However, many applications and protocols utilize multiple socket connections in parallel and, as such, are not well-served by this definition. Further, some applications utilize a mixture of both TCP and UDP connections, so a failure to consider both types in tandem further compromises the representativeness of the trace. To remedy this, a unique definition of an application-specific trace is employed whereby a trace contains a record of all data transmitted between a host and client independent of socket and transport-level protocol. As an example, Table 3.2 describes the number of concurrent connections (TCP & UDP) per application for the dataset employed and provides evidence to support the claim that different traffic types employ different numbers of concurrent socket connections.

<table>
<thead>
<tr>
<th>Application</th>
<th>Number of Concurrent Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>92.1%</td>
</tr>
<tr>
<td>DNS</td>
<td>72.3%</td>
</tr>
<tr>
<td>Shoutcast</td>
<td>99.4%</td>
</tr>
<tr>
<td>VOIP</td>
<td>100%</td>
</tr>
<tr>
<td>Web</td>
<td>17.2%</td>
</tr>
<tr>
<td>YouTube</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 3.2: A summary of the maximum number of concurrent sockets utilized by traffic type.

Of particular interest are the “Web” and “YouTube” classes, which make
consistent use of multiple socket connections and hence stand to most greatly benefit from the inclusion of concurrent connections into the definition of a traffic trace.

3.4.1 Definition of A Traffic Trace

Traffic traces are logs of communicated data observed by a given network interface. They are generated using specific software and hardware tools that can be configured to record each packet that is processed by the network interface in sequence, complete with various header fields and user data. Table 3.3 is an example of a single trace presented in a human readable form where the various columns consist of data extracted from packet header fields. Each row in the table represents an individual observed packet and the five columns are defined as follows:

**TS** a timestamp in SECONDS.MICROSECONDS format.

**SRC_IP** A 4 byte field in the IP header, identifies the IPv4 IP Address from which the packet originated.

**DST_IP** A 4 byte field in the IP header, identifies the destination IPv4 IP address of the packet.

**SRC_PORT** A 2 byte field in the TCP/UDP header, identifies the originating port of the packet.

**DST_PORT** A 2 byte field in the TCP/UDP header, identifies the destination port of the packet.

**PKT_SIZE** The size of the entire packet, including all headers (in bytes).
Table 3.3: A human-readable representation of an Internet traffic trace.

<table>
<thead>
<tr>
<th>TS</th>
<th>SRC_IP</th>
<th>DST_IP</th>
<th>SRC_PORT</th>
<th>DST_PORT</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3409</td>
<td>192.168.1.105</td>
<td>173.203.58.10</td>
<td>37504</td>
<td>80</td>
<td>676</td>
</tr>
<tr>
<td>1290118861.3425</td>
<td>192.168.1.105</td>
<td>173.203.58.10</td>
<td>37505</td>
<td>80</td>
<td>684</td>
</tr>
<tr>
<td>1290118861.3490</td>
<td>192.168.1.105</td>
<td>173.203.58.10</td>
<td>37506</td>
<td>80</td>
<td>683</td>
</tr>
<tr>
<td>1290118861.3507</td>
<td>192.168.1.105</td>
<td>173.203.58.10</td>
<td>37507</td>
<td>80</td>
<td>682</td>
</tr>
<tr>
<td>1290118861.4386</td>
<td>173.203.58.10</td>
<td>192.168.1.105</td>
<td>80</td>
<td>37504</td>
<td>444</td>
</tr>
<tr>
<td>1290118861.4534</td>
<td>173.203.58.10</td>
<td>192.168.1.105</td>
<td>80</td>
<td>37505</td>
<td>1428</td>
</tr>
<tr>
<td>1290118861.4547</td>
<td>173.203.58.10</td>
<td>192.168.1.105</td>
<td>80</td>
<td>37505</td>
<td>536</td>
</tr>
<tr>
<td>1290118861.4621</td>
<td>173.203.58.10</td>
<td>192.168.1.105</td>
<td>80</td>
<td>37506</td>
<td>591</td>
</tr>
<tr>
<td>1290118861.4729</td>
<td>173.203.58.10</td>
<td>192.168.1.105</td>
<td>80</td>
<td>37507</td>
<td>877</td>
</tr>
</tbody>
</table>

For continuity, subsequent examples in this section will be based on this trace. Traffic traces must be further analyzed and processed in order to produce datasets amenable to the development of Internet traffic classifiers. The subsequent processing typically involves two stages. The first centres on the identification of packet sequences between two end points, while the second focuses on the extraction of relevant features from said sequences. Figure 3.3 is a graphical representation of the trace described in Table 3.3 which emphasizes the presence of multiple concurrent socket connections within the trace.
Traditionally, the first stage of analysis and processing aims to identify sequences of packets belonging to a single socket session. Sockets represent a logical connection between two end points of a network and possess both a definitive establishment and termination procedure facilitating their identification from raw traffic traces containing many concurrent socket connections. Sockets are utilized by applications to communicate over TCP/IP (or UDP/IP) and are a convenient context in which to frame the extraction of features as those extracted from packets belonging to the same socket which by extension reflect the same generating application. Packets belonging to the same socket can be identified via referencing fields of the respective packet headers as well as aspects of the underlying protocols. In the case

Figure 3.3: A graphical representation of the trace described in Table 3.3.
of TCP sockets, both the “3-way handshake” and “4-way handshake” define when a specific TCP socket connection is established and terminated respectively [1]. In the case of UDP traffic traces, since the UDP protocol is stateless, there are no establishment/termination procedures. TCP handshake procedures do not differ by application type as they are a transport layer procedure and as such are not expanded upon here above and beyond their utilization in marking the beginning and end of a socket connection.

Performing the traditional first stage of processing described above on Table 3.3 results in the generation of five new tables, each corresponding to a specific socket instance. Since each table corresponds to a unique socket, there is no need to include the SRC_IP, DST_IP and SRC_PORT, DST_PORT fields. Instead, a binary direction field is added which indicates whether the packet was observed travelling from the source to the destination (0) or vice-versa (1). As was the case with Table 3.3, the size of the packet is chosen as the only feature of interest, though it could just as easily have been the packet inter-arrival time, TCP header flags, or the presence of a given binary signature within the TCP payload portion of the packet or even the packet itself. In the context of previous work, the features extracted from these four traffic traces would constitute four individual training and/or testing sequences, as depicted in Tables 3.4 - 3.7

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3409</td>
<td>0</td>
<td>676</td>
</tr>
<tr>
<td>1290118861.4386</td>
<td>1</td>
<td>444</td>
</tr>
</tbody>
</table>

Table 3.4: The first trace instance, corresponding to communication between sockets 37504/80.
Aligning the lifetime of a traffic trace to that of a TCP/UDP socket is convenient given that in most cases traces are extracted from large repositories of continuously captured traffic where there is no contextual information regarding what a particular host/client pair are doing at a particular in point in time. Often DPI is employed as a substitute for this contextual information. Given that such is not the case in this study (in that the generating application for each unique socket connection is known \textit{a priori}), there is an opportunity to leverage such contextual information to more broadly define what constitutes a traffic trace to include all concurrent communications between a host and client. When all sockets involved in a given scenario are combined into a single trace, the resulting trace more closely resembles the original one in Table 3.3, with the addition of a column (SOCK) to

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3425</td>
<td>0</td>
<td>684</td>
</tr>
<tr>
<td>1290118861.4534</td>
<td>1</td>
<td>1428</td>
</tr>
<tr>
<td>1290118861.4547</td>
<td>1</td>
<td>536</td>
</tr>
</tbody>
</table>

Table 3.5: The second trace instance, corresponding to communication between sockets 37505/80.

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3490</td>
<td>0</td>
<td>683</td>
</tr>
<tr>
<td>1290118861.4621</td>
<td>1</td>
<td>591</td>
</tr>
</tbody>
</table>

Table 3.6: The second trace instance, corresponding to communication between sockets 37506/80.

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3507</td>
<td>0</td>
<td>682</td>
</tr>
<tr>
<td>1290118861.4729</td>
<td>1</td>
<td>877</td>
</tr>
</tbody>
</table>

Table 3.7: The second trace instance, corresponding to communication between sockets 37507/80.
differentiate between the respective sockets.

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>SOCK</th>
<th>PKT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3409</td>
<td>0</td>
<td>0</td>
<td>676</td>
</tr>
<tr>
<td>1290118861.3425</td>
<td>0</td>
<td>1</td>
<td>684</td>
</tr>
<tr>
<td>1290118861.3490</td>
<td>0</td>
<td>2</td>
<td>683</td>
</tr>
<tr>
<td>1290118861.3507</td>
<td>0</td>
<td>3</td>
<td>682</td>
</tr>
<tr>
<td>1290118861.4386</td>
<td>1</td>
<td>0</td>
<td>444</td>
</tr>
<tr>
<td>1290118861.4534</td>
<td>1</td>
<td>1</td>
<td>1428</td>
</tr>
<tr>
<td>1290118861.4547</td>
<td>1</td>
<td>1</td>
<td>536</td>
</tr>
<tr>
<td>1290118861.4621</td>
<td>1</td>
<td>2</td>
<td>591</td>
</tr>
<tr>
<td>1290118861.4729</td>
<td>1</td>
<td>3</td>
<td>877</td>
</tr>
</tbody>
</table>

Table 3.8: An example of a human-readable traffic trace consisting of multiple concurrent sockets.

Redefining what constitutes a single traffic trace to encompass multiple concurrent socket connections increases the entropy of the trace. Via combining concurrent sockets into a single trace, application-specific characteristics like the interrelated multi-socket signalling employed by the HTTP protocol can be leveraged as features for classification. As a relevant example to the trace described in Figure 3.3, the incorporation of multiple socket connections into a single trace reduces the variation of the first few packets observed during a typical web page request. This stabilization occurs because of the isolation of HTTP requests which typically retrieve mostly ASCII-encoded HTML code from subsequent requests which are more likely to request binary data such as images. When concurrent sockets are treated as independent traces, this distinction is lost as all socket connections are treated as individual traces and as a result the variance increases due to the bimodal nature of HTTP responses involving ASCII-encoded HTML code vs. those containing mostly binary image data. This unique definition of a traffic trace is an important distinction between the state-of-the-art and the contributions of this thesis to the field.
3.4.2 Feature Definitions

This section details how the targeted features of this study are computed. With respect to the network dependent features introduced previously, this section mainly serves to provide a more rigorous mathematical derivation. With respect to network independent features, this section will both introduce as well as derive how the features are computed from traffic traces.

3.4.2.1 Packet Size

Packet Size is defined as the number of network-layer bytes communicated by one end-point to another. This definition includes the IP, TCP/UDP headers and application-layer data and is typically bounded to a maximum of 1500 bytes, though it can be restricted further depending upon the MTU configuration of the underlying physical layer. The minimum packet size is 40 bytes in the case of TCP traffic and 28 bytes in the case of UDP traffic. With respect to the traffic trace described in Figure 3.3, each packet is carrying 40 bytes worth of TCP/IP header information and as such the total number of bytes utilized for communicating HTTP protocol data is 40 bytes less than what is stated. With respect to the traffic traces defined above, the “PKT_SIZE” column adheres to this definition and as such, a formal derivation is not necessary.
3.4.2.2 Inter-Arrival Time

Inter-arrival Time describes the amount of elapsed time between consecutive observations of packets. As such, for the first packet in a sequence of packets between a client/socket within the context of a given TCP/UDP socket, the Inter-arrival time is always 0. For all subsequent elements, the inter-arrival time is computed as the elapsed time between observation of the current packet and the immediately preceding one. The inter-arrival time tends to follow a Poisson distribution with a significant heavy-tail, with concentrations of values in the sub-millisecond range and extremes of 20-30 seconds in some cases. For the $i$th packet, where $i > 0$ with an associated time stamp $TS$ in a sequence, the packet inter-arrival time (IAT) is defined as,

$$IAT_i = TS_i - TS_{i-1}.$$  

(3.1)

3.4.2.3 Message Size

First introduced in [60], the message size (MSG_SIZE) feature aims to better reflect the nature of the information exchanged at the application layer than packet size, which is heavily influenced by and as a consequence most representative of the underlying transport layer. It is constructed by aggregating segmented packets together, thus more accurately representing the size of the original application layer packet data unit. With respect to the trace depicted in Table 3.8, utilization of the message size features entails combining consecutive packets in the same direction. More formally, with respect to the trace defined in Table 3.8, given a time stamp $\tau$, 

a direction $\delta$ and a socket $\sigma$ where $\tau$, $\delta$, $\sigma$ correspond to the first packet exchanged in specific direction $\delta$, via a given socket $\sigma$ beginning at a particular time $\tau$, the corresponding $MSG\_SIZE$ is defined as,

$$MSG\_SIZE(\tau,\delta,\sigma) = \sum_{DIR=\delta,TS=\tau,SOCK=\sigma}^{DIR\neq\delta} PKT\_SIZE(\tau,\delta,\sigma), \quad (3.2)$$

resulting in the trace described below in Table 3.9. Of particular interest is the removal of a row corresponding to $TS=1290118861.454678$ due to the aggregation of consecutive packets communicated via the same socket in the same direction. As such, traces containing $MSG\_SIZE$ as a feature possess the property of alternating directions of packet flow for any two consecutive messages communicated via the same socket.

<table>
<thead>
<tr>
<th>TS</th>
<th>DIR</th>
<th>SOCK</th>
<th>MSG_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290118861.3409</td>
<td>0</td>
<td>0</td>
<td>676</td>
</tr>
<tr>
<td>1290118861.3425</td>
<td>0</td>
<td>1</td>
<td>684</td>
</tr>
<tr>
<td>1290118861.3490</td>
<td>0</td>
<td>2</td>
<td>683</td>
</tr>
<tr>
<td>1290118861.3507</td>
<td>0</td>
<td>3</td>
<td>682</td>
</tr>
<tr>
<td>1290118861.4386</td>
<td>1</td>
<td>0</td>
<td>444</td>
</tr>
<tr>
<td>1290118861.4547</td>
<td>1</td>
<td>1</td>
<td>1964</td>
</tr>
<tr>
<td>1290118861.4621</td>
<td>1</td>
<td>2</td>
<td>591</td>
</tr>
<tr>
<td>1290118861.4729</td>
<td>1</td>
<td>3</td>
<td>877</td>
</tr>
</tbody>
</table>

Table 3.9: An example of a human-readable traffic trace employing the message size feature and multiple concurrent sockets.

### 3.4.2.4 Serialization

The serialization of traces containing multiple sockets is non-trivial for two reasons. First, for traffic classes which are largely unidirectional the process of aggregating consecutive packets into a single message may introduce exceedingly large
messages into the feature vector and as a result compromise the applicability of the feature to largely unidirectional traffic. These large messages occur because uninterrupted sequences of packets in a single direction are aggregated into a single message, which becomes impossible to differentiate from other unidirectional traffic types. To combat this, a maximum message size was introduced which forced the creation of a new message once a threshold of bytes inside a message was exceeded. In this implementation, the threshold was set to 5000 bytes.

