BEHAVIOURAL FEATURE EXTRACTION FOR CONTEXT-AWARE TRAFFIC CLASSIFICATION OF MOBILE APPLICATIONS

by

AUN YICHIET

Thesis submitted in fulfillment of the requirements

for the Degree of

Doctor of Philosophy

June 2018

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Selva for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor for my PhD study.

Besides my supervisor, I would like to thank my co-supervisors of Dr. Shanker for his insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

Last but not least, I would like to thank my family and friends that are always by my side, motivate me and help me, to make my work possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SDA	Sanitized Data Collection
SCA	Data Sanitization Method
BFE	Feature Learning Methods
CAT	Context-Aware Traffic Classification Model
DBA	Dependencies Behavioural Analysis
PCA	Payload Correlations Analysis
SAB	Stateful Application Behavioural Traits
FE	Feature Extraction Methods
LCS	Longest Common Substring
LD	Leviathan Distance
TF	Term Frequency
HTTP	Hypertext Transfer Protocol
TLS	Transport Layer Security
CFS	Correlational Feature Selection

KAEDAH PENGENALAN DESKRIPTOR TRAFIK YANG BERKONTEKS INSAF UNTUK MENGKLASIFIKASI TRAFIK APLIKASI-APLIKASI DALAM PLATFORM MUDAH-ALIH

ABSTRAK

Pengunaan applikasi mudah-alih yang semakin meningkat telah mempelbagaikan jenis kelas trafik. Fenomena ini menambah kepada kerumitan dalam kaedah klasifikasi trafik untuk mengenal-pasti kelas-kelas trafik sedemikian. Sepadannya, usaha dalam meningkatkan keberkesanan kaedah klassifikasi terutamanya dalam meningkatkan kualiti and kuantiti deskriptor trafik amat diperlukan. Tesis ini menganalisis ciri-ciri rangkaian Internet demi mengenalpasti deskriptor-deskriptor trafik yang lebih berkesan berasaskan kaedah pembelajaran-mesin. Tesis ini mereka bentuk rangka kerja yang mengandungi satu kaedah sanitasi data, satu kaedah pengenalpasti trafik deskriptor dan satu kaedah pemetaan algoritma klasifikasi yang bersesuaian dengan skop trafik-trafik yang dipertimbangkan. Terutamanya, satu kaedah sanitasi data direka bentuk untuk menapis paket IP yang tidak berkenaan daripada data trafik demi meningkatkan kualiti data untuk membina model yang berkesan. Seterusnya, kaedah pengenalpastian deskriptor menganalisi korelasi trafik SSL dan HTTP dengan applikasi iOS untuk mencari deskriptor yang berpotensi meningkatkan ketepatan klassifikasi. Seterusnya, satu kaedah pengesan deskriptor dari muatan paket IP direka bentuk untuk memudahkan pengemas-kinian deskriptor secari automatik bersepadan dengan applikasi yang sentiasa dikemas kini. Dengan integrasi kaedah-kaedah ini, klasifikasi didapati telah meningkat setinggi 0.923 dan 0.902 dalam unit ukuran 'ketepatan'.

BEHAVIOURAL FEATURE EXTRACTION FOR CONTEXT-AWARE TRAFFIC CLASSIFICATION OF MOBILE APPLICATIONS

ABSTRACT

Traffic classification is becoming more complex due to proliferations of mobile applications coupled with growing diversity of traffic classes. This motivates the needs for improved traffic classification method that preserve classification accuracy while supporting more traffic classes. This thesis identified domain-specific features that are effective for accurate, large-scale and scalable mobile applications classification using machine learning techniques. This thesis designed a context-aware traffic classification framework that includes a set of sequential algorithms from cleaning datasets, to identifying new features and detecting optimal classifier(s) based on problem contexts to improve classification accuracy in multi-variate traffic classification. In data pre-processing, a data sanitization algorithm is designed to filter irrelevant traffic noise such as control traffic and Operating System induced traffic to improve the quality of training data. In feature engineering, the statistical correlations of user applications towards Secure Sockets Layer (SSL) and Hypertext Transfer Protocol (HTTP) behaviours are decoded into features using domain knowledge to better discriminate encrypted applications. Meanwhile, a self-learning signature construction method is designed to update application's signatures to reflect salient behavioural changes in mobile application that are constantly updating. Lastly, a context-aware algorithm is designed for classes-specific detection of corresponding optimal classifiers and feature set for improved accuracy in multi-variate traffic classification. The experimental results showed improved classification accuracy compared to the closest existing works at 0.923 for encrypted traffic and 0.902 for unencrypted traffic using the resulting models tested on 10-fold validation using J48 as the classifier. The experiment results consolidate the significance of proposed framework in addressing growing diversity of traffic classes towards pervasive and accurate traffic classification.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Traffic classification is a popular domain in machine learning to autonomously categorize traffic classes based on communication traits (Nguyen & Armitage, 2008; Anantavrasilp, 2009; Puš, 2012; Yildirim & Radcliffe, 2010). It is based on the principle that all applications exhibit some deterministic communication behaviors (Nguyen & Armitage, 2008; Anantavrasilp, 2009; Puš, 2012) that can be used to distinguish flows identity. For Internet Service Providers (ISP), traffic classification enables Quality of Service (QoS) management at the application level (Nguyen & Armitage, 2008; Nossenson & Polacheck, 2015). Meanwhile, traffic classification is also useful for pervasive indexing of services, servers and connected devices coherent to the proliferation of Internet of Things (*IoT*) (Kraijak & Tuwanut, 2015). Traffic classification also adds perceptible information to some avenues of network monitoring; such as by providing network engineers with accurate applications identity rather than implying on unreliable port numbers inferencing during forensic analysis. In the early days, applications are less diverse and traffic flows can be manually annotated (Dong, Guo, Li, Xu, & Wei, 2016; Hsieh, Tung, & Lin, 2016). Eventually, the proliferation of mobile applications and IoT led to the diversification of applications traffic (Vincent, F. & Riccardo S, Mauro, 2016).