Another possible problem with the computation of message size arises with respect to serializing multiple overlapping messages. It is possible that due to the arrival time of different packet segments across multiple concurrent sockets that the resulting messages overlap in time, thus making naïve serialization based on global packet arrival time impossible. To address this, each respective socket in a given trace is first processed into a sequence of MSG\_SIZE features and is then serialized into a single feature vector based upon the arrival of the first packet within each comprising message, independent of socket. Figure 3.4 provides an example of this serialization approach and contrasts it with the single socket approach employed by previously surveyed works.
Figure 3.4: A contrast of different Internet traffic serialization mechanisms.
3.5 Hidden Markov Models

Hidden Markov Models (HMMs) are stateful, statistical models of partially observable Markov chains. Markov chains are a type of stochastic process which uphold the Markov property which stipulates that each state be independent of each other in terms of their respective transition probabilities. More formally, given a Markov chain $\lambda$, the probability of being in any state at time $t$ (denoted $s_t$) is given by,

$$P(s_t = i | s_{t-1} = j, s_{t-2} = k, \ldots, \lambda) = P(s_t = i | s_{t-1} = j, \lambda).$$

The implication of the simplification is the efficient computation of process behaviour independent of the complexity of the chain itself. Partially observable Markov chains then are Markov chains wherein an additional, unobserved component is introduced. Typically, this “hidden” component manifests itself as a probability distribution at each state of the Markov chain. Under these conditions, Hidden Markov Models and their associated methods can be used to estimate the parameters of the hidden probability distributions. Such scenarios occur frequently in the natural sciences, in areas as diverse as speech recognition and bioinformatics wherein observable events (e.g., phonemes, DNA sequences) are thought to be generated by a partially observable, stateful stochastic process.

This thesis describes an approach utilizing HMMs as a means of modelling the statefulness of various types of Internet traffic under the assumption that such models can be used to construct effective classifiers. This section introduces more formal aspects of HMMs and how their various parameters are estimated given a
sequence of observations. Emphasis will be placed on a particular type of HMM structure where each state is connected to every other and the number of states is fixed at five. This HMM configuration was chosen as it aligns with findings of a recent work in the field which identified said structure to be optimal from an Internet traffic classification perspective [15].

The section is divided into three parts. The first is devoted to outlining the structural properties of the HMM, while the second describes how the parameters of the HMM are estimated. In addition, attention will be given to cultivating an appreciation for the relative strengths of the HMM approach in the context of modelling the statefulness of the protocols which comprise much of the Internet traffic studied. Developing such an understanding will facilitate comprehension of the final section, which details how HMMs can be used to construct effective Internet traffic classifiers.

3.5.1 Introduction

The HMMs employed by this study can be described via the 5-tuple:

\[ \{N, A_N, B_N, M, \pi_N \} \]

where,

\[ N \] is the number of states in the model. States are enumerated as \{1, 2 \cdots N\} and \( q_t \) represents being in a state at position \( t \) within a sequence of observations.

\[ A \] is the state transition probability distribution \( A = \{a_{ij}\} \) where,

\[ a_{ij} = P(q_{t+1} = j|q_t = i), 1 \leq i, j \leq N. \]  (3.4)
$M$ is the "per-state" dimension of the observation vector $O$. When $M > 1$ there are multiple observations per state and as a consequence multiple parameters to be estimated. Observations can be either discrete or continuous in nature.

$B$ is the joint probability, $B = \{b_{jk}(x)\}$ where,

$$b_{jk}(x) = P(o_k|q_t = j), 1 \leq k \leq M. \quad (3.5)$$

It is important to note that each of the $M$ observations can be described by different distributions, be they discrete or continuous in nature. If continuous in nature however, the distributions are assumed to be log-concave in nature [6], though methods do exist to accommodate others [44].

$\pi$ is the initial state probability distribution $\pi = \{\pi_i\}$ where,

$$\pi_i = P(q_1 = i), 1 \leq i \leq N. \quad (3.6)$$

Graphically, a HMM can be represented as a finite state machine with $N$ states interconnected via $\frac{N(N-1)}{2}$ arcs with an additional $N$ arcs originating and terminating at each node, as depicted in Figure 3.5. Each arc connecting two states is annotated with the corresponding conditional probability $a_{ij}$, while each enumerated state consists of both a joint observation probability function $b_{jk}(x)$ and initial state probability $\pi_i$. Like deterministic finite automata (DFA) which can recognize the class of regular languages [33], HMM’s can “recognize” (read: compute a probability
of acceptance) the class of probabilistic regular languages [20]. With the ability to estimate the parameters of a HMM, it is possible to use HMMs to recognize classes of languages purely via examining examples from said languages. When the examples are taken from Internet traffic traces, the resulting HMMs become capable of recognizing specific types of Internet traffic. When many of these models are combined, the result is an Internet-traffic classifier, as depicted in Figure 3.5.

Figure 3.5: A graphical representation of a fully connected 5-state Hidden Markov Model.

HMMs are well-suited to modelling Internet traffic because a significant degree of the data communicated over the Internet is stateful in nature and is likely to be fairly well described by models containing multiple states. Similarly, it is quite common for Internet traffic to possess stochastic properties due to the variable nature of the data being communicated, which aligns quite well with the probabilistic aspects
of HMMs described above. Finally, the flexibility of HMMs to be used in conjunction with different numbers and types of observations (discrete vs continuous) readily accommodates the vast array of features which can be utilized for traffic classification purposes. Despite these strengths however, HMMs do possess a weakness regarding the necessity to estimate both the model parameters and the topology. Ideally the structure of the HMM itself would be inferred from training samples in the same manner as the model parameters, however this task is computationally expensive [20]. Hence, this study relies on the findings of others to support the utilization of a fully connected topology consisting of five states, leaving only the model parameters to be estimated from the sampled Internet traffic traces.

3.5.2 Three Canonical HMM Problems

As specified in the previous section, it is not always immediately obvious how a HMM can be used to solve a given problem. This transition from abstract probabilistic graphical model to practical tool is distilled into three canonical problems concerning HMMs, the solutions to which provide a means of utilizing the abstract definitions above and facilitating the use of HMMs in different contexts, notably as components of Internet traffic classifiers.

Problem 1 Given a model $\lambda = (A, B, M, N, \pi)$ and a set of observation sequences $O$, efficiently compute $P(O|\lambda)$.

The ability to compute the probability of observing a given sequence is vital in the context of classifier architecture where multiple trained HMMs (one per
possible classification) are employed and the one which possesses the highest probability of observing a given sequence ultimately determines the resulting classification.

**Problem 2** Given a model \( \lambda = (A, B, M, N, \pi) \) and an observation sequence \( O \) of length \( T \), efficiently compute the most probable state transition sequence \( q = (q_1, q_2, \ldots, q_t) \).

This problem is of significant importance when HMMs are utilized for sequence alignment purposes [39], but is of little importance in a classification context and as such is not discussed any further.

**Problem 3** Given a model \( \lambda = (A, B, M, N, \pi) \), maximize \( P(O|\lambda) \) via re-estimation of \( A, B, \pi \).

This is by far the most difficult of the three problems to solve, but in so doing a HMM’s parameters can be adjusted to maximize the probability of observing a specific set of sequences. As a result, HMMs can be trained using specially crafted sets of sequences corresponding to specific classes of interest. The resulting set of trained HMMs then form the basis of HMM-based classifiers.

Solving Problem 3 is best approached after the solution to Problem 1 is understood. As such, the following two sections discuss the respective solutions for each problem in turn.
3.5.3 Computing \( P(O|\lambda) \)

Given a sequence of observations \( O \) of size \( T \) and a model \( \lambda = (A, B, M, N, \pi) \), the probability of observing a sequence \( O \) is

\[
P(O|\lambda) = \sum_{i=1}^{i<N} P(O|q_i, \lambda) P(q_i|\lambda),
\]

\[
\sum_{i=1}^{i<T} \prod_{k=1}^{M} b_{q_{ik}}(o_{1k}) a_{q_{1q_k}} \prod_{k=1}^{M} b_{q_{2k}}(o_{2k}) \cdots a_{q_{T-1}q_T} \prod_{k=1}^{M} b_{q_{Tk}}(a_{q_k}).
\]

(3.7)

Naïve calculation of the above statement requires on the order of \( 2T \cdot MN^T \) operations. A more sophisticated approach involving the use of dynamic programming is employed to compute \( P(O|\lambda) \) more efficiently. The forward and backward procedures (henceforth referred to as \( \alpha(x) \) and \( \beta(x) \) respectively) are expressed via the following recurrence relations

\[
\alpha_t(i) = P(o_1, o_2, \ldots, o_T, q_T = i|\lambda) -
\]

\[
\begin{cases}
\pi_i \prod_{k=1}^{M} b_{ik}(o_{tk}), & 1 \leq i \leq N, t = 1 \\
\sum_{i=1}^{N} \alpha_t(i) a_{ij} \prod_{k=1}^{M} b_{jk}(o_{tk}), & 2 \leq t \leq T - 1, 1 \leq j \leq N \\
\sum_{i=1}^{N} \alpha_t(i) \beta_t(i), & t = T, 1 \leq i \leq N
\end{cases}
\]

(3.8)
\[ \beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \lambda) = \]
\[
\begin{cases} 
1, & 1 \leq i \leq N, t = T \\
\sum_{j=1}^{N} a_{ij} \left[ \prod_{k=1}^{M} b_{jk}(o_{tk}) \right] \beta_t(j), & t = T - 1, T - 2, \ldots, 1, \\
1 \leq i \leq N
\end{cases}
\]  
(3.9)

and as a consequence require only \(NM^2T\) calculations.

### 3.5.4 Estimating HMM Parameters

In [6], Baum et al. introduce the “Baum-Welch” algorithm as a means of estimating the parameters of a discrete HMM. The algorithm is iterative in nature, refining the estimates of the HMM parameters \((\pi, A, B)\) until a local minimum is reached. The following is largely based on the derivations found in [57] and which interested parties are encouraged to refer to for greater detail. As a specialization of the Expectation-Maximization (EM) algorithm, the Baum-Welch algorithm seeks to optimize \(\lambda\) via the auxiliary function below with respect to \(\lambda' = (\pi', A', B')\), which Baum et al. proved satisfies either \(\lambda = \lambda'\) or \(P(O|\lambda) < P(O|\lambda')\):

\[
Q(\lambda', \lambda) = \sum_{q} P(O, q | \lambda') \log P(O, q | \lambda),
\]  
(3.10)
which is guaranteed to converge to a locally optimal solution given,

\[ Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(O|\lambda) \geq P(O|\lambda'). \]  (3.11)

Expressing \( P(O|\lambda) \) in terms of the HMM model parameters yields,

\[ P(O, q|\lambda) = \pi q_0 \prod_{t=1}^{T} a_{q_{t-1}q_t} \prod_{k=1}^{M} b_{q_tk}(o_{tk}), \]  (3.12)

which when expressed as a logarithm is,

\[ \log P(O, q|\lambda) = \log \pi q_0 + \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \log \sum_{t=1}^{T} \log \sum_{k=1}^{M} b_{q_tk}(o_{tk}). \]  (3.13)

Baum’s auxiliary function can then be expressed as,

\[ Q(\lambda', \lambda) = Q_\pi(\lambda', \pi) + \sum_{i=1}^{N} Q_{a_i}(\lambda', a_i) + \sum_{i=1}^{N} \sum_{k=1}^{M} Q_{b_{i,k}}(\lambda', b_{i,k}), \]  (3.14)

where \( \pi = [\pi_1, \pi_2, \cdots, \pi_N] \), \( a_i = [a_{i1}, a_{i2}, \cdots, a_{iN}] \) and \( b_{ik} \) is \( [b_{i1}, b_{i2}, \cdots, b_{iM}] \).

\[ Q_\pi(\lambda', \pi) = \sum_{i=1}^{N} P(O, q_0 = i|\lambda') \log \pi_i, \]

\[ Q_{a_i}(\lambda', a_i) = \sum_{j=1}^{N} \sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda') \log a_{ij}, \]  (3.15)

\[ Q_{b_{ik}}(\lambda', b_{ik}) = \sum_{t=1}^{T} P(O, q_t = i|\lambda') \log \sum_{k=1}^{M} b_{ikt}(o_{tk}). \]
The three terms can be optimized with respect to \( \lambda \) independently, subject to the constraints,

\[
\sum_{j=1}^{N} \pi_j = 1, \\
\sum_{j=1}^{N} a_{ij} = 1, \forall j.
\]

With respect to \( b_{ik}(o_{tk}) \), two different constraints need to be considered depending upon whether probability density is discrete or continuous in nature. For the discrete case,

\[
\sum_{k=1}^{M} b_{ik}(o_{tk}) = 1, \forall i,
\]

and the continuous case,

\[
\int_{-\infty}^{\infty} b_{ik}(o_{tk}) do_{tk} = 1, \forall i.
\]

Independent maximization of each of these auxiliary functions yields a new set of model parameters \( \bar{\lambda} = [\bar{\pi}, \bar{A}, \bar{B}_m] \) where,

\[
\bar{\pi}_i = \frac{P(O, q_0 = i|\lambda)}{P(O|\lambda)}, \\
\bar{a}_{ij} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda)}{\sum_{t=1}^{T} P(O, q_{t-1} = i|\lambda)}, \\
\bar{b}_{ik}(x) = \frac{\sum_{t=1}^{T} P(O, q_t = i|\lambda) P(O_{tk} = x)}{\sum_{t=1}^{T} P(O, q_t = i|\lambda)},
\]

where \( P(O_{tk} = x) \) is defined as either a discrete or continuous (log-concave) probability distribution.
With the utilization of \( \alpha, \beta \) defined in the previous section, the expressions above can be restated as,

\[
P(O, q_t = i | \lambda) = \alpha_t(i) \beta_t(i),
\]

\[
P(O | \lambda) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i),
\]

\[
P(O, q_{t-1} = i, q_t = j | \lambda) = \alpha_{t-1}(i) a_{ij} \sum_{k=1}^{M} b_{jk}(o_{tk}) \beta_t(j),
\]

yielding the parameter re-estimation formulae

\[
\bar{\pi}_i = \frac{\alpha_0(i) \beta_0(i)}{\sum_{j=1}^{N} \alpha_0(j) \beta_0(j)},
\]

\[
\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} \beta_t(j) \sum_{k=1}^{M} b_{jk}(o_{tk})}{\sum_{t=1}^{T} \alpha_{t-1}(i) \beta_{t-1}(i)},
\]

\[
\bar{b}_{ik}(x) = \frac{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i) b_{ik}(o_{tk})}{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i)}.
\]

### 3.5.5 Efficient Maximum Likelihood Estimation of the Gamma Distribution

The two parameter Gamma distribution, \( \Gamma(a, b) = \frac{x^{a-1} \exp(-x/b)}{\Gamma(a) b^a} \),

\[
\Gamma(a, b) = \frac{x^{a-1} \exp\left(-\frac{x}{b}\right)}{\Gamma(a) b^a},
\]

where \( \Gamma(a) = (n-1)! \) is used to model all continuous-valued features studied in this thesis. Like the Gaussian distribution it is both a member of the exponential family of distributions as well as possessing two parameters which reflect the shape and scale of
the distribution. Unlike the Gaussian distribution it is not symmetric and as a result, is more appropriate for describing skewed data (see Figure 3.6), which happens to be likely when the data to be modelled is related to network packets.

![2 Parameter Gamma Distribution](image)

Figure 3.6: Plots of various parameterizations of the 2-parameter Γ probability density function.

Unfortunately, no closed-form solution for the maximum likelihood estimate (MLE) of the shape parameter $a$ exists. In [49], Minka derives the following compu-
tationally attractive approximation where,

\[
\ell(\Gamma(a, b)) = (a - 1) \sum_{i=1}^{N} \log x_i - n\log\Gamma(a) - n\log b - \frac{1}{b} \sum_{i=1}^{N} x_i,
\]

\[
= n(a - 1)\log \bar{x} - n\log\Gamma(a) - n\log b - \frac{n\bar{x}}{b},
\]

the MLE of the scale parameter \( \hat{b} \) is \( \frac{\bar{x}}{a} \). Substituting into \( X \) yields

\[
\ell(\Gamma(a, b)) = n(a - 1)\log \bar{x} - n\log\Gamma(a) - n\log\bar{x} + n\log a - na. \tag{3.20}
\]

Using a generalized Newton approach as outlined in [49], an approximation of the MLE of \( a \) is computed via,

\[
\frac{1}{a_{new}} = \frac{1}{a} + \frac{\log \bar{x} - \log \bar{x} + \log a - \Phi(a)}{a^2(\frac{1}{a} - \Phi'(a))}, \tag{3.21}
\]

where \( \Phi(x) \) is the Digamma function and \( \Phi'(x) \) is it’s first-derivative, \( \frac{d}{dx}\Phi(x) \) otherwise known as the 2nd order Polygamma function.

The above update step converges quickly, often within four iterations as reported by Minka. In order to satisfy the requirement of log-concavity which requires that \( a > 1 \) without employing computationally expensive constrained optimization techniques, the starting point recommended in [49] was altered to ensure \( a > 1 \) as follows,

\[
\hat{a} = \max(\frac{0.5}{\log \bar{x} - \log x}, 1.1). \tag{3.22}
\]
Similarly, $a$ was evaluated during each iteration of the approximation procedure to ensure $a > 1$. In the event that the condition was violated, the result of the previous iteration was used as the parameter estimate.

### 3.5.6 HMMs as Internet Traffic Classifiers

The concepts introduced above regarding the structure and associated computations involving HMMs require only slight modification in order to be useful in the context of Internet traffic classification. The basic structure of the HMM itself needn’t change, however the re-estimation formulae for $a_{ij}$ and $b_{ik}$ need to be generalized to accommodate multiple observation sequences. This generalization enables the utilization of multiple traffic traces during training which in turn enables the generation of HMMs which are parameterized to recognize sequences of a given class with high probability. The modifications to accommodate a set of independent observation sequences $O^S$ are,

$$
\bar{\pi}_i = \frac{\alpha_0(i)\beta_0(i)}{\sum_{j=1}^{N} \alpha_0(j)\beta_0(j)},
$$

$$
\bar{a}_{ij} = \frac{\sum_{s=1}^{S} \frac{1}{P_s} \sum_{t=1}^{T_s-1} \alpha_t^s(i)a_{ij}\beta_{t+1}^s(j) \sum_{k=1}^{M} b_{jk}^s(o_{tk})}{\sum_{s=1}^{S} \frac{1}{P_s} \sum_{t=1}^{T_s-1} \alpha_t^s(i)\beta_{t+1}^s(i)},
$$

$$
\bar{b}_{ik}(x) = \frac{\sum_{s=1}^{S} \frac{1}{P_s} \sum_{t=1}^{T_s} \alpha_t^s(i)\beta_{tk}^s(i)b_{ik}(o_{tk})}{\sum_{s=1}^{S} \frac{1}{P_s} \sum_{t=1}^{T_s} \alpha_t^s(i)\beta_{tk}^s(i)},
$$

where $\frac{1}{P_s} = P(O^s|\lambda)$. With the ability to train a HMM with respect to a group of observation sequences, it is also possible to train multiple HMMs independently with different sets of observation sequences representing different classes of internet
traffic. These independent HMMs form integral parts of the Internet traffic classifier, which operates by classifying sequences based on which HMM returned the highest probability of observing the candidate sequence. As such, in its most simple form, the HMM-based Internet traffic classifier employed in this thesis will always produce a classification, even if the probability of observing the sequence is very low. Figure 3.7 depicts the architecture of this classifier.

![Diagram](image-url)

Figure 3.7: A HMM-based Internet-traffic classifier architecture.

The classifier consists of $N$ trained HMMs ($\lambda_1, ..., \lambda_N$), each corresponding to a specific class of traffic. When presented with a sequence $O$ of length $T$, the
probability of observing the sequence $P(O|\lambda)$ is calculated for each of the $N$ HMMs. The traffic type corresponding to the HMM with the highest probability of observing $O$ is chosen as the class.

### 3.6 Statistical Analysis

To arrive at sound conclusions regarding the validity of a stated hypothesis, an appropriate analysis of the experimental data needs to be performed. Linear models satisfy such requirements and provide a powerful and widely accepted means of analyzing data. Via parameterization of a linear model, all of the variables of interest can be represented and their relationships quantified. With a parameterized linear model in hand, reaching sound conclusions regarding various hypothesis can be accomplished via a related statistical analysis technique known as Hypothesis Testing. Application of both of these techniques is sufficient to draw sound conclusions regarding the effect of the variables of interest in this thesis.

#### 3.6.1 Univariate Linear Models

Univariate Linear Models provide a statistical means of quantifying the relationship between a set of independent variables and some dependent variable as a linear equation of the form,

$$ Y = \beta X + \epsilon, \quad (3.24) $$

where $Y$ is a vector containing the dependent variable observations, $X$ is a specifically structured matrix containing the independent variable observations, $\epsilon$ is a vector of
errors and \( \beta \) is vector of model parameters to be estimated via application of Ordinary Least Squares (OLS). As a statistical technique, the resulting inferences regarding the relationship between variables are only estimates and as such vary in terms of their inherent quality depending upon the representativeness of the sample data, the model structure and the underlying relationship between variables. The technique makes several assumptions which must be upheld in order for the approach to be properly applied. In the event where the data to be analyzed do not meet such assumptions (as is the case with the data analyzed in this thesis), transformations can often be applied to address discrepancies, though often at the cost of increasing the difficulty in interpreting the model parameters. If all assumptions are met, conclusions regarding the significance of the relationship between variables in the population can be reached using a technique called Hypothesis Testing based solely on examination of sample data. All of the above are relevant to the analysis of the experimental results within this thesis and are described in more detail below.

3.6.1.1 Terminology

Univariate linear models are often used in the context of prediction, whereby the goal is to predict the response in a dependent variable due to some change in the value of one or more independent variables. It follows then that independent and dependent variables are sometimes referred to as predictor and response variables respectively. Changes in the response variable due to changes in the predictor variables are referred to as effects and as the name implies, conceptually can be thought of as the /effect/ a given value of an independent variable imparts on the dependent
variable. When the independent variables are categorical in nature (as they are in this thesis), the various values that each variable can take are referred to as levels. The nature of the change in both the dependent and independent variables is assumed to be linear though related techniques can be employed if this is not the case. With respect to Equation (3.24), prior to application of OLS, $\beta$ and $\epsilon$ are unknown whereas $X$ and $Y$ are known. Since $\beta$ and $\epsilon$ are terms in the linear model which are estimated via OLS and not known a priori, they are referred to as model parameters and their estimated values as parameter estimates.

In addition to effects of one level of an independent variable on the dependent variable, the effect of the variable as a whole is often of interest and is referred to as a main effect. Further, if the effect of one level of an independent variable is suspected/found to be dependant on the level of another, the compound effect is referred to as an interaction. In the context of the parameter estimation procedure via OLS, interactions are to be considered model parameters (i.e., occupy columns within the $\beta$ vector) and their effect is estimated in the same way as the effects of a specific independent variable level.

The results of evaluation of a parameterized model with previously unobserved independent variable values are referred to as model predictions. The difference between a model’s prediction and an actual dependent variable observation is referred to as residual error. Again referencing Equation (3.24), after application of OLS the change in the dependent variable as a function of change in one or more of the independent variables is contained within the vector $b$ (the estimated values of the vector $\beta$) while the residual errors are found within $\epsilon$, which is typically omitted from the
equation as it can be computed directly from the other terms in the model.

The variable $X$ in (3.24) forms what is referred to as the “design matrix” and must adhere to a specific structure and possess certain properties in order for OLS to be applied. Independent of the contents of the matrix, each row must correspond to a single observation of $Y$ while each column must correspond to a column in $\beta$. Supposing $x$ and $y$ were the number of columns and rows of $X$ respectively, the dimensionality of $X$ must adhere to the rule $x \leq y$, otherwise the system of equations resulting from multiplication with $\beta$ is under-determined and cannot be solved using OLS. Given that all of the independent variables of interest in this thesis are categorical, the $X$ will consist entirely of binary values. Referred to as “dummy variables”, each binary variable corresponds to a specific combination of independent categorical variable levels. There are numerous ways to code categorical variables into a design matrix which largely differ in terms of the way in which the resulting estimates are interpreted [31]. For this analysis, “Treatment Coding” was employed which results in parameter estimates that are to be interpreted relative to a reference combination of categorical variable levels.

### 3.6.1.2 Ordinary Least Squares

By convention, when deriving the estimates of $\beta$, $b$ is used in place of $\beta$ as a means of differentiating the estimated values from the model parameters. This
results in the simplified equation,

\[ Y = bX, \]
\[ X'Xb = X'Y, \]
\[ (X'X)^{-1}(X'X)b = (X'X)^{-1}X'Y, \]
\[ b = (X'X)^{-1}X'Y. \]

After evaluation of (3.25), \( b \) contains the parameter estimates for each dummy variable, which corresponds to each column of the design matrix as well each combination of independent variable levels. The parameter estimates represent the amount of change observed in the dependent variable which can be attributed to the influence of a given independent variable level. As a consequence of the theoretical properties of the OLS procedure, \( b \) is guaranteed to be optimal with \( \epsilon \) containing the smallest possible residual values given a linear model with design matrix \( X \) and observation vector \( Y \).

### 3.6.1.3 Assumptions

Utilization of OLS in the type of experiment conducted in this study brings with it assumptions regarding the relationship between independent and dependent variables, the manner in which sample data was collected/generated as well as the nature of the sample data itself. More specifically, the following assumptions must be met in order for OLS to be applied:

- **Linearity**: A linear relationship exists between the independent and dependent
variables.

- Independence: The residuals are uncorrelated.

- Homoskedasticity: The residuals exhibit constant variance.

- Distribution: The residuals are Gaussian distributed.

With the exception of Independence assumption, the three remaining assumptions revolve around the nature of the residuals. It is important to note that since the residuals are a function of the design matrix \( X \) and the dependent variable \( Y \), specification of the assumptions with respect to \( \epsilon \) in effect places the same assumptions on them as well. Failure to satisfy these assumptions can lead to incorrect inferences and, as a result, unsubstantiated conclusions.

Formal tests exist [53] to evaluate each of the aforementioned assumptions, though typically said tests require a slightly less-restrictive set of assumptions to be upheld in order to be applicable. Ultimately, the decision of whether a given data set satisfies a set of assumptions rests with the researcher and their interpretation of the results of each test. As such, evidence regarding satisfaction of the OLS assumptions will be provided, but mainly in graphical form. Sometimes referred to as “diagnostic plots”, such visualizations are commonplace in statistical analysis and often provide more rapid insight into the underlying structure of data than a given hypothesis test targeted at a specific assumption. Though less rigorous, the plots do provide substantial evidence regarding satisfaction of the stated hypothesis in an easily communicated form.
In some cases, even when an assumption is violated it is possible to continue to apply OLS. This can be achieved by performing a transformation of the sample data or by way of introducing additional model parameters. As an example, in the event where the variation attributed to each of the independent variables is not homogeneous (and as a consequence the assumption of homoskedasticity is violated), a slightly modified form of OLS can be employed. Known as Weighted Least Squares, the technique involves an additional coefficient vector $W$ which contains weights to be applied to each observation as shown in (3.26)

$$Y = bXW.$$  \hspace{1cm} (3.26)

Ensuring $W$ contains weights which are inversely proportional to the variance of each independent variable in $X$ ensures that the residuals are homogeneously distributed and as such, do not violate the homoskedasticity assumption. There is however a trade-off which must be considered when utilizing such techniques as it is often the case that the resulting model parameters may become harder to interpret or a loss of statistical power of any subsequent Hypothesis Testing may result.

### 3.6.2 Hypothesis Testing

Hypothesis Testing is a technique which enables sound evaluation of properly formulated hypotheses regarding statistical entities like linear models or samples of observations drawn from a population. Such hypotheses are often formalizations of questions regarding the relationship between variables within a sample. In this
thesis, hypothesis testing is used to formally test hypotheses regarding the results of the experiment described in this chapter. Though not prevalent in the Internet traffic classification field, usage of Hypothesis Testing is quite common in other scientific disciplines and brings with it many advantages beyond less rigorous analytical techniques. First, it is a structured approach which is widely practised and understood. Second, it is a mathematically valid approach which though not impervious to abuse, does provide a framework from which to critique the validity of results and the evidence used to substantiate conclusions. Finally, in part as a consequence of the previous two points, reproduction of results and corresponding analysis by different sources can be achieved with reasonable effort.

Two distinct types of hypothesis tests are employed in this thesis, each targeted at answering different questions. The first and most general test is used to determine if the parameterized linear model itself is statistically significant, as there is little value in proceeding with more specific tests if the model poorly captures the characteristics of the data. The second and more versatile type of hypothesis test is used to directly assess the evidence supporting specific claims about the relationships between combinations of independent variables and the dependent variable. This is accomplished via expression of each hypothesis in terms of the model parameters which are then evaluated to determine if sufficient evidence exists to substantiate the claims.
3.6.3 Tests of Significance

Hypothesis testing is a technique which can be used to assess whether the probability of an observation or relationship involving potentially many parameters is significantly different than that which could be expected due to random variation alone. The probability of said observation or relationship occurring by chance can be used to substantiate conclusions regarding the statistical significance of the phenomena of interest. Such an approach is necessary due to the potentially large influence of the sampling process on the validity of estimates of population parameters computed via random sampling. The technique takes into consideration the inherent uncertainty that random sampling introduces into the estimation process by imposing restrictions on the nature of the hypothesis and samples. In addition to being random, the distribution of the estimated quantity of interest is assumed to be known, as is the size of the random sample.