Today, mobile application traffic dominates an enormous share of total traffic in global Internet and continues to grow (Qinglong,W. & Amir, 2015). For some context, Figure 1.1 depicts the implied steady growth of mobile traffic chronologically (from time spent); while Figure 1.2 shows the increasing dominance of mobile traffic over their web-based counterparts. The proliferation of mobile applications has induced traffic classes diversification that adds to application classification's complexity (Dong, Guo, Li, Xu, & Wei, 2016).

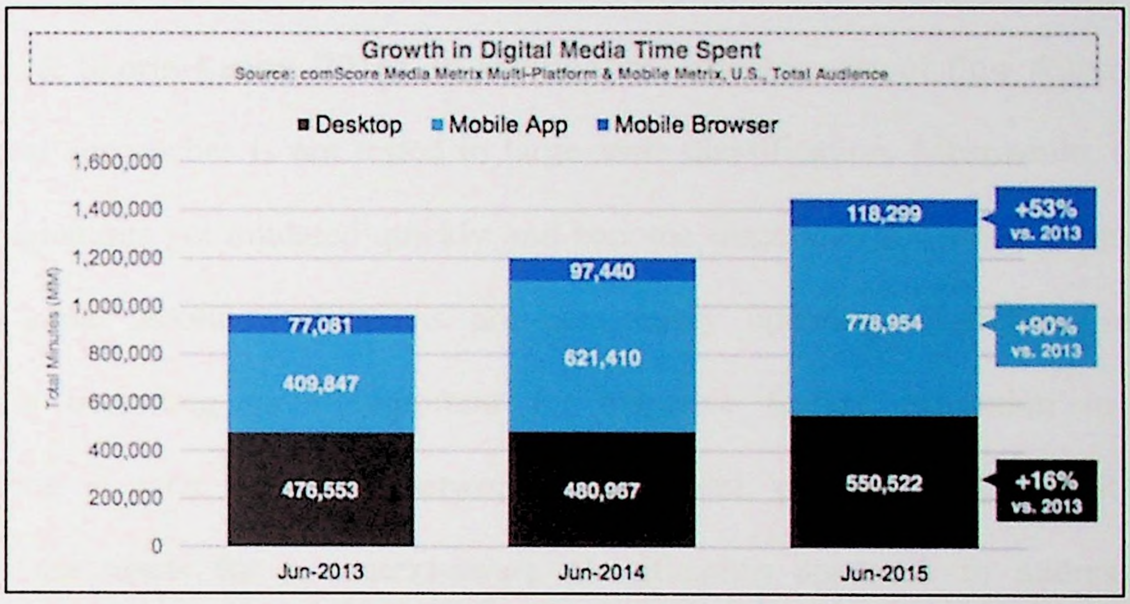


Figure 1.1 Growth of Digital Media Time Spent

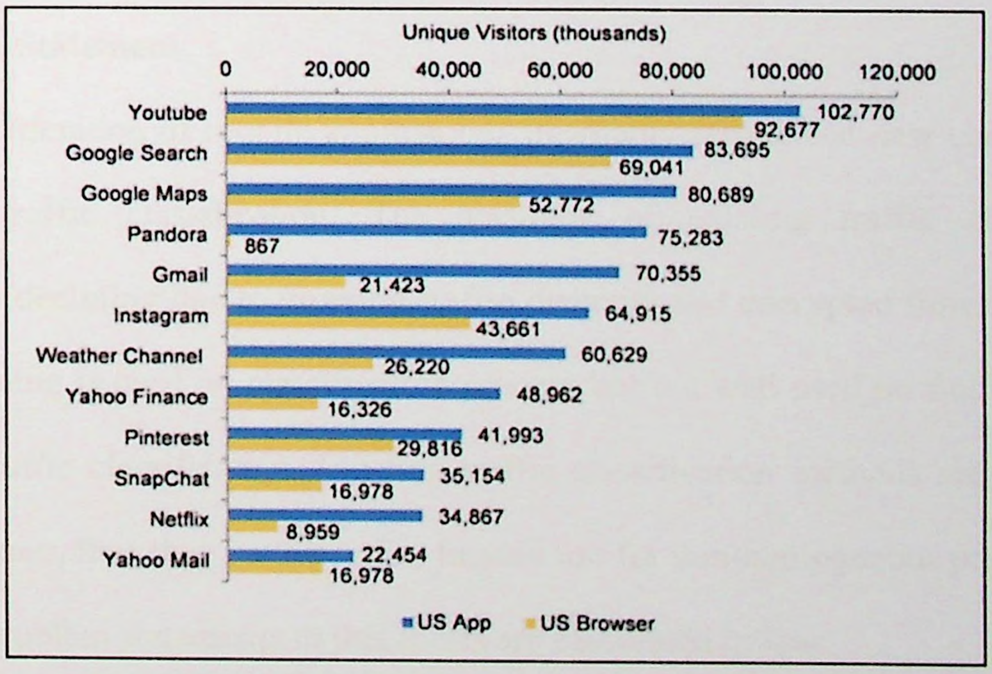


Figure 1.2 Comparing the Popular Application Platforms; between App-based and Web-based

Consequently, traffic classification becomes difficult since user applications are less predictable in communication behaviors as observed in their random port usage and lack of documented *RFC(s)* by Internet Engineering Task Force (IETF) for heuristic analysis. As existing traffic classification methods are designed for small synthetic applications sets; these methods become less accurate for classification of mobile traffic that showed more dynamic and diverse behaviours (Liu, et al., 2016; Orsolic, Pevec, Suznjevic, & Skorin-Kapov, 2016). In addition, the effectiveness of flow features used in statistical approaches is not tested in large-scale classification. Meanwhile, existing payload signatures got outdated quickly and become inaccurate for mobile application detection since mobile applications are persistently updating. Therefore, machine learning is becoming more important for effective feature extraction in traffic classification to cater for future network management, planning, and security. This motivates the needs for a context-aware classification approach to address these emerging challenges towards accurate and robust mobile application classification.

1.2 Problem Statement

The proliferation of mobile applications inevitably introduced new challenges for real world traffic classification. The accuracy of existing traffic classification approaches is declining due to growing traffic diversity and encrypted flows. Currently, machine-learning is used on classification process but not well-used on finding features needed for traffic classification. Existing traffic classification methods are mostly not contextual aware, thus they have a rather limited use for non-homogenous problems. Three main problem statements in this thesis are elaborated below:

Problem 1: Existing data collection techniques do not effectively reduce traffic noise

Flows in training data are currently being labelled based port information is lacking of data integrity (Nguyen & Armitage, 2008) due to irregular usage of port number(s). In addition; to my best knowledge, there is currently no efficient mechanism to reduce traffic noises from training data in existing data collection methods.

Problem 2: Existing flow feature and payload signatures are not optimized for classifying large numbers of mobile traffic classes

The vector of flow statistic features is susceptible to network congestion and conditions (Zhang & Figueiredo, 2006; Khurana, Turaga, Samulowitz, & Parthasrathy, 2016). Meanwhile, payload signature extraction requires domain knowledge and requires manual signature update to scale with application updates. In addition, existing feature extraction methods are not self-learning (Goo, Shim, Lee, & Kim, 2016); thus, newer features that are more correlated to application's behaviours are not automatically extracted to reflect periodical changes in application behaviours or in detecting new traffic classes.

Problem 3: Existing traffic classification methods are designed to classify predetermined set traffic classes

Traditionally, traffic classification identifies an optimal classifier with some pre-selected feature set corresponding to a fixed set of applications of interests (Nguyen & Armitage, 2008). In static selection approach, the synergy between feature and classifiers towards traffic classes are not explored (Thornton, Hutter, Hoos, & Leyton-Brown, 2013; Komer et al., 2014; Kiepas, Bobek, & Nalepa, 2015). Implicitly, domain knowledge is required to determine optimal pairing of classifiers and feature set to every new classification problems (different compositions of traffic classes).

1.3 Research Questions

The thesis's main research question is '*how to enhance the components of traffic classification to optimally classify user applications in high diversity.*

Specifically,

1. <u>On the increasing traffic diversity</u>
    a. What is the impact of increasing traffic diversity on the classification performance of existing traffic classification methods?
    b. Are existing traffic classification methods effective in discriminating mobile applications' traffic?
2. <u>On enhancing data collection methods</u>
    a. Does traffic noise influence the integrity of classifier models' construction; and how to integrate noise reduction mechanism in collecting and annotating training data?
3. <u>On automating feature engineering with machine learning</u>
    a. Can existing feature sets effectively discriminate large-scale traffic classes; and how machine learning can improve existing feature engineering to identify more useful features in the pervasive and scalable manner?
    b. Can feature search space be expanded beyond traffic classes of interest itself; to some adjacent and correlated protocols' flows?

   c. What is the most optimal length of payload data needed for accurate traffic classification?
   d. How to parameterize application communication traits and find network metrics equivalence in describing these fine grain behaviors?

4. <u>On addressing the homogenous problem in traffic classification methods</u>
   a. What is the bonafide classifier and feature sets in traffic classification domain, if any; and how to identify the most optimal classification component variables for multi-context classification problems effortlessly?

1.4 Research Objectives

The research objectives are:

1. To design a dataset annotation and cleaning method for feasible large scale dataset labelling while reducing traffic noise in data pre-processing.
2. To design feature-learning methods for automated identification of domain features to improve the classification accuracy on growing diversity of traffic classes.
3. To design a context-aware traffic classification architecture for automatic detection of optimal classifier(s) and feature sets based on dynamic compositions of application sets.
4. To design independent experiments to evaluate the impact of sanitized data and domain features towards overall improvement on traffic classification accuracy.