The hypotheses are required to be expressed as expressions involving parameter estimates (or functions thereof) of interest computed from the random sample and can range in complexity from simple, one-sided hypotheses of the form $x < C$, (where $C$ is some constant) to complex involving many parameters simultaneously satisfying a given set of criteria. The representation of an informal claim regarding a parameter estimate typically takes the form of two complementary expressions, referred to as the “Null” and “Alternate” hypotheses, often referred to as $H_0$ and $H_A$ respectively. $H_0$ is an expression of the expectation of the parameter estimates when the stated claim is unsubstantiated, while $H_A$ is the opposite; an expression
of the expectation of the parameter estimate of interest when the claim is supported by statistical evidence. As an example, consider the task of substantiating the claim that the means of two independent Gaussian-distributed variables \( \mu_1, \mu_2 \) with equal variance and equal sample size are drawn from different populations with different means. \( H_0 \) and \( H_A \) would take the form

\[
H_0 : \mu_1 = \mu_2, \\
H_A : \mu_1 \neq \mu_2.
\]

(3.27)

Though there are two expressions, they are mutually exclusive and as such determining the validity of one is sufficient to arrive at a conclusion regarding the other. Determining the validity of any claim is achieved via computation of a probability of observing a given test statistic (which is a function of the parameter estimates under consideration) given the sample size, sampling and distribution assumptions. This probability is then used to either reject or fail to reject \( H_0 \), based upon comparison to an established threshold. This concept becomes clearer when the hypotheses above are restated in a slightly more arithmetic fashion:

\[
H_0 : \mu_1 - \mu_2 = 0, \\
H_A : \mu_1 - \mu_2 \neq 0.
\]

(3.28)

The restatement, above, is sufficient to support the next step in the hypothesis testing procedure as it clearly states the quantity of interest, the difference between the respective means of the two samples \( \mu_1, \mu_2 \) on the left-hand side and a constant (0)
on the right. If there is truly a difference in terms of the mean of the populations from which the samples were drawn, the difference in the sample means should $\neq 0$. Given the restrictions noted above regarding the nature of the samples and their distribution, it is possible to compute the probability distribution of the difference between the respective sample means via calculation of Student’s $t$-statistic. Equation (3.29) describes computation of the $t$ statistic corresponding to the hypotheses defined in (3.28) and enables the assignment of a probability to the observed difference between sample means.

$$t = \frac{\mu_1 - \mu_2}{s/\sqrt{n}},$$  \hspace{1cm} (3.29)$$

where $s$ is the standard deviation and $n$ is the sample size. It is $P(t_{\alpha,DF})$ which forms the basis of the test of significance via comparison to predefined thresholds which separate what is considered to be “statistically significant” from plausible due to chance alone. This threshold is often referred to as $\alpha$ as well as the “critical value”. While the value of this threshold is somewhat arbitrary, it is customary to see $\alpha$ equal to .05, .01, or .001. As an example, the critical value corresponding to a lower 1-sided test with $\alpha = 0.05$ and $DF=2$ can be found at $x = -2.920$ in Figure 3.8.
There is a relationship between $\alpha$ and the possibility of reaching the wrong conclusion. Mistakes can be made via incorrectly rejecting $H_0$ (Type I Error) as well as incorrectly failing to reject $H_0$ (Type II Error), otherwise referred to as a “false positive” and “false negative” error rates, respectively. In general, the larger the value of $\alpha$ the greater the Type I Error and the smaller the Type II Error. However, though $\alpha$ and Type I Error share a linear relationship, $\alpha$ and Type II Error do not. Finally, the smaller the value of $\alpha$, the stronger the evidence must be in order to
reject $H_0$, where strength of evidence can be achieved through a combination of effect size, sample size and hypothesis test power.

### 3.6.4 Testing Model Significance

Given the categorical nature of the data to be analyzed, it is convenient to interpret the linear model as a generalization of an Analysis of Variance (ANOVA) procedure [27]. When interpreted as an ANOVA design, hypothesis testing of the model itself and each predictor variable and their interactions can be performed in a consistent way. In the simplest cases, such hypothesis tests are strictly concerned with testing whether individual model parameters are statistically significant. $H_0$ and $H_A$ for these tests typically take the form,

\[
    H_0 : \beta_1 = 0, \quad H_A : \beta_1 \neq 0,
\]

and as such are intended to test whether $\beta_1$ is significantly different from 0, which would indicate whether the corresponding variable $X_1$ is a significant predictor of the response variable $Y$ in the linear model $Y = b_0 + b_1 X_0 \cdots b_N X_{N-1}$. In the context of the experimental design employed in this thesis (which is described in detail in Chapter 4), such a test might assess whether HMM-based Internet traffic classifiers trained with sequences of packet size features of length 10 under a large MTU network configuration differ significantly from those utilizing packet inter-arrival time features. Incidentally, performing hypothesis tests of this nature does not require utilization
of any ANOVA formalisms and can be achieved by calculating Student’s $t$ statistic alone. However, if a more complicated hypothesis is constructed that involves the simultaneous testing of multiple parameters, a different approach is required. Such a scenario arises in testing the multiple hypothesis outlined in Section 4.3.2, which do not exclusively correspond to comparisons between specific experimental conditions. Utilization of the ANOVA technique simplifies the hypothesis testing procedure by way of generalizing the problem to one which can be accommodated by a single framework wherein both the validity of the linear model itself and arbitrary combinations of it’s parameters can be readily tested. In addition, given that multiple hypothesis tests are to be carried out using the same sample, proliferation of Type I/Type II error can be controlled via application of Tukey’s Honest Significant Difference (HSD) test.

3.6.4.1 Analysis of Variance

Like univariate linear models, ANOVA aims to attribute the variation observed in a particular response variable to one or more predictor variables. As a consequence of this decomposition, the ANOVA procedure can be used to determine if there exists a significant difference in the mean response of any categorical factors. The technique can also be used to examine the validity of a linear model via partitioning and ascribing the variation in the response to arbitrary sets of predictor terms. In that scenario where the model terms are not strictly categorical, the output of the ANOVA procedure is not a test of the difference in means of some factorial experimental design but instead an assessment of the predictive capability
of the model terms. In this thesis, both uses are employed to first establish validity of
the model and second to assess the relative performance characteristics of the various
experimental conditions.

Usage of the ANOVA procedure is often accompanied via a tabular repre-
sentation of the decomposition of variance. Known as the ANOVA table, it provides
the necessary computations required to perform tests of significance regarding the
parameters of interest. Construction of the ANOVA table occurs after application of
OLS and as such is not utilized during the parameter estimation process. Rather,
it is simply a convenient way of summarizing the decomposition of variance within
the model. It is important to note that Table 3.10 is reflective of a table structure
which would be employed if one were interested in assessing the validity of a model
as a whole. If one were interested in examining the decomposition of variance to
the individual model terms (as is the case in this thesis), the “Model” row would be
replaced with a single row for each of the model terms. Independent of how these
parameters are represented in the table, the “Residual” and “Total” rows and their
associated values remain the same as the decomposition of the “Model” row does not
introduce any new parameters into the model as it is simply an aggregation of the
individual model terms.

<table>
<thead>
<tr>
<th>Source</th>
<th>Sums of Squares (SS)</th>
<th>Degrees of Freedom</th>
<th>Mean Square (MS)</th>
<th>F Statistic (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>$SS_R$</td>
<td>$P$</td>
<td>$MS_R$</td>
<td>$F(P+1, N-(P+1))$</td>
</tr>
<tr>
<td>Error</td>
<td>$SS_E$</td>
<td>$n - (P+1)$</td>
<td>$MSE_E$</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>$SS_T$</td>
<td>$n - 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.10: The Analysis of Variance (ANOVA) table structure and associated
calculations.
where subscript $E$, $R$ and $T$ refer to “Error”, “Residual” and “Total” respectively, $P$ refers to the number of model parameters (or categorical levels), $n$ the sample size and the remaining parameters computed as follows,

\[
SS_R = b'X'Y - n\bar{Y}^2,
\]

\[
SS_E = Y'Y - b'X'Y,
\]

\[
SS_T = Y'Y - n\bar{Y}^2,
\]

\[
MS_R = SS_R/p,
\]

\[
MS_E = SS_E/n - (p + 1),
\]

\[
F(P + 1, N - (P + 1)) = \frac{MS_R}{MS_E} \tilde{F}_{p,n-p-1}.
\]

The $F$ statistic [61] on the far-right is what is used to perform the hypothesis test. It is defined as the ratio between the mean-squared error within a group (in the factorial setting) or a set of predictor variables (in the linear model validity setting) to the global mean square error. Computation of the associated PDF is a function of both $P$ and $N$ and is depicted below in Figure 3.9.
Figure 3.9: A figure of the empirical $F$ distribution for various combinations of $P$ and $N$.

3.6.4.2 Tukey’s Honest Significant Difference Test

While examination of the $F$ statistic can be used to test the significance of the entire linear model and arbitrary combinations of it’s individual components, simultaneous comparison between pairs of components requires a different approach. Tukey’s HSD [34] is a posthoc test for simultaneous pairwise comparison of model parameters. As a post hoc test, it is typically performed after construction of a linear model and assessment of model validity. It can control for the exponential growth of
Type I/Type II error which arises due to performing multiple hypothesis tests using either the $t$ or $F$ statistics.

Like the utilization of univariate linear models, Tukey’s HSD assumes sample independence and homoskedasticity and as such can benefit from the utilization of weighted least squares when the later is violated. Tukey’s HSD compares the difference of model parameter estimates ($\Delta$) against the $q$ statistic, and concludes a significant difference exists if $|\Delta| > q$. The $q$ statistic can be expressed using the same quantities found in the ANOVA table as follows:

$$HSD = q \sqrt{\frac{MS_{R_i}}{n}},$$  \hspace{1cm} (3.31)

where $MS_{R_i}$ refers to the $MS_R$ of the $i$th model parameter (or in the ANOVA table decomposition). Via application of Tukey HSD’s it is possible to assess the significance of all $2^P - (P + 1)$ pairwise combinations of model parameters in a single step and as a result, assess the validity of many hypotheses simultaneously without incurring exponential growth of Type I/II error.

### 3.7 Conclusion

This chapter focused on communicating the requisite theoretical and conceptual understanding needed to fully grasp the material presented in subsequent chapters. Significant attention was given to precisely defining what specifically this
thesis was examining with respect to the underlying network configuration and feature set employed during classification of Internet traffic. Only the necessary theoretical aspects of HMMs were introduced and even then only two of three canonical problems were examined in any great detail. Similar focus was employed when describing the requisit underpinnings of the statistical analysis approach employed, with care being taken to ensure sufficient detail to facilitate an understanding of subsequent data presentations. This chapter should serve to support subsequent discussion of the methodology employed as well as the analysis of the experimental results in the chapters ahead.
Chapter 4

Methodology

4.1 Introduction

This chapter describes the experimental methodology employed to assess the generalization performance of network independent features in 5-state, fully-connected HMM-based Internet traffic classifiers. Details regarding the environmental conditions present during data set generation as well as the specific tools and methods used to generate the different classes of traffic will be presented. Description of the experimental design, including the principal variables of interest will also be discussed. Validation of assumptions regarding data distribution and Homoskedasticity will also be provided. This chapter builds upon Chapter 3 via showing how the more theoretical concepts introduced are applied in the form of a reproducible experimental design which satisfies all assumptions of the statistical analysis procedures used to analyze the resulting data in Chapter 5.
4.2 Environmental Setup

4.2.1 Network Topology

The network configuration depicted in Figure 4.1 was used to generate and capture all data analyzed in this thesis. A standard residential DSL connection with (5 Mbps download, 1 Mbps upload throughput) was employed to connect the local area network to the Internet. The network hosted two workstations which were used to either generate or capture sample data used for training and testing purposes. Each workstation was connected to the residential DSL connection via a series of Ethernet and WiFi connections delivering a maximum throughput of 54 Mbps.

Figure 4.1: Network topology employed during data set generation.

The majority of sample data was generated via customized software execut-
ing on “Workstation A” and is explained in more detail in Section 4.2.2. Two classes of traffic were not amenable to the automated approach described therein and as such had to be captured rather than generated. To facilitate this process, two additional pieces of hardware were employed: An Analog Telephone Adaptor (ATA) manufactured by Zoom Technologies (Model 5801) and an Ethernet hub manufactured by Compex (Model TPC1016C). The ATA was used to provide voice over IP (VOIP) service to the residence. The network traffic generated by the ATA during normal operation was captured via placement of the Ethernet hub between the adaptor and the Wireless Router (en route to the DSL modem). In so doing, all traffic destined for and generated by the adaptor was made available for capturing via any device connected to the hub. “Workstation B” served no other purpose other than to capture this traffic.

4.2.2 Sample Generation

With the exception of the VOIP class of traffic which was passively recorded from the local network, all sample traces were generated using automated means in the network environment described in Section 4.2.1. All traffic classes were generated under two different network configurations intended to simulate the effects of different network configurations on traffic similar to those presented in [54]. The result was a data set containing two instances of several different classes of application traffic which could be used to assess the relative classification performance of network independent vs. dependent features.
4.2.2.1 Tools

All data analyzed in this thesis were captured using the popular tcpdump tool [63]. Tcpdump is an application front-end which uses the libpcap packet capturing library. Without additional filtering, a network trace captured via tcpdump will contain the entire sequence of packet transmissions between a host and server. Since the goal of this thesis was to focus on network independent features, tcpdump was configured to record nothing below the transport layer. To filter the irrelevant information from the resultant traces, a modified version of the TcpFlow tool was employed [21]. TcpFlow’s original function was to facilitate analysis of an exchange between a client/server from the application’s layer perspective by stripping away all transport-layer features. This included transport-layer signalling like that which would be associated with connection hand-shaking, retransmission and congestion control scenarios handled by TCP [62]. Figure 4.2 provides an example of the types of traffic features retained by the TcpFlow tool.

Although most transport-layer specific features were removed from the sample traces by the modified TcpFlow tool, some features of interest (i.e., TCP packet size and port) were retained along with network dependent features like packet size and inter-arrival time. The modifications to TcpFlow were solely to support the retention of relevant traffic features as well as formatting the resulting output in a convenient way. By capturing and filtering the auto-generated traffic traces by Tcpdump and the modified TcpFlow tools, a data set containing only the features was produced.
4.2.3 Trace Generation Algorithm

With the exception of the VOIP and BitTorrent traffic classes, all sample traces were generated and captured using the same algorithm. Algorithm 1 was employed twice for each traffic class, with the underlying network interface being configured with a different MTU size (e.g., small, large). The algorithm was used to generate 1000 trace files under each MTU condition in libpcap format, containing a complete copy of the data exchanged between host/server during a “typical” use case for a given traffic class.

The principle considerations in determining whether a given traffic class was amenable to automated generation was the availability of both an accessible
**Algorithm 1:** GenerateSampleTraces() - The procedure used to generate traffic traces.

**Input:**
- \( Q \): An entity which can be queried via English tokens, returning hypertext links and durations
- \( A \): An executable application
- \( D \): An English language dictionary [65]

**Output:** 1000 libpcap trace files

```plaintext
while i < 1000 do
    word[] ← Sample(D, 3)
    results[], duration[] ← Query(Q, word[])
    if Length(results) < 10 then
        next i
    end
    url ← Sample(results, 1)
    filename ← url + "-captured-trace-" + i + ".libpcap"
    TcpdumpCaptureStart(url, filename)
    processID = ApplicationLaunch(A, url)
    Sleep(duration)
    TcpdumpCaptureStop(url, filename)
    if ApplicationCrashed(processID) then
        DeleteFile(filename)
    else
        TerminateApplication(processID)
        i ← i + 1
    end
end
```
repository of class-specific URLs as well as a sufficiently capable application which could be used to stimulate the intended type of traffic. As an example, in the case of the “Web” class, Google’s search engine was ultimately chosen as the URL repository as it references a vast number of URLs as well as providing a means of accessing the repository by way of keyword-based search queries. Mozilla Firefox was chosen as the stimulating application as it could be configured to accept specification of a URL via the command line. Table 4.1 summarizes the applications chosen to stimulate the different classes of traffic studied.