1.5 Research Contributions

The contributions of this thesis are to improve application classification accuracy while extending application diversity support. This thesis proposes a pervasive and robust traffic classification system for accurate classifications mobile application in high diversity with the use of machine learning in feature engineering and classification process. Individual contributions are elaborated below:

1. <u>Developed a sandbox data collection method (SDA) to improve the signal-to-noise ratio of annotated datasets</u>

   SDA collects data in a sandbox environment to minimize unwanted traffic noise that includes control traffic, iOS-induced traffic and peer applications traffic in data pre-processing to improve the integrity of training data.

2. <u>Identified highly-correlated features using domain knowledge that are more effective in discriminating mobile traffic classes</u>

   The proposed *behavioral feature extraction* (BFE) use machine learning and domain knowledge to enhance feature extraction. BFE extends the search area of features from target applications of interest to correlated neighbor protocols while using reinforcement learning to automatically update applications' signatures.

3. <u>Optimized the selection of features and classifiers based on traffic compositions</u>

   Context-aware classification adaptively finds the most optimal feature sets, classifiers and their corresponding parameters based on classification problems (multi-variate traffic classes compositions) to enable accurate classification in heterogeneous network environment.

4. Improved the accuracy and robustness of traffic classification while supporting higher application diversity

   a. The traffic's diversity is expanded to supported 50 unique mobile applications running on *iPhone Operating System* (iOS). The proposed methodology improves classification accuracy using application layer features enhanced with domain knowledge to address diversity complexity and applications' behavioral complexity.

1.6 Research Scope

Concerning the Scope of Applications of Interests

The classifier model of the proposed method is trained using 50 iOS applications; therefore, it is only capable of classifying the traffic flows of these 50 applications only. The selections of application of interests are based on regional Appstore (MY) popularity at the time of writing. Although the thesis is evaluated using traffic classes limited to iOS mobile applications; the proposed data collection method is actually universal for multi-variate traffic classes.

Concerning Feature Extraction

The learning mechanic in this thesis is used for to identify features; instead of learning on new application behaviors. Therefore, this approach is not capable of classifying new applications unless the apriori information is available. In event of application updates and behavioral changes, the proposed method is only able to detect if the model is retrained with new training data reflecting those changes. In this thesis, time and computation complexity is not part of the optimization for both feature learning and classification. PCA and SAB features are only works on unencrypted traffic.

Concerning Performance Evaluation

The proposed methods are not tested with short flows; since they normally do not contain sufficient information to be classified. *Timeliness* is not considered as the proposed work is not a real-time system. *Memory usage* is also not accounted as these metrics are specific to payload signature optimization methods. For this reason, performance analysis (big(O)) for the algorithms proposed in this thesis are not evaluated. The computation resources are not significant as large-scale machine learning model can be trained using cloud such as IBM's Hadoop platform.

1.7 Research Framework

There are five components in this thesis leading the research accomplishments. The flow of research stages is depicted in Figure 1.3.

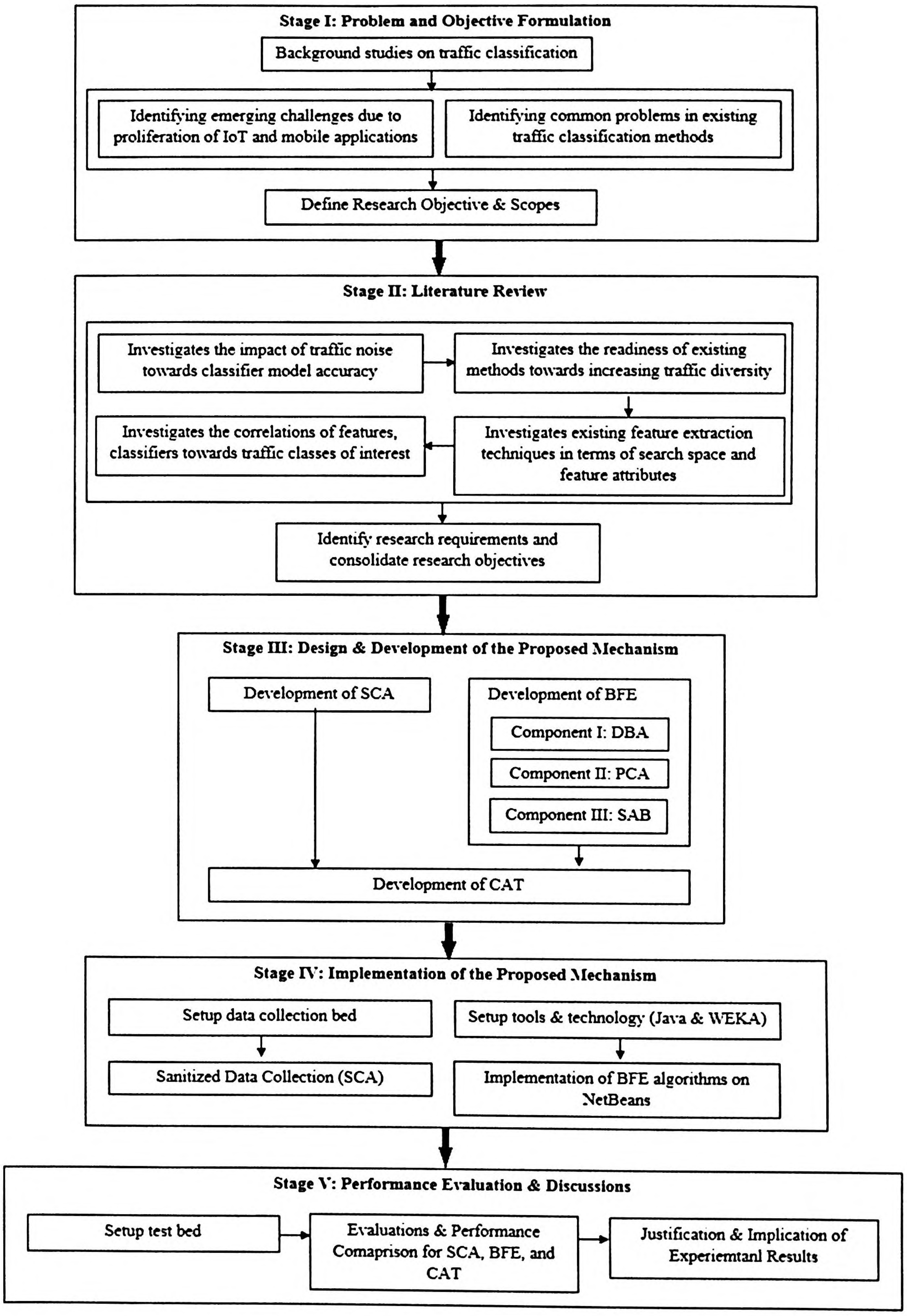
The first stage of the research is concerned with background studies and problem formulation. Here, the emerging challenges in traffic classification induced by the proliferation of mobile applications and Internet of Things are identified. Persistent problems that are common in this literature such as the homogenous nature of classification scope is investigated. These findings are then used to clearly formulate the problem statements, research objectives, and projected contributions of this thesis.