<table>
<thead>
<tr>
<th>Class</th>
<th>Generating Application</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS</td>
<td>nslookup</td>
<td>google.com</td>
</tr>
<tr>
<td>Shoutcast</td>
<td>VLC 1.1.3</td>
<td>shoutcast.com</td>
</tr>
<tr>
<td>YouTube</td>
<td>VLC 1.1.3</td>
<td>youtube.com</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>BitTorrent Client</td>
<td>slackware.com</td>
</tr>
<tr>
<td>VOIP</td>
<td>Zoom 5801 ATA Adapter</td>
<td>link2voip.com</td>
</tr>
<tr>
<td>Web</td>
<td>Firefox 3.6.3</td>
<td>google.com</td>
</tr>
</tbody>
</table>

Table 4.1: Applications chosen to simulate different classes of traffic and the respective sources.

4.2.4 VOIP and Bittorrent Dataset Generation Procedure

The VOIP dataset was not generated, but rather accumulated over a period of time via passive monitoring of an active VOIP account. As described in Figure 4.1, the IP-traffic generated by an ATA serving a residential location was captured by rebroadcasting all packets addressed to or emanating from the adaptor to another interface running tcpdump using a traditional ethernet hub. This technique was employed due to the perceived difficulty in generating VOIP traffic and the availability of access to the adaptor. Only the characteristics of each packet (e.g., packet size,
Inter-arrival time) were recorded and utilized in generating features for classification.

BitTorrent traffic was generated manually via initiating a download of a then-recent Slackware Linux distribution under both network configurations and capturing all traffic generated until 20% of the target file had been acquired. This path was chosen, in part, because of the vast number of connections established during a typical BitTorrent file-sharing session and the difficulty in determining a priori which hosts would be contacted. Such a requirement could only be fulfilled via examination of the torrent tracker file itself, which was deemed too invasive.

4.3 Experimental Design

Generally speaking, the experiment described in this section was designed to determine what, if any advantage network independent features offer over network dependent features when used in conjunction with fully-connected 5 state HMM-based Internet traffic classifiers. Though easily understood as stated, the above goal needs to be operationalized into a scientifically valid, directly testable hypothesis with minimal ambiguity regarding measured quantities and outcome criteria if a valid experiment is to be devised to test it. To this end, the following hypotheses are to be interpreted in the context of fully connected, 5 state HMM-based classifiers with performance measured via aggregate $F$-score as described in 4.3.1.

1. Network independent features can be used to classify Internet traffic

2. There is a significant change in classification accuracy when sequences of lengths greater than 5 are used in conjunction with network independent features
3. Classifiers using network independent features perform better than those using network dependent features when the training and test set differ in terms of network configuration.

4. There is a difference in resulting aggregate F-score when network independent features are used and the sequence length employed is increased from 5 to 10 or 15.

5. When the distribution of MTU size within the test/training set is homogeneous, there is a difference in resulting aggregate F-score between network independent vs dependent feature sets.

6. The aggregate F-score of classifiers utilizing network independent features is significantly different from those utilizing network dependent features when the concentration of network configurations within the test/training set is mixed.

7. Classifiers using network independent features generate significantly different aggregate F-scores than those using network dependent features when the training and test set are generated using completely independent network configurations.

Given that the focus of each hypothesis is the same (i.e., relative classifier performance or a function thereof) as well as the controlled nature of all other parameters (e.g., sample traffic trace length, features employed, network configuration etc.), a statistically-inspired experimental design and accompanying analysis was found to be most appropriate. In this context, the term difference in the aforementioned hy-
hypothesis is recast as a requirement of statistical significance which must be satisfied for a conclusion regarding any of the stated hypotheses to be reached. Satisfaction of this requirement is achieved by way of formal hypothesis tests, which are widely employed across many scientific disciplines where similar inferences are sought (e.g., effectiveness of medication on treating a particular disease, differences in income between demographic groups, etc.).

4.3.1 Measuring Classifier Performance

There are a multitude of ways in which the performance of a given classifier can be measured [25]. Typically the exact metric employed is chosen depending upon which aspects of classifier performance are of particular interest in conjunction with the nature of the classification problem at hand. In this case, assessing generalization performance in the context of a classification problem involving multiple classes was of primary interest. For simplicity, no preference or consideration is given to (mis)classification of any specific class; all classes are considered of equivalent importance and as such the misclassification of any class for another should be reflected in the performance metric equally. Given that statistical procedures were to be employed in the analysis of the resulting performance data it was desirable for the chosen metric to have attractive statistical properties. After consideration of these criteria, $F$-score paired with a $k$-fold stratified cross-validation procedure as proposed in [26] was employed.
4.3.1.1  $k$-fold Stratified Cross-Validation

Before describing the $F$-score, it is important to understand the underlying process which it is measuring. $k$-fold, stratified cross-validation is a specialization of the general cross-validation method, which is used to assess the performance of a classifier given a set of labelled data. More formally, $k$-fold stratified cross-validation partitions the set of all samples, $S$ into $k$ equally sized, disjoint subsets or “folds” and assigns one of the subsets to a training set $T_{\text{test}}$ and the rest to $T_{\text{train}}$. Stratified cross-validation imposes the constraint that the distribution of classes $C$ within each $k$-th subset of $S$ be equal. $T_{\text{train}}$ is then used to train a given classifier which is evaluated via testing with $T_{\text{test}}$. As such, each iteration of $k$-fold cross-validation yields $k$ sets of classifier performance metrics as depicted in Figure 4.3.
4.3.1.2 $F$-score with 10-fold Stratified Cross-Validation

The $F$-score is a composite measure comprising of two components: precision and recall. Precision is the proportion of classifications assigned to a given class that actually belong to that class, whereas recall is the proportion of classifications assigned to a given class in general. In the context of a typical confusion matrix,

$$P = \frac{TP}{TP + FP},$$  \hspace{1cm} (4.1)
\[ R = \frac{TP}{TP + FN}, \]  

which, when expressed as a function of \( P \) and \( R \), the \( F \)-score is defined as,

\[ F = 2 \cdot \frac{P \cdot R}{P + R}. \]  

As noted above, application of \( k \)-fold cross-validation results in the generation of \( k \) performance metrics. Though it is possible and arguably more intuitive to compute the \( F \)-score for each of the folds and then average each score into a single scalar for the entire trial, such a calculation results in a \( F \)-score with unattractive statistical properties, chiefly among them considerable negative bias [26]. Realizing that the \( F \)-score as presented above is actually the harmonic mean of precision and recall, the following alternative computation yields a substantially reduced, though still slightly negatively biased estimate of aggregate \( F \)-score:

\[ F = \frac{2 \cdot \sum TP}{2 \cdot \sum TP + \sum FP + \sum FN}. \]  

To reinforce the utilization of this specific type of \( F \)-score and distinguish it from other varieties, henceforth it will be referred to as the aggregate \( F \)-score.

4.3.2 Experiment Structure

The experiment consists of three independent variables and one dependent variable. The independent variables are parameters which are hypothesized to influence the dependent variable. All independent variables in this experiment are
categorical in nature and as such can take a finite number of values, whereas the dependent variable is a strictly positive real number between $[0, 1]$. The three independent variables correspond to the maximum length of traffic sequence (Sequence Length) and the specific trace features employed during test/training (Feature) as well as the network configuration (read: MTU) present during generation of the train/test samples (Method). The dependent variable is the aggregate $F$-score, computed as described earlier via a 10-fold stratified cross-validation procedure. The three categorical independent variables combine to form a total of 45 different experimental states resulting in a total result set of 450 aggregate $F$-scores. The notation $Condition=Level$ will be used throughout the remainder of the thesis to refer to specific values of the independent variables. The specific conditions and their abbreviations are summarized in Table 4.2.
<table>
<thead>
<tr>
<th>Method</th>
<th>Seq. Length</th>
<th>Feature</th>
<th>Packet Size + IAT</th>
<th>Network Independent</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Small MTU</td>
<td>Train</td>
<td>Packet Size</td>
<td>μ(S,PS,5)</td>
<td>μ(S,P,15)</td>
<td>1-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>μ(S,PI,5)</td>
<td>μ(S,NI,5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(S,NI,5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>μ(S,PS,10)</td>
<td>μ(L,PS,5)</td>
<td>μ(L,P,15)</td>
<td>4-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(L,P,5)</td>
<td>μ(L,PI,15)</td>
<td>μ(L,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(L,NI,5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>μ(S,PS,15)</td>
<td>μ(L,PS,15)</td>
<td>μ(L,P,15)</td>
<td>7-9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(L,P,15)</td>
<td>μ(L,PI,15)</td>
<td>μ(L,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(L,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large MTU</td>
<td>Train</td>
<td>Packet Size</td>
<td>μ(SL,PS,5)</td>
<td>μ(SL,P,15)</td>
<td>10-12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>μ(SL,PI,5)</td>
<td>μ(SL,NI,5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(SL,NI,5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>μ(SL,PS,10)</td>
<td>μ(SL,P,10)</td>
<td>μ(SL,NI,10)</td>
<td>13-15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(SL,P,10)</td>
<td>μ(SL,PI,15)</td>
<td>μ(SL,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(SL,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>μ(SL,PS,15)</td>
<td>μ(SL,P,15)</td>
<td>μ(SL,PI,15)</td>
<td>16-18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(SL,P,15)</td>
<td>μ(SL,PI,15)</td>
<td>μ(SL,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(SL,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large MTU</td>
<td>Train</td>
<td>Packet Size</td>
<td>μ(LS,PS,5)</td>
<td>μ(LS,P,15)</td>
<td>19-21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>μ(LS,PI,5)</td>
<td>μ(LS,NI,5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(LS,NI,5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>μ(LS,PS,10)</td>
<td>μ(LS,P,10)</td>
<td>μ(LS,NI,10)</td>
<td>22-24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(LS,P,10)</td>
<td>μ(LS,PI,15)</td>
<td>μ(LS,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(LS,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>μ(LS,PS,15)</td>
<td>μ(LS,P,15)</td>
<td>μ(LS,PI,15)</td>
<td>25-27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(LS,P,15)</td>
<td>μ(LS,PI,15)</td>
<td>μ(LS,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(LS,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined</td>
<td>Train</td>
<td>Packet Size</td>
<td>μ(C,PS,5)</td>
<td>μ(C,P,15)</td>
<td>37-39</td>
</tr>
<tr>
<td>(Small + Large MTU)</td>
<td></td>
<td></td>
<td>μ(C,PI,5)</td>
<td>μ(C,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(C,NI,5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>μ(C,PS,10)</td>
<td>μ(C,P,10)</td>
<td>μ(C,NI,15)</td>
<td>40-42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(C,P,10)</td>
<td>μ(C,PI,15)</td>
<td>μ(C,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(C,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>μ(C,PS,15)</td>
<td>μ(C,P,15)</td>
<td>μ(C,NI,15)</td>
<td>43-45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(C,P,15)</td>
<td>μ(C,PI,15)</td>
<td>μ(C,NI,15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ(C,NI,15)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Summary of all 45 experimental conditions.

Each cell in Table 4.2 constitutes an independent result set of 100 aggregate F-scores. The number of trials was chosen to be sufficiently large to support
application of the statistical analysis techniques introduced above while also being computationally feasible given the resources at hand. In terms of minimizing the correlation between subsequent trials and thus upholding the spirit of each sample being independent (a prerequisite of the statistical analysis procedures employed), the composition of the stratified $k$-fold set was randomized from a master set of 1000 sample traces as were the initial weights of the various HMM parameters. Thus, in total $45 \times 100 = 4500$, 10-fold, stratified cross-validated classification trials were performed, requiring the training of $6 \times 45 \times 100 = 27000$ HMMs and testing of 4500 HMM-based classifiers. After all experimental states had been evaluated, the resulting 450 aggregate $F$-scores needed to be analyzed in order to determine if there was sufficient evidence to accept or reject each of the aforementioned hypotheses. As an improvement over the approach employed by many of the works cited in Chapter 2 involving direct numerical comparison of a statistic (e.g., $\mu$) of some performance measure (e.g., $F$-score), formal hypothesis tests were employed in order to arrive at substantiated conclusions.

### 4.4 Validation of Assumptions

Employing linear models fitted via OLS as a means analyzing the resulting performance metrics from the experimental design introduced in the previous section is dependent on the satisfaction of four criteria. Two of these criteria, Independence and Linearity can be assessed without inspection of the parameterized model as they are assumptions which are more concerned with the applicability of the experimental
methodology in a general sense than the specific numeric properties of the resulting parameter estimates. The remaining two require consideration of the resulting model itself and as such are dealt with separately.

4.4.1 Linearity and Independence Assumptions

The Linearity assumption assumes that a linear relationship exists between the independent and dependent variables. In the case of this design where the independent variables are all categorical in nature, there is no opportunity for this assumption to be violated as the model parameters that are estimated via OLS are simply contrasts from a reference. These contrasts can either be positive, negative or zero and reflect the difference observed in the dependent variable between different combinations of levels of the independent variables. The assumption of Independence is concerned with ensuring that observations share no inherent correlational structure that may bias the analysis. This assumption is typically upheld via ensuring that random sampling is employed. In the case of this experiment, two sources of random sampling contribute to satisfaction of the Independence assumption. The first is concerning the random initialization of HMM model parameters and the second is the randomization applied to the construction of the stratified 10-fold cross-validation test and training sets. These two sources of error ensure that consecutive observations within a given category are uncorrelated and as a result, ensure the Independence assumption is not violated.
4.4.2 Distribution Assumption

The Distribution assumption is concerned with the statistical distribution of the residuals within a linear model. This assumption is necessary as the optimality of the OLS estimated parameters can only be guaranteed when the resulting residuals are Gaussian distributed. As was the case with the Linearity and Independence assumptions, the strictly categorical nature of the independent variables employed affects how this assumption is validated. Given that there are no continuous independent variables within the model, the residuals of each model term are guaranteed to be Gaussian distributed if the observations themselves are Gaussian in nature. Thus, to confirm that this assumption is met one must only verify that a standardized plot of the observations themselves is Gaussian distributed, as is the case in Figure 4.4.
4.4.2.1 Homoskedasticity

Like the Distribution assumption, the Homoskedasticity assumption is also primarily concerned with properties of the residuals resulting from the estimation of the parameters of a linear model via OLS. It follows then that the same approach regarding examining the observations themselves in lieu of the residuals is sufficient to visually confirm the assumptions have been met. Figure 4.5 is a plot of all 45
experimental conditions, where each condition is summarized via a box and whisker figure. The box and whisker figures represent both the centre of the observations as well as the variation. The bold line near the centre of the box corresponds to the median of the observations for a given condition, whereas the box itself represents the 25th and 75th quantiles. The whiskers extend to +/- 1.5 the interquartile range, thus reflecting the degree of dispersion/variation. As is evident from inspection of the plot, the box and whisker figures are not uniformly shaped across all conditions. More specifically, considerable variation can be seen when comparing the interquartile range between groups. This is evidence that the variance across all conditions is not homogeneous and as such, violates the Homoskedasticity assumption.
Fortunately, violation of the Homoskedasticity assumption can be overcome via employing WOLS, where the weights of each experimental condition are set inversely proportional to the variance of the observations for a given condition. In so doing, the optimality of the resulting parameter estimates is preserved. This procedure is particularly suited to this scenario as it requires that the nature of the Homoskedasticity be described. Given the strictly categorical nature of the data to be analyzed, the nature of the Homoskedasticity could be described as being equal
to the estimated variance within each experiment condition. When this relationship is taken into account, the residuals appear homogeneous in nature across all experimental conditions as seen in Figure 4.6.

Figure 4.6: Aggregate $F$-score vs. Factor Level with weighting.

4.5 Validity of Linear Model

The table below describes the 45 WOLS estimated model parameters as well as their respective significance values. The reference level $b_0$ corresponds to
Method = Large MTU size, Sequence Length = 5 packets and Feature Set = Network Independent. The last three rows contain a summary of the significance of the model as a whole.
<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>0.8375</td>
<td>0.0015</td>
<td>546.6510</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,N1,5}$</td>
<td>0.0003</td>
<td>0.0021</td>
<td>0.1560</td>
<td>0.8758</td>
</tr>
<tr>
<td>$b_{SL,N1,5}$</td>
<td>-0.1175</td>
<td>0.0027</td>
<td>-43.3500</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{C,N1,5}$</td>
<td>-0.0228</td>
<td>0.0022</td>
<td>-10.5490</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{L,N1,10}$</td>
<td>0.0462</td>
<td>0.0020</td>
<td>23.0640</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{L,N1,15}$</td>
<td>0.0813</td>
<td>0.0020</td>
<td>40.6960</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{L,PS,5}$</td>
<td>-0.0959</td>
<td>0.0026</td>
<td>-37.4740</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{L,PI,5}$</td>
<td>-0.1147</td>
<td>0.0035</td>
<td>-33.2330</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,N1,10}$</td>
<td>-0.0021</td>
<td>0.0027</td>
<td>-0.7730</td>
<td>0.4395</td>
</tr>
<tr>
<td>$b_{LS,N1,15}$</td>
<td>-0.0326</td>
<td>0.0035</td>
<td>-9.2140</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{C,N1,10}$</td>
<td>-0.0125</td>
<td>0.0026</td>
<td>-4.7710</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,N1,10}$</td>
<td>-0.0228</td>
<td>0.0035</td>
<td>-20.2260</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,NI,10}$</td>
<td>-0.0075</td>
<td>0.0050</td>
<td>-1.5180</td>
<td>0.1291</td>
</tr>
<tr>
<td>$b_{LS,NI,15}$</td>
<td>0.0510</td>
<td>0.0047</td>
<td>10.7800</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{C,NI,15}$</td>
<td>0.0324</td>
<td>0.0043</td>
<td>7.4690</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,NI,15}$</td>
<td>0.0807</td>
<td>0.0043</td>
<td>18.6890</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,PS,10}$</td>
<td>-0.0073</td>
<td>0.0035</td>
<td>-2.0790</td>
<td>0.0376</td>
</tr>
<tr>
<td>$b_{LS,PS,10}$</td>
<td>-0.0249</td>
<td>0.0036</td>
<td>-6.8850</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,PS,10}$</td>
<td>-0.0127</td>
<td>0.0041</td>
<td>-3.0600</td>
<td>0.0022</td>
</tr>
<tr>
<td>$b_{PS,10}$</td>
<td>-0.0284</td>
<td>0.0042</td>
<td>-6.8030</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,PS,15}$</td>
<td>0.0013</td>
<td>0.0052</td>
<td>0.2490</td>
<td>0.8034</td>
</tr>
<tr>
<td>$b_{LS,PS,10}$</td>
<td>-0.0263</td>
<td>0.0048</td>
<td>-5.5200</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{C,PS,10}$</td>
<td>0.0711</td>
<td>0.0044</td>
<td>16.1100</td>
<td>$&lt;&lt; 0.010$</td>
</tr>
<tr>
<td>$b_{S,PS,10}$</td>
<td>0.0363</td>
<td>0.0046</td>
<td>7.8190</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,PS,15}$</td>
<td>0.0029</td>
<td>0.0052</td>
<td>0.5570</td>
<td>0.5774</td>
</tr>
<tr>
<td>$b_{LS,PS,15}$</td>
<td>-0.0042</td>
<td>0.0048</td>
<td>-0.8810</td>
<td>0.3784</td>
</tr>
<tr>
<td>$b_{C,PS,15}$</td>
<td>0.0849</td>
<td>0.0045</td>
<td>19.0070</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,PS,15}$</td>
<td>0.0548</td>
<td>0.0046</td>
<td>11.7980</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,PI,10}$</td>
<td>0.0084</td>
<td>0.0060</td>
<td>1.4040</td>
<td>0.1602</td>
</tr>
<tr>
<td>$b_{LS,PI,15}$</td>
<td>0.0089</td>
<td>0.0058</td>
<td>1.5410</td>
<td>0.1233</td>
</tr>
<tr>
<td>$b_{C,PI,15}$</td>
<td>0.0564</td>
<td>0.0052</td>
<td>10.7840</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,PI,10}$</td>
<td>0.0692</td>
<td>0.0052</td>
<td>13.3630</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{LS,PI,15}$</td>
<td>0.0058</td>
<td>0.0066</td>
<td>0.9660</td>
<td>0.3339</td>
</tr>
<tr>
<td>$b_{SL,PI,15}$</td>
<td>0.0272</td>
<td>0.0057</td>
<td>4.8020</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{C,PI,15}$</td>
<td>0.0847</td>
<td>0.0053</td>
<td>16.0760</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$b_{S,PI,15}$</td>
<td>0.0747</td>
<td>0.0052</td>
<td>14.2320</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
</tbody>
</table>

Multiple $R^2$: 0.9714
Adj. $R^2$: 0.9711

$F$ statistic: 3442 on 44 and 4455 DF
$p$-value: $<< 0.01$

Table 4.3: Univariate linear model incorporating all 45 experimental conditions.
An adjusted $R^2$ of 0.9714 and $F(44,4455)=3442$ and $p << 0.001$ indicates the model is a good fit to the data, with little unexplained variation. Construction of an ANOVA table (see Table 4.4) reveals that indeed each main effect and their associated interactions are statistically significant, indicating that there exists evidence that changes in any of Method, Sequence Length or Feature can result in statistically significant changes in aggregate $F$-score at $\alpha=0.05$. The significance of the interaction terms indicates that significant differences exist within specific combinations of the interacting variables in addition to any significant main effects attributed to the non-interacting terms of the model.

<table>
<thead>
<tr>
<th>Main Effect</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>$F$ statistic</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>4</td>
<td>139707</td>
<td>34927</td>
<td>34927.7860</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Sequence Length</td>
<td>2</td>
<td>6000</td>
<td>3000</td>
<td>2999.8490</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Feature</td>
<td>2</td>
<td>11457</td>
<td>5729</td>
<td>5728.5490</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Method:Sequence Length</td>
<td>8</td>
<td>5480</td>
<td>685</td>
<td>685.0230</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Method:Feature</td>
<td>8</td>
<td>12909</td>
<td>1614</td>
<td>1613.5980</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Sequence Length:Feature</td>
<td>4</td>
<td>288</td>
<td>72</td>
<td>71.9470</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Method:Sequence Length:Feature</td>
<td>1157</td>
<td>72</td>
<td>72.3390</td>
<td>156.3400</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>Residuals</td>
<td>4455</td>
<td>4455</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: ANOVA table corresponding to the model described in Table 4.3.

Although the linear model described in Table 4.3 point toward a significant relationship between the predictor variables and the aggregate $F$-score response, further analysis is required in order to determine with greater detail which specific conditions differ and to what degree. As suggested in Chapter 3, Tukey’s HSD post hoc test was employed for this purpose, the results of which are described in the next chapter.
4.6 Conclusion

This chapter outlined the environmental conditions present during data set generation, the algorithm employed to automatically generate the various types of Internet traffic traces under investigation, as well the structure of the univariate linear model employed to facilitate statistical analysis of the experimental results. Evidence supporting satisfaction of the assumptions regarding the nature of the experimental result data was presented in graphical form. Assessment of the linear model’s $F$ statistic indicated that the model was significant. Computation of the ANOVA table revealed that each of the main effects was also significant and in so doing meeting all of the criteria for utilizing Tukey’s HSD post hoc test to determine which specific experimental conditions differ from each other. As a result, it will be possible to assess the strength of evidence for or against the eight stated hypotheses in the next chapter.
Chapter 5

Results

5.1 Introduction

In this chapter the results of the experiment and accompanying analysis introduced in Chapter 4 are presented. Subsequent sections within the chapter aim to gradually narrow the scope of discussion from assessment of the linear model validity to presentation of the evidence for/against the stated hypotheses. This organization was chosen as it coincides well with the statistical approach employed in both the experimental design and analysis process wherein it is necessary to validate general assumptions before more specific inferences can be made regarding parameters of interest.

The hypotheses introduced in Chapter 4 represent the primary motivators for the experimental design employed in this thesis. Each experimental condition was chosen to provide suitable evidence in which to test each hypothesis and ultimately reach a better understanding regarding the generalization performance of network dependent features in the context of 5 state, fully connected HMM-based Internet traffic classifiers. Table 5.1 contains the resulting aggregate $F$-scores for the 45 experimental conditions. The aggregate $F$-scores in bold typeface indicate the highest
score observed across all three levels of the Feature variable for a given combination of Sequence Length and Method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Seq. Length</th>
<th>Feature</th>
<th>Packet Size</th>
<th>Packet Size + IAT</th>
<th>Network Independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small MTU</td>
<td>Small MTU</td>
<td>5</td>
<td>0.8470</td>
<td>0.8224</td>
<td>0.8564</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8906</td>
<td>0.8995</td>
<td>0.8770</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td><strong>0.9009</strong></td>
<td>0.8927</td>
<td>0.8804</td>
</tr>
<tr>
<td>Large MTU</td>
<td>Large MTU</td>
<td>5</td>
<td>0.7416</td>
<td>0.7228</td>
<td>0.8375</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.7805</td>
<td>0.7563</td>
<td>0.8837</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.7980</td>
<td>0.7757</td>
<td>0.9189</td>
</tr>
<tr>
<td>Small MTU</td>
<td>Large MTU</td>
<td>5</td>
<td>0.6939</td>
<td>0.6562</td>
<td>0.7200</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.6739</td>
<td>0.6661</td>
<td>0.7336</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.6749</td>
<td>0.6652</td>
<td>0.7302</td>
</tr>
<tr>
<td>Large MTU</td>
<td>Small MTU</td>
<td>5</td>
<td>0.7430</td>
<td>0.7156</td>
<td>0.8379</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.7811</td>
<td>0.7553</td>
<td>0.8819</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.8029</td>
<td>0.7750</td>
<td>0.9198</td>
</tr>
<tr>
<td>Combined</td>
<td>Combined</td>
<td>5</td>
<td>0.7583</td>
<td>0.7324</td>
<td>0.8148</td>
</tr>
<tr>
<td></td>
<td>Small + Large MTU</td>
<td>10</td>
<td><strong>0.8557</strong></td>
<td>0.8098</td>
<td>0.8484</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td><strong>0.8783</strong></td>
<td>0.8489</td>
<td>0.8748</td>
</tr>
</tbody>
</table>

Figure 5.1: Mean aggregate F-score for each experimental conditional. Note that bold typeface does not indicate a statistically significant difference exists between any of the observations.

Each of the cell means will be subjected to hypothesis tests in order to determine if there is sufficient evidence to conclude that the observed differences are
statistically significant. The outcome of these tests will be used to either support or refute conclusions regarding the relationship between the variables of interest (Feature, Method, Sequence Length) and the resulting classifier generalization performance as measured via aggregate $F$-score.

5.2 Experimental Results

5.2.1 Examination of Hypotheses

5.2.1.1 Hypothesis 1

*Network-independent features can be used to classify Internet traffic*

To begin, it is necessary to determine if network-independent features can be used with any success in the context of the traffic classes and model structure introduced in Chapter 4. Previous studies have shown network dependent features can produce classification accuracy scores as high as 90% when Sequence Length = 5. Since MTU size was not considered in previous studies, it is assumed that the parameter was left unchanged and hence would be equivalent to the “Large MTU” condition. To determine if network-independent features can be used to classify Internet traffic, the condition corresponding to Feature=NI, Method=L, and Sequence Length=5 needs to be tested to ensure it could not have been observed by random chance alone. Given that aggregate $F$-score is employed in this thesis which is itself composed of precision and recall (two statistics which are not universally reported by works reviewed in Chapter 2), for the purposes of defining what aggregate $F$-score can be expected by a “randomized” classifier operating in a multi-label environment it is
assumed that precision and recall are distributed equally. This assumption effectively renders the aggregate F-score equivalent to nominal classification accuracy and as a result it is reasonable to expect an aggregate F-score of 0.16 to be achievable by a “randomized” classifier operating in the scenario defined above. It follows then that $H_0$ and $H_A$ would be defined as:

$$H_0 : b_0 \leq 0.0156,$$

$$H_A : b_0 > 0.0156.$$  

The hypothesis can be tested via calculation of Student’s $t$ statistic,

$$t = \frac{b_0 - \beta_0}{SE(b_0)},$$

$$= \frac{0.83752 - 0.016}{0.00153},$$

$$= 536.9412,$$

$$P(t_{0.4455} \leq 536.9412|H_0) << 0.01.$$  

As such, there is sufficient evidence to reject $H_0$ with $\alpha = 0.05$ and conclude that network independent features can be used to classify Internet traffic.

5.2.1.2 Hypothesis 2

There is a significant change in classification accuracy when sequences of lengths greater than 5 are used in conjunction with network independent features.

Previous results suggested that there was little value in examining sequences of length greater than 5 for the purposes of classification when independently seri-
alized traffic traces were used. To test the validity of this claim in the context of wholly serialized traffic traces, the differences in aggregate $F$-score for the two network dependent features with Sequence Length = 10 and 15 are compared across all experimental conditions with Sequence Length = 5. More formally, $H_0$ and $H_A$ corresponding to the hypothesis above are defined as,

$$H_0 : \forall x, y \mid \mu_{x,y,5} = \mu_{x,y,10} \land \mu_{x,y,5} = \mu_{x,y,15},$$

$$H_A : \exists x, y \mid \mu_{x,y,5} \neq \mu_{x,y,10} \lor \mu_{x,y,5} \neq \mu_{x,y,15},$$

(5.1)

$x \in \{S, L, C, SL, LS\}, y \in \{PS, PI\}$.