The second stage critically reviewed existing traffic classification methods in the context of emerging challenges such as increasing traffic diversity towards real-world classification. The studies are important to synthesize and gauge the readiness of existing works in addressing new challenges in modern network classification. Some pre-exquisite works including evaluating the correlations among classification components and evaluating the accuracy of existing methods in large-scale context are presented here. These findings are then used to consolidate the requirements and significance of the proposed mechanism.

The third stage of the research presents the proposed mechanism to tackles the set of problems defined in section 1.2 and achieve the objectives in section 1.4. This thesis proposed three independent approaches to address scalability and diversity concerns in traffic classification; in data collection, feature engineering and classification phases. *Sanitized data collection and annotation* (SDA) is a sandbox data acquisition technique used to label apriori needed for subsequent classification. SDA also improve resulting classifier model accuracy by reducing traffic noise and irrelevant flows in training data. *Behavioral feature extraction* (BFE) is a feature learning method that is further divided into three components to improve features quality and quantity. BFE use DBA to improve feature search space, PCA for unsupervised and scalable signature extraction and SAB to enhance feature set using heuristics. *Context-aware traffic classification* (CAT) is an adaptive model to select optimal features set and classifiers algorithms based on types of traffic classes involved. CAT serves to aggregate all the proposed work by providing a universal traffic classification model to address the constantly changing application compositions in a modern network.

Extending from the previous stage, the <u>fourth stage</u> shows the implementation of the proposed mechanism. Here, the respective architectures, frameworks, algorithms, models, and flow are implemented and described in the order of SDA, BFE, and CAT.

Lastly, <u>stage five</u> evaluate and compare classification accuracy of the proposed model against some prominent classification works. Here, several experiments are designed to verify the accomplishment of proposed components in achieving the objectives listed in Section 1.4.



Stage I: Problem and Objective Formulation
Background studies on traffic classification
Identifying emerging challenges due to proliferation of IoT and mobile applications
Identifying common problems in existing traffic classification methods
Define Research Objective & Scopes
Stage II: Literature Review
Investigates the impact of traffic noise towards classifier model accuracy
Investigates the readiness of existing methods towards increasing traffic diversity
Investigates the correlations of features, classifiers towards traffic classes of interest
Investigates existing feature extraction techniques in terms of search space and feature attributes
Identify research requirements and consolidate research objectives
Stage III: Design & Development of the Proposed Mechanism
Development of SCA
Development of BFE
Component I: DBA
Component II: PCA
Component III: SAB
Development of CAT
Stage IV: Implementation of the Proposed Mechanism
Setup data collection bed
Setup tools & technology (Java & WEKA)
Sanitized Data Collection (SCA)
Implementation of BFE algorithms on NetBeans
Stage V: Performance Evaluation & Discussions
Setup test bed
Evaluations & Performance Comaprison for SCA, BFE, and CAT
Justification & Implication of Experiemtanl Results

Figure 1.3 Research Methods

1.8 Organisation of Thesis

Chapter 1 introduced the motivation for a context-aware traffic classification framework to accurately classify mobile applications that are more unpredictable in behaviours. The remaining chapters of the thesis are organized in the order of:

Chapter 2 synthesizes and present the literature review of existing traffic classification methods. This is written categorically based on research objectives; in the order of *data collection*; *feature engineering* and common *classifiers model* in traffic classification domain: Next, the proposed methodology on *context-aware application classification methods* in presented in Chapter 3. Firstly, a data sanitization method (SDA) developed to prepare clean training data is discussed. Secondly, the proposed feature learning methods based on *behavioral feature extraction* (BFE) is presented. Thirdly, a universal and scalable traffic classification model is presented as a *context-aware traffic classification model* (CAT). In Chapter 4, the proposed mechanisms are implemented and their respective algorithms are discussed. In Chapter 5, the proposed work is evaluated and compared against some other prominent traffic classification methods; in terms of classification *accuracy* and *diversity*. Experimental results are justified and corresponding implications are also discussed here. Lastly, some potential future works to address the limitations of the proposed methodology towards real world applicability is discussed in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

This chapter synthesizes some prevalent components in traffic classification from the perspective of mobile traffic classification. The chapter starts with critical analysis of some prominent traffic classification methods including statistical methods, deep packet inspection methods and fingerprinting methods. Next, some common feature selection techniques are discussed. Lastly, some popular machine learning algorithms including decision tree, regression and instance based classifiers are discussed.