If a significant difference is found wherein aggregate $F$-score differs significantly across the aforementioned experimental conditions it would indicate that unlike the independently serialized case, there is value in utilizing sequences of length greater than 5 when multiple concurrent sockets are considered.
Table 5.1 summarizes the results of the hypothesis tests. It is clear via examination of the \( p \)-value and \( \Delta \) columns that there is sufficient evidence to reject \( H_0 \) and as such conclude that there is a significant difference in aggregate \( F \)-score when network dependent features are used and the Sequence Length is changed from 5 to either 10 or 15. In the majority of cases, increasing the sequence length results in an increased aggregate \( F \)-score. The exceptions being those experimental conditions where Method=Small/Large MTU which generated decreases and no change in aggregate \( F \)-score when Feature=PS and Feature=PI and Sequence Length=15 respectively. This result suggests an interaction between the Method=Small/Large MTU condition and Sequence Length variable.
5.2.1.3 Hypothesis 3

Classifiers utilizing network dependent features are affected by changes in either the composition of test/training sets or the specific network configuration present during generation.

If shown to be true in the context of this experimental design, this hypothesis would serve to quantify the shortcoming identified in [54] regarding how well classifiers built using network dependent features cope with differing network configurations present during training and evaluation use cases. Combining the results of the previous hypothesis test with the similarity of previous experimental designs reviewed in Chapter 2, the Method=Large MTU, Sequence Length=15 condition served as the reference for the evaluation of evidence for/against this hypothesis. Formally expressed, the hypothesis takes the following form.

\[ H_0 : \forall x, y, z_{opt}(x, y) \mid \mu_{x,y,z_{opt}(x,y)} = \mu_{L,y,15}, \]
\[ H_A : \exists x, y, z_{opt}(x, y) \mid \mu_{x,y,z_{opt}(x,y)} \neq \mu_{L,y,15}, \]

\[
x \in \{S, L, C, SL, LS\}, y \in \{PS, PI\}, z_{opt}(x, y) = \begin{cases} SL, PS & 5 \\ SL, PI & 10 \\ else & 15 \end{cases} \tag{5.2}
\]

The same general technique seen above for evaluating the results was applied and the following table should be interpreted in the same manner.
Table 5.2: Results of applying Tukey’s HSD to Hypothesis 3.

<table>
<thead>
<tr>
<th>Level Comp.</th>
<th>$\Delta$</th>
<th>Lower 95 C.I. %</th>
<th>Upper 95 C.I. %</th>
<th>adj. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{S,PS,15} - \mu_{L,PS,15}$</td>
<td>0.1029</td>
<td>0.0931</td>
<td>0.1127</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{C,PS,15} - \mu_{L,PS,15}$</td>
<td>0.0803</td>
<td>0.0704</td>
<td>0.0901</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{LS,PS,15} - \mu_{L,PS,15}$</td>
<td>0.0049</td>
<td>-0.0049</td>
<td>0.0148</td>
<td>0.9985</td>
</tr>
<tr>
<td>$\mu_{SL,PS,5} - \mu_{L,PS,5}$</td>
<td>-0.0477</td>
<td>-0.0576</td>
<td>-0.0379</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{LS,PI,15} - \mu_{L,PI,15}$</td>
<td>-0.0008</td>
<td>-0.0106</td>
<td>0.0091</td>
<td>1.0</td>
</tr>
<tr>
<td>$\mu_{SL,PI,10} - \mu_{L,PI,10}$</td>
<td>-0.0902</td>
<td>-0.1000</td>
<td>-0.0803</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{S,PI,15} - \mu_{L,PI,15}$</td>
<td>0.1170</td>
<td>0.1071</td>
<td>0.1268</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{C,PI,15} - \mu_{L,PI,15}$</td>
<td>0.1261</td>
<td>0.1163</td>
<td>0.1359</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
</tbody>
</table>

It is clear via inspection of Table 5.2 that $H_0$ can be rejected with $\alpha=0.05$. As such, there is sufficient evidence to conclude that the performance of classifiers utilizing network dependent features is sensitive to changes in either the test/training set concentration (Method=C, SL, LS) or the underlying network configuration present (Method=S, L). However, the effect was not consistent across all conditions as there was insufficient evidence to conclude a statistically significant difference in mean aggregate $F$-score existed between the Method=LS and and Method=L conditions.

5.2.1.4 Hypothesis 4

*There is a difference in resulting aggregate $F$-score when network-independent features are used and the sequence length employed is increased from 5 to 10 or 15*

This hypothesis is a formalization of the notion that network independent features may benefit more from increases in the number of packets within a trace due to the inherent dimensionality reduction employed in the mapping from packets to transactions/messages. Equation (5.3) contains the formal specification of the corresponding $H_0$ and $H_A$ hypothesis.
\[ H_0 : \forall x \mid \mu_{x,NI,5} = \mu_{x,NI,10} \land \mu_{x,NI,5} = \mu_{x,NI,15}, \]

\[ H_A : \exists x \mid \mu_{x,NI,5} \neq \mu_{x,NI,10} \lor \mu_{x,NI,5} \neq \mu_{x,NI,15}, \quad (5.3) \]

\[ x \in \{S, L, C, SL, LS\}. \]

Table 5.3 contains the results of the hypothesis tests.

<table>
<thead>
<tr>
<th>Level Comp.</th>
<th>( \Delta )</th>
<th>Lower 95 C.I. %</th>
<th>Upper 95 C.I. %</th>
<th>adj. ( p )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu_{L,NI,10} - \mu_{L,NI,5} )</td>
<td>0.0462</td>
<td>0.0361</td>
<td>0.0562</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{L,NI,15} - \mu_{L,NI,5} )</td>
<td>0.0564</td>
<td>0.0466</td>
<td>0.0663</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{S,NI,10} - \mu_{S,NI,5} )</td>
<td>0.0206</td>
<td>0.0105</td>
<td>0.0306</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{S,NI,15} - \mu_{S,NI,5} )</td>
<td>0.0539</td>
<td>0.0440</td>
<td>0.0647</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{C,NI,15} - \mu_{C,NI,5} )</td>
<td>0.1200</td>
<td>0.1102</td>
<td>0.1298</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{C,NI,10} - \mu_{C,NI,5} )</td>
<td>0.0336</td>
<td>0.0236</td>
<td>0.0437</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{LS,NI,15} - \mu_{LS,NI,5} )</td>
<td>0.0819</td>
<td>0.0721</td>
<td>0.0918</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{LS,NI,10} - \mu_{LS,NI,5} )</td>
<td>0.0441</td>
<td>0.0340</td>
<td>0.0541</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{SL,NI,10} - \mu_{SL,NI,5} )</td>
<td>0.0136</td>
<td>0.0036</td>
<td>0.0237</td>
<td>&lt;&lt; &lt; 0.01</td>
</tr>
<tr>
<td>( \mu_{SL,NI,15} - \mu_{SL,NI,5} )</td>
<td>0.0102</td>
<td>0.0003</td>
<td>0.0200</td>
<td>0.0302</td>
</tr>
</tbody>
</table>

Table 5.3: Results of applying Tukey’s HSD to Hypothesis 4.

As can be seen via inspection of the \( \Delta \) and \( p \)-value columns of Table 5.3, there is sufficient evidence to reject \( H_0 \) with \( \alpha = 0.05 \) and conclude that a significant difference in classifier performance is observed when network independent features are used and the number of packets considered is increased from 5 to either 10 or 15. While there is agreement that increases from Sequence Length=5 to 10 results in consistently improved aggregate \( F \)-score, increases from 10 to 15 do not exhibit the same uniform response. Such a result suggests an interaction between the Sequence Length=15 and the Method=Combined MTU and Method=Small/Large MTU conditions when Feature=Network Independent.
5.2.1.5 Hypothesis 5

When the distribution of MTU size within the test/training set is homogeneous, there is a difference in resulting aggregate F-score between network independent vs dependent feature sets.

This hypothesis attempts to establish network independent features as viable alternatives to network dependent features by examining respective classification performance when the test and training set are both homogeneous in composition. As such, the hypothesis involves comparing performance across both the Method=Small MTU and Method=Large MTU conditions. Network independent features were compared against both types of network dependent features and the sequence length employed chosen to align with that which produced the strongest aggregate $F$-score for the network dependent features examined in Hypothesis 2. Equation 5.4 and Table 5.4 present the formal hypothesis to be tested and the results of the tests respectively.

\[ H_0 : \forall x, y, z_{opt}(x, y) \mid \mu_{x,y,z_{opt}(x,y)} = \mu_{L,NI,z_{opt}(x,y)}, \]

\[ H_A : \exists x, y, z_{opt}(x, y) \mid \mu_{x,y,z_{opt}(x,y)} \neq \mu_{L,NI,z_{opt}(x,y)}, \]

\[ x \in \{S, L\}, y \in \{PS, PI\}, z_{opt}(x, y) = \begin{cases} S, PI & 10 \\ else & 15 \end{cases}. \]
133

<table>
<thead>
<tr>
<th>Level Comp.</th>
<th>Δ</th>
<th>Lower 95 C.I. %</th>
<th>Upper 95 C.I. %</th>
<th>adj. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{L,PI,15} - \mu_{L,NI,15}$</td>
<td>-0.1431</td>
<td>-0.1529</td>
<td>-0.1333</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{L,PS,15} - \mu_{L,NI,15}$</td>
<td>-0.1209</td>
<td>-0.1307</td>
<td>-0.1110</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{S,PI,10} - \mu_{S,NI,10}$</td>
<td>0.0225</td>
<td>0.0127</td>
<td>0.0324</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
<tr>
<td>$\mu_{S,PS,15} - \mu_{S,NI,15}$</td>
<td>0.0205</td>
<td>0.0107</td>
<td>0.0304</td>
<td>$&lt;&lt; 0.01$</td>
</tr>
</tbody>
</table>

Table 5.4: Results of applying Tukey’s HSD to Hypothesis 5.

The p-value column of Table 5.4 indicates there is sufficient evidence to reject $H_0$ and conclude that the Feature=Network Independent condition results in a different aggregate $F$-score than that observed under either of the Feature=Packet Size + Inter-arrival Time or Feature=Packet Size conditions, when the optimal sequence is compared at alpha=0.05. However, when the Feature=Small MTU, the network dependent features do indeed outperform the network independent feature set. According to the 95% C.I., the magnitude of this difference in aggregate $F$-score is between 0.012 and 0.03, 95/100 times, whereas the difference under the Method=Large MTU condition is between 0.11 and 0.15, 95/100 times. This discrepancy suggests an interaction between the influence of the Feature=Network Independent and Method=Small MTU conditions.

5.2.1.6 Hypothesis 6

The aggregate $F$-score of classifiers utilizing network-independent features is significantly different from those utilizing network dependent features when the concentration of network configurations within the test/training set is mixed.

Between the two extremes of completely homogeneous and heterogeneous test/training set compositions like those presented above is a third scenario wherein
the composition of MTU size is defined as a mixture of the Method=Small MTU and Method=Large MTU conditions. Although unlikely to be observed in a real-world setting and arguably unrealistic if only a balanced 50% mixture is considered, the more general case where one might expect to find arbitrary mixtures of (possibly different) MTU sizes in test/training sets is at least remotely plausible. For this reason, the sensitivity of the respective features to different test/training set compositions is an important hypothesis to evaluate even if the mixture concentration is not representative of what might be encountered in a real-world setting. Equation 5.5 is a formal expression of the hypothesis to be tested.

\[
\begin{align*}
H_0 : \forall C, y, 15 \mid \mu_{C,y,15} &= \mu_{C,NI,15}, \\
H_A : \exists C, y, 15 \mid \mu_{C,y,15} \neq \mu_{C,NI,15}, & \quad (5.5) \\
y \in \{PS, PI\}.
\end{align*}
\]

The influence of the Method=Combined condition across all features is examined with Sequence Length=15, as this was determined to be optimal as a result of previous hypothesis tests. Table 5.5 contains the results of the hypothesis tests.

<table>
<thead>
<tr>
<th>Level Comp.</th>
<th>( \Delta )</th>
<th>Lower 95 C.I. %</th>
<th>Upper 95 C.I. %</th>
<th>adj. ( p )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu_{C,PS,15} - \mu_{C,NI,15} )</td>
<td>0.0035</td>
<td>-0.0064</td>
<td>0.0133</td>
<td>1.0</td>
</tr>
<tr>
<td>( \mu_{C,PI,15} - \mu_{C,NI,15} )</td>
<td>-0.0259</td>
<td>-0.0358</td>
<td>-0.0161</td>
<td>&lt;&lt; 0.01</td>
</tr>
</tbody>
</table>

Table 5.5: Results of applying Tukey’s HSD to Hypothesis 6.

Examination of the \( p \)-value column indicates there is sufficient evidence to reject \( H_0 \) and conclude that network independent features behave differently than network dependent features under the aforementioned experimental conditions. The
nature of the difference depends on the specific network dependent feature. For Feature=Packet Size + Inter-arrival Time, the difference in resulting aggregate $F$-scores is between about -0.016 to -0.036, 95/100 times whereas there is insufficient evidence to conclude there is a difference between the network independent feature and the Feature=Packet Size condition.

5.2.1.7 Hypothesis 7

Classifiers using network-independent features generate significantly different aggregate $F$-scores than those using network dependent features when the training and test set are generated using completely independent network configurations

This hypothesis is squarely targeted at exposing sensitivity of the resulting aggregate $F$-score to differing network configurations employed during testing and training of a given classifier. If shown to be true, the result would support the notion of a different (and potentially improved) generalization capability of network-independent features. Expressed more formally in (5.6), the hypothesis test compares the relative performance of the network dependent features (Feature=Packet Size, Feature=Packet Size + Inter-arrival Time) to network independent features (Feature=NI) under both “independent” network configuration scenarios, Method=Small/Large MTU and Method=Large/Small MTU.
In all cases the Sequence Length=15 condition was used as it was determined to be optimal as a result of previous hypothesis tests. Table 5.6 contains the results of the test of significance for this hypothesis.

<table>
<thead>
<tr>
<th>Level Comp.</th>
<th>Δ</th>
<th>Lower 95 C.I. %</th>
<th>Upper 95 C.I. %</th>
<th>adj. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>µ_{LS,PS,15} - µ_{LS,NI,15}</td>
<td>-0.1169</td>
<td>-0.1267</td>
<td>-0.1070</td>
<td>&lt;&lt; 0.01</td>
</tr>
<tr>
<td>µ_{LS,PI,15} - µ_{LS,NI,15}</td>
<td>-0.1448</td>
<td>-0.1547</td>
<td>-0.1350</td>
<td>&lt;&lt; 0.01</td>
</tr>
<tr>
<td>µ_{SL,PI,15} - µ_{SL,NI,15}</td>
<td>-0.0650</td>
<td>-0.0748</td>
<td>-0.0551</td>
<td>&lt;&lt; 0.01</td>
</tr>
<tr>
<td>µ_{SL,PS,15} - µ_{SL,NI,15}</td>
<td>-0.0553</td>
<td>-0.0651</td>
<td>-0.0454</td>
<td>&lt;&lt; 0.01</td>
</tr>
</tbody>
</table>

Table 5.6: Results of applying Tukey’s HSD to Hypothesis 7.

Inspection of Table 5.6 reveals that there is sufficient evidence to reject $H_0$ and conclude that network independent features lead to significantly different resulting aggregate $F$-scores. In both conditions which qualify as “independent” network configurations, network independent features resulted in an aggregate $F$-score between 0.045 and 0.15 greater, 95/100 times. This result suggests that network independent features are less sensitive to discrepancies in the underlying network configuration present during testing/training of Internet traffic classifiers than network
dependent features.

5.3 Conclusion

In preparation for further discussion, the results presented in this chapter are summarized in Table 5.7.
<table>
<thead>
<tr>
<th>Section</th>
<th>Experimental Condition</th>
<th>Hypothesis</th>
<th>Result</th>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1.1</td>
<td>METHOD=L FEATURE=PS,PI LENGTH=5</td>
<td><em>Network independent features can be used to classify Internet traffic</em></td>
<td>Confirmed</td>
<td>None</td>
</tr>
<tr>
<td>5.2.1.2</td>
<td>METHOD=L,S FEATURE=PS,PI LENGTH=5,10,15</td>
<td><em>There is a significant change in classification accuracy when sequences of lengths greater than 5 are used in conjunction with network independent features</em></td>
<td>Confirmed</td>
<td>None</td>
</tr>
<tr>
<td>5.2.1.3</td>
<td>METHOD=L,S,C,LS,SL FEATURE=PS,PI LENGTH=5,10,15</td>
<td><em>Classifiers utilizing network dependent features are affected by changes in either the composition of test/training sets or the specific network configuration present during generation</em></td>
<td>Confirmed</td>
<td>Insufficient evidence to reject $H_0: \mu_{LS,PS,15} = \mu_{L,PS,15}$</td>
</tr>
<tr>
<td>5.2.1.4</td>
<td>METHOD=L,S,C,LS,SL FEATURE=NI LENGTH=5,10,15</td>
<td><em>There is a difference in resulting aggregate F-score when network-independent features are used and the sequence length employed is increased from 5 to 10 or 15</em></td>
<td>Confirmed</td>
<td>None</td>
</tr>
<tr>
<td>5.2.1.5</td>
<td>METHOD=S,L FEATURE=NI,PS,PI LENGTH=10,15</td>
<td><em>When the distribution of MTU size within the test/training set is homogeneous, there is a difference in resulting aggregate F-score between network independent vs dependent feature sets.</em></td>
<td>Confirmed</td>
<td>None</td>
</tr>
<tr>
<td>5.2.1.6</td>
<td>METHOD=C FEATURE=NI,PS,PI LENGTH=15</td>
<td><em>The aggregate F-score of classifiers utilizing network-independent features is significantly different from those utilizing network dependent features when the concentration of network configurations within the test/training set is mixed.</em></td>
<td>Confirmed</td>
<td>Insufficient evidence to reject $H_0: \mu_{C,PS,15} = \mu_{C,NI,15}$</td>
</tr>
<tr>
<td>5.2.1.7</td>
<td>METHOD=LS,SL FEATURE=NI,PS,PI LENGTH=15</td>
<td><em>Classifiers using network-independent features generate significantly different aggregate F-scores than those using network dependent features when the training and test set are generated using completely independent network configurations.</em></td>
<td>Confirmed</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 5.7: Substantiated conclusions to the seven hypotheses introduced in Chapter 4 concerning the relative generalization performance of network independent vs. dependent features in the context of 5-state, fully-connected, HMM-based Internet traffic classifiers.
The results summarized in Table 5.7 confirm the validity of each the stated hypotheses. In each case, sufficient evidence of a significant difference was observed, supporting confirmation of each hypothesis, in turn. The viability of wholly serialized traffic traces in conjunction with network independent features in the context of Internet traffic classification was empirically demonstrated. In addition, the observed sensitivity of network dependent features to changes in the underlying network configuration serves to not only provide a plausible explanation to observations in [54], but also to reinforce the superiority of network independent features in general.
Chapter 6

Discussion

The results presented in the previous chapter can be summarized by two related principles: the sensitivity of network dependent features to variability in the distribution of MTU size present in the test/training set employed to construct a HMM-based Internet traffic classifier and the viability of network independent features as less-sensitive and better performing alternatives. However, what is not clear from examination of the evidence in support of the aforementioned hypotheses are the possible real-world implications, especially when considering the exceptions and possible interactions amongst experimental conditions which were identified. This chapter will aim to address these issues, as well as highlighting future avenues for further research and possible improvements to the methodology employed.

6.1 Comparison to the state-of-the-art

The results presented in the previous chapter stand in contrast to some of those reviewed in Chapter 2, the most prominent of which being the discrepancy in optimal sequence length. In previous works, sequence lengths of five were found to be optimal, whereas in this thesis sequence lengths of 15 were found to be op-
timal and evidence suggests greater performance might be possible if even larger sequences lengths were employed. One significant difference between this work and those surveyed is how Internet traffic features were serialized into feature vectors. In the literature reviewed, serialization revolved around packets exchanged via a single socket connection, whereas in this thesis all concurrently active socket connections between a host/client were utilized. This difference likely contributed to the differing findings due to the offset nature of concurrent socket connections within a wholly serialized traffic trace. If the presence of packets within multiple sockets are offset and their respective packets serialized into a single stream (as they are in this thesis), the entropy attributed to the first five packets within each constituent socket is aggregated in the wholly serialized trace. Depending upon the degree of offset of each socket within the trace, it is conceivable that each respective socket contributes a unique set of packets to the resulting serialized feature vector. In so doing, the entropy associated with the first few packets of each socket is extended to the entire sequence. While there is insufficient data to determine if such an increase in entropy was observed as a result of utilizing wholly serialized traffic traces, the observations of increased performance (for both network dependent and independent feature sets) with sequence lengths greater than five provides some support for this claim.

Another point of comparison with the state-of-the-art involves the nominal classification performance observed concerning both network independent and dependent features. While it is difficult to draw sound conclusions regarding relative performance given the number of variables in conflict between each respective experimental design, it is reasonable to state that the classification performance reported in
Chapter 5 is similar to that found in contemporary research concerning HMM-based Internet traffic classifiers. With respect to the experimental conditions involving network dependent features employed herein, the similarity in nominal classification performance to contemporary research serves to reinforce the validity of the approach taken to generate the majority of traffic traces used for testing/training. While by no means conclusive (especially considering the influence of the serialization process) establishing similarity to existing works in this way aided in the task of assessing the contribution of other factors like network independent features and different underlying network configurations, both of which were significant aspects of this study. Where classification performance did disagree with that reported by similar studies, the existence of key differences in the environment or feature set supported the notion that such differences were due to experimental design as opposed to disagreement in fundamental assumptions regarding expected behaviour of network dependent features. These observations were further supported by statistically valid experimental designs and associated hypothesis tests, thus leaving little doubt as to nature of the discrepancy. As a result, it is reasonable to state that network independent features are viable alternatives to network dependent features when wholly serialized Internet traffic traces are used.

The final discussion point of note regarding the implications of the results concerns the sensitivity of network dependent features to changes in the underlying network configuration present during test/training set generation. While it is true that all three different feature sets exhibited significant decreases in classification performance, the most significant decrease was observed when network dependent
features were employed. This is not surprising given the relationship between MTU and packet size and inter-arrival time whereby the distribution of packet size (and as a consequence, the inter-arrival time) is effectively limited by the MTU. As a consequence, it is conceivable that poor generalization performance reported in contemporary literature can be attributed to discrepancies in MTU distribution in the data sets utilized for training and testing. However, while the (slightly less) poor performance of the network independent message size feature indicates a less sensitive feature and as a consequence a viable alternative, it also draws attention to a common weakness shared by the two feature sets. One plausible explanation for the common degradation in performance could be related to how sequence length was harmonized in both experimental conditions. Since a sequence length of $N$ packets with an MTU of 1500 bytes contains a considerably different number of bytes than one observed with a sequence length of $M$ packets with an MTU of 576 bytes, it is reasonable to entertain the possibility that such significant drop in performance may be attributable (at least in part) to this aspect of the experimental design. To address this shortcoming, the number of bytes in the sequence could be considered in place of the number of packets, which would effectively eliminate the influence of the relationship between sequence length and MTU from unfairly influencing the resulting classification performance in the particular challenging conditions where the MTU size differs between the test and training data sets. Independent of this however, network independent features were shown to exhibit less sensitivity to such changes and as a consequence satisfy one of the primary goals of this study which is to investigate, assess and address possible causes of poor generalization performance reported
6.2 Review of Contributions and Advances

Although much attention was given to assessing the viability of message size and socket enumeration (collectively referred to as “Network Independent (NI)” traffic features, how multiple concurrent sockets are serialized into a single feature vector and the possible affect on the resulting classifier performance warrants further discussion. In contrast to previous studies, which utilized Flow-Level features derived from single socket connections, the Connection-Level approach employed utilized all concurrent socket connections between a host/client. While not all types of Internet traffic utilize multiple concurrent socket connections, for the classes studied in this thesis (which were chosen to be representative of popular types of Internet traffic) Table 3.2 is evidence of how prevalent multiple concurrent sockets are. Failure to adequately capture this aspect of traffic behaviour makes the classification task more difficult via increasing the variance observed in the resulting feature vectors for each class under consideration. This increase in variance stems from the artificial grouping of possibly different types of protocol traffic together under a single over-arching traffic class, as is the case when each atomic socket connection is considered an example of a given class of traffic. However, many types of Internet traffic routinely utilize multiple concurrent sockets in a predictable manner, often utilizing different types of protocols for each respective concurrent socket connection. For example, the “YouTube” class described in Table 3.2 utilizes at least five concurrent socket connections 95% of the
Figure 6.1 is an example of an actual YouTube traffic trace, consisting of five concurrent TCP and UDP socket connections.

Of the five concurrent connections, two utilize the TCP transport protocol, while the remaining three utilize UDP. The protocol of establishing a connection via exchange of TCP packets, followed by the initiation of UDP packet flow ending with the exchange of TCP packets is a distinguishing feature of Youtube traffic which is lost if multiple, concurrent sockets are not considered. Further, if the goal is to deploy a trained classifier into a real network, the ability to identify groups of concurrent socket connections as belonging to a single instance reduces the number of concurrent classifications that need to occur as the concurrent sockets identified as belonging to a single instance of a given class need not be analyzed further.

Hypothesis 7 was primarily concerned with testing if the poor generalization performance of HMM-based Internet traffic classifiers reported in contemporary
literature could be attributed to the sensitivity of network dependent features to discrepancies in the MTU present in the training data versus the operating environment where the classifier is ultimately deployed and evaluated. To evaluate this hypothesis, classifiers trained using traces consisting exclusively of the minimum or a range consisting of the minimum to maximum were tested using the opposite condition. Under this scenario, network independent features were found to consistently outperform dependent features providing sufficient evidence to conclude that dependent features are more sensitive to discrepancies in MTU and as a result exhibit poorer generalization performance. In addition to continuing to motivate the search for better traffic features which are less sensitive to the underlying network configuration, this result supports further investigation into the distribution of MTU within the various publicly available data sets typically employed for Internet traffic classification research in order to gain a sense of whether the sensitivity identified in this thesis could be implicated as the source of the poor generalization performance reported by recent studies.

It is important to note however that while network dependent features did indeed lead to inferior aggregate $F$-score under the Method=Small/Large MTU and Method=Large/Small MTU conditions, the magnitude of difference with respect to Method=Large MTU condition (which most closely resembles those described in surveyed literature) appears suspiciously similar. To determine if the magnitude of difference can be attributed entirely to the aforementioned sensitivity would require a new set of hypotheses and accompanying statistical analysis. Such an analysis was left as a possible extension to the work presented here as it was beyond the scope
of the original research objective which is satisfied by the results presented in the preceding chapter.

6.3 Possible Improvements & Future Research

Although this work is unique when compared to contemporary Internet traffic classification research with respect to the structured experimental design and accompanying analysis of results, there are none-the-less areas which could be improved. The generation of the BitTorrent and VOIP datasets was generally passive in nature and could have been improved via employing an approach more in line with the other traffic classes. This would have the benefit of removing doubt regarding the representativeness of the respective data sets. Along similar lines, the experimental design could have been improved by better isolating specific variables of interest. As an example, the effect of the serialization of multiple sockets is impossible to isolate given the experiment design employed in this study where all classifiers evaluated utilized features extracted from multiple concurrent sockets. Finally, the confounding of the two network independent features of message size and socket enumeration prevents examining the individual relationship of either feature on the resulting generalization performance. While these shortcomings do indeed limit what can be gleaned from this study, the experimental design and analysis employed were sufficient to address stated research objectives and as such were by no means grossly deficient or inadequate.

In addition to addressing the limitations noted above, there are logical extensions to this study which would be valuable to explore. The extensions are concerned
mainly with widening the range of variables considered, though the results described above also motivate investigation along entirely different lines of enquiry. The first extension involves determining the ideal sequence length for classification purposes when multiple concurrent sockets are serialized into a single feature vector. As noted, previous research showed a sequence length of five to be optimal when network dependent features were employed. The results of this study, however, are not conclusive regarding whether a sequence length of 15 is optimal. Via simply changing the number of packets considered via introduction of new experimental conditions corresponding to larger sequence lengths, this question could be addressed. Either in parallel or as a separate investigation, an examination of variability of MTU across different networks would also seem to be a next logical step. Given that the aforementioned results highlight network dependent features as being sensitive to differences in MTU between test and training data sets, it would be worthwhile to measure how variable MTU actually is across the Internet. Results of such an investigation would seem not to be directly applicable to any specific classifier or feature set, but would rather provide valuable insight into the possibility of poor generalization performance arising due to mismatches in MTU. Finally, extending the number of features evaluated, the number of traffic classes considered or the types of classifier architectures would all be straightforward extensions worthy of further study. Each of these extensions can be readily accommodated by the experimental design employed, though the resulting analysis would benefit from addressing some of the aforementioned weaknesses as well.
Chapter 7

Conclusion

The primary goal of this thesis was to determine if network independent features could be used to achieve improved generalization performance in 5-state, fully-connected, HMM-based Internet traffic classifiers. In recognition of the possibility of multi-socket correlational structures, wholly serialized Internet traffic traces were employed. To overcome concerns regarding establishing ground-truth, all traffic studied was either generated or passively recorded under defined and controlled conditions. Due to the combination of both fixed and random variables involved, a statistically sound methodology was employed which influenced all aspects of the investigation, including hypothesis formulation and statement, experimental design and analysis of results. This methodology facilitated objective assessment of evidence in support of stated hypotheses regarding the superior generalization performance network independent features offer over network dependent features. As a result of the application of a sound experimental design and associated statistical analysis techniques, substantiated conclusions regarding the superior generalization performance of network independent features were reached and insight into a possible explanation for poor real-world Internet traffic classifier made. As a consequence of the approach taken, the results within can be readily replicated, verified and extended into new
directions. Though not originally conceived of as something this work should aim to address explicitly, the statistical techniques employed to assess the relative performance of multiple classifiers serves as an additional contribution of this work to the field of Internet traffic analysis. Finally, it is hoped that both the findings of this research endeavour and the methodology used to discover them serve as novel contributions to the field.
Bibliography