2.1 Overview of Traffic Classification

Traffic classification is a machine learning technique for flows identification and labeling based on corresponding traffic classes (Nguyen & Armitage, 2008). The inputs to traffic classification are labelled IP flows. The resulting output is a traffic model trained using these IP flows using algorithmically selected features and classifier. The three major components that makes accurate traffic classification are prevalent *traffic flows,* highly correlated *features* and effective *classifiers* (Nguyen & Armitage, 2008; Anantavrasilp, 2009). Flow statistics (flow size, packet arrival time, flow duration) and applications fingerprint (payload signatures) are examples of network attributes capable to discriminate correlated traffic classes; these are called feature(s) in machine learning. Meanwhile, the classification process is performed using classifiers algorithm such as *Naïve-bayes, J48,* and *Adaboost.* Traffic classification is can be supervised (flows need to be labelled for training) or unsupervised as in clustering algorithms (Keralapura, Nucci, & Chuah, 2009). Figure 2.1 visualize the generic flows of traffic classification.

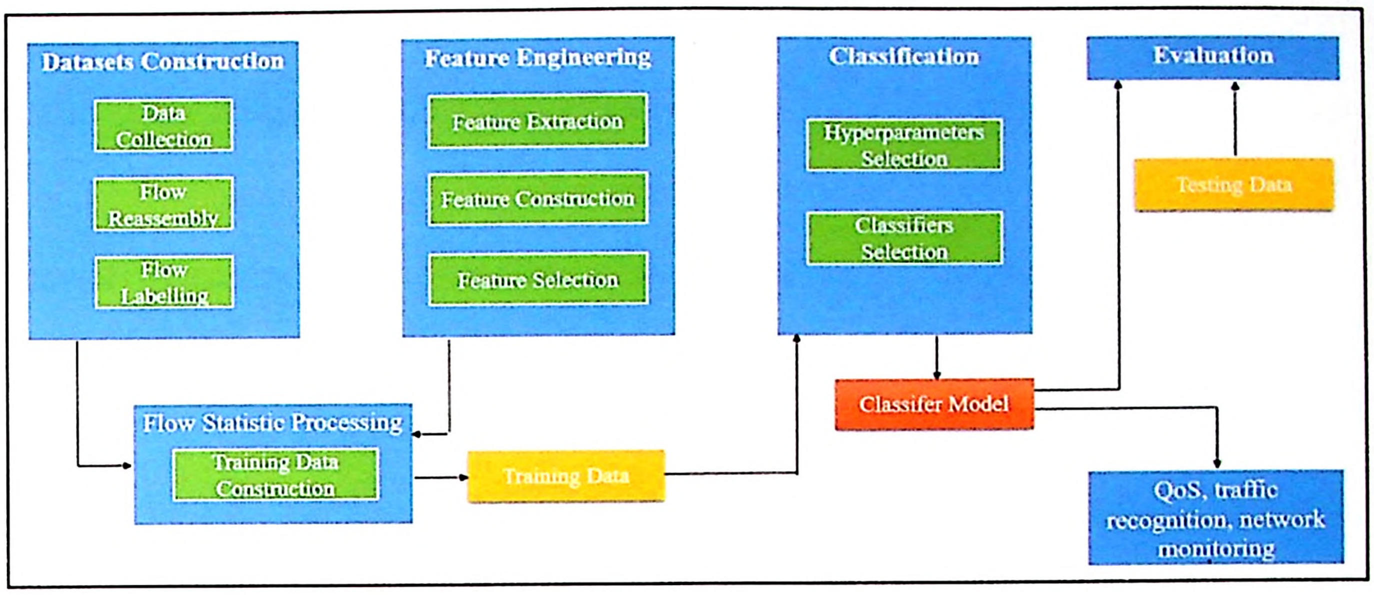


Figure 2.1 Generic Flow for Traffic Classification

2.2 Prevalent Traffic Classification Methods

Traffic classification has come a long way since its inception. It started with manual classification, evolved into to port-based classification and then statistical (Bayesian) & payload (pattern matching) classification (Nguyen & Armitage, 2008; Kurt, Cemgil, Mungan, Polat, & et al., 2012; Anantavrasilp, 2009; Puš, 2012) that are trending now. The types of features employed and scope of classification together defines the criterion that distinguishes traffic classification works. Figure 2.2 shows the taxonomy of existing traffic classification methods.

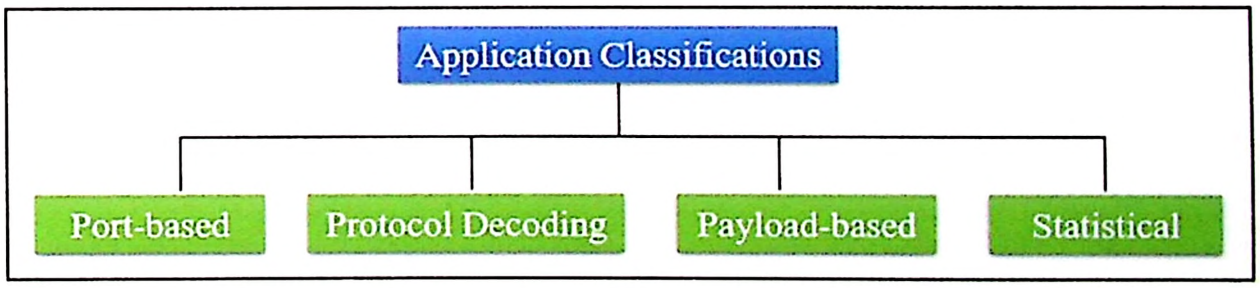


Figure 2.2 Category of Traffic Classification Methods

In early days, applications are classified manually. However, the enormous growth of traffic size introduced make manual classification infeasible. Port based methods are proposed to address this by detecting applications according to *IANA* allocated port numbers (Finsterbusch, Richter, Rocha, & Muller, 2012). The dynamic port usage in user applications however deter port effectiveness as to uniquely identify flows classes. In addition, single user applications typically opened many ports for concurrent sessions; thus, port number(s) is only effective on standard protocols detection (Liao, et al., 2016; Haffner, Sen, Spatscheck, & Wang, 2005). Port-based classification is also vulnerable to port spoofing attacks; therefore, spoofing detection mechanism are needed to ensure classification integrity if prior data are inferred on the port. Consequently, port numbers become an unreliable tracking mechanism that leads to high number of false positive(FP) and false negatives(FN) (Nguyen & Armitage, 2008) for machine learning.

2.2.1 Payload-Based Traffic Classification

Payload-based classifications are based on deep packet inspection (DPI) that derive signatures that are unique to individual application from the payload data in IP packets (Finsterbusch, Richter, Rocha, & Muller, 2012; Goo, Shim, Lee, & Kim, 2016). Payload signature is a subset of some deterministic portion of payload bytes used to infer traffic classes. The rationale is that certain payload bytes' sequence is unique to specific applications; which made them suitable as traffic descriptors (G. Ranjan,2016). DPI methods are highly accurate; however, they only work on unencrypted. Given the growing emphasis on traffic encryption; payload signatures are becoming less effective especially for mobile traffic classification (Kurt, Cemgil, Mungan, Polat, & et al., 2012).

One of the most prominent work in DPI is proposed by Moore (Moore & Papagiannaki, Toward the accurate identification of network applications, 2005); it achieved 79% of detection accuracy using unique first K-byte of flows payload. Since DPI is computation intensive, some popular works like *Nmap* (Hacker, 1989), *LASER* (Keralapura, Nucci, & Chuah, 2009) proposed using variable length payload as application signature to optimize the speed of detection to support real time detection. Further optimisations through manipulating the lexical correlations in payload data are proposed to reduce memory consumption are observed in works proposed by (Yang, 2013) and (Wang, Jiang, Tang, Liu, & Wang, 2011). Table 2.1 shows some example of payload signature for *BitTorrent*, *FTP*, *DNS*, *POP3* and *SSL.*

Table 2.1 Examples of Payload Signatures used in Traffic Classification

Applications	Signature
BitTorrent	"^\x13BitTorrent protocol\x00{5}\x10\x00\x05"(c&s) "^Get announce php?passkey==.*info hash=="(c)
FTP	"^(SIZE)|(TYPE.{2}331)|(PASS\x0D\x0A)|(USER.*\x0D\x0A)"(c)
DNS	"^.{2}\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00"(c)
POP3	"^+OK"(s) "^(QUIT|PASS|USER|(\x43\x41\x50\x41)).*\x0D\x0A"(c)
SSL	"^[\x14\x15\x16\x17][\x01\x02\x03]"(c&s)

The accuracy of DPI methods often has a strong start and eventually diminished when signature become outdated (Alok T, 2016); which is inevitable in mobile application context (Nguyen & Armitage, 2008; Finsterbusch, Richter, Rocha, & Muller, 2012) given that they are constantly being updated. Enormous efforts are required for signature maintenance to preserve classification accuracy; since it is difficult to determine the time-to-live of each signature (Sen, Spatscheck, & Wang, 2004; Haffner, Sen, Spatscheck, & Wang, 2005). In ACAS (Haffner, Sen, Spatscheck, & Wang, 2005), LASER (Rawlins & Gordon-Ross, 2012) and Content-Packet-Flow Signature (Goo, Shim, Lee, & Kim, 2016), the extracted trivial words are pruned from signatures since these words are common across multiple classes. Meanwhile, StriD^{2}FA (Wang, Jiang, Tang, Liu, & Wang, 2011) and VS-DFA (Hua, Song, & Lakshman, 2009) use certain normalization technique to remove irregular characters that are non-discriminative to optimize signature set. In addition, StriD^{2}FA and VS-DFA also support self-learning mechanism for automated signature extraction. Table 2.2 compares the key attributes of 5 prominent signature extraction methods.

Table 2.2 Comparing the Signature Extraction Methods in DPI Methods

Methods	**Signature Length (*k*)**	**Normalized**	**Non-Trivial**	**Learning**
ACAS (Haffner, Sen, Spatscheck, & Wang, 2005)	64	✗	✓	✗
StriD^{2}FA (Wang, Jiang, Tang, Liu, & Wang, 2011)	Adaptive	✓	✓	✓
VS-DFA (Hua, Song, & Lakshman, 2009)	Adaptive	✓	✗	✓
LASER (Rawlins & Gordon-Ross, 2012)	Adaptive (LCS)	✗	✓	✗
Content-Packet-Flow Signature (Goo, Shim, Lee, & Kim, 2016)	k=Fixed by classes	✓	✓	✗

Variable-Stride Multi-Pattern Matching Algorithm (VS-DFA) is classified using string matching using block-oriented pattern scheme over conventional byte-oriented pattern (Schleimer, Wilkerson, & Aiken, 2003) to optimize memory usage. It fares better due to the variable number of byte scanned in one pass using *Winnowing* algorithm. Next, the author uses an FA construction approach, a derivative of *Aho-Corasick DFA* algorithm for pattern extraction. The algorithm is benchmarked with

Snort & ClamAV pattern sets and proclaimed less memory usage for every 3 bytes of the character pattern. A two-stage self-learning traffic classifier (SLTC) is proposed to classify P2P traffic based on flow timing correlations (Keralapura, Nucci, & Chuah, 2009). The first stage use *Time Corrélation Matrix* (TCM) to distinguish P2P flows from mass traffic; High-Speed Monitors (HSM) is deployed to speed up payload signature-based classification. The second stage infers on connection patterns and direction to deduce P2P application for remaining unclassified flows. Classified *Rabin-Karp* with Binary search and Two-level hashing (CRKBT) (Lin, Lin, Lai, & Lee, 2008) is another string matching algorithm that emphasis on computational performance optimization while preserving classification accuracy and completeness. The author evaluates the enhanced C*RKBT* algorithm and evaluates the improvement on *ClamAV*, *DansGuardian,* and *Snort* package and claimed noticeable speed enhancement across different packages.

Length-Based Matching (LBM) with accelerating scheme for RegEx matching, a variation of RegEx matching that use enhanced *Dual-Finite Automata* (DFA) called *Stride-DFA (StriD^{2}FA)* is proposed to speed up pattern matching (Wang, Jiang, Tang, Liu, & Wang, 2011). The algorithm takes cues from data compression techniques, it first converts byte stream into integer stream in form of Stride-Length (SL) before feeding it to StriD^{2}FA. The novelty is mainly observed in improved computation speed and reduce memory consumption. However, the compression is a lossy process and some wrongly discarded data may result in erroneous classification. A lightweight DPI (Fernandes, et al., 2009) is proposed with the objective to minimize DPI induced performance bottleneck. The algorithm selectively processes random payload unit among all flows, and randomly read partial segments of individual payload to reduce operational cost. The author uses readily available signature from L7-filter (Keralapura, Nucci, & Chuah, 2009) that covers for 60 applications definitions. The author claims that computation overhead is significantly reduced without any accuracy trade-off. A simple pattern matching using first 2 Bytes of payload (Chung, Park, Won, & Strassner, 2009) with some inferred statistical vector is proposed for user-centric application classification. The algorithm works by checking on the similarity index of some baseline vectors against the extracted vector from unknown sequential flows to distinguish application classes. The scope of classification includes *BitTorrent*, *Emule*, *YouTube*, *Figure* and *Afreeca*. The author claims 98% accuracy can be achieved using just 2 bytes of data in small-scale classification; although the stability towards bigger scope and unknown flows are not accounted for. A generic content matching algorithm based on edit distance is proposed to evaluate payload structure effectively. *Content-based pattern matching* (Choi, Choi, Ha, & Ban, 2006) introduces the concept of sliding windows to adaptively shift the *k*-bytes length of payload data based on distance calculations to minimize signature size. *APSC system* (Yuan, Xue, & Dong, 2013; Dainotti, Donato, Pescape, Rossi, & et al., 2008) argues that temporal cues are useful for discriminating traffic class; it proposes using the sequence correlations of payload bytes instead of the payload itself for signatures construction. *Multi-level signature* (Goo, Shim, Lee, & Kim, 2016) solve the trivial problems in payload inference methods such as dictionary words and common strings using 3-level interpretation; on the packet, flow and connection level respectively. Lastly, *ClassifyDroid* (Dong, Guo, Li, Xu, & Wei, 2016) is a statistics-payload hybrid method developed for large-scale Android applications classification. The author claims that modern applications invoke multiple functions call and these behaviors are observable in *android_sdk* or *API* call patterns. Table 2.3 summarizes some of the prevalent payload based traffic classification methods.

Table 2.3 Deep-Packet Inspection based Traffic Classification Methods

Methods	Features	Scope	Accuracy (local)
Static application signature (Sen, Spatscheck, & Wang, 2004)	P2P specific signatures	5 types of P2P applications	up to 99.99%
ACAS (Haffner, Sen, Spatscheck, & Wang, 2005)	TCP/UDP header information	FTP, SMTP, POP3, IMAP, HTTPS, HTTP, SSH	up to 100%
Autograph (Kim & Karp, 2004)	Heuristics	HTTP worms	Not specified
Hamsa (Li, Sanghi, Chen, Kao, & Chavez, 2006)	Content based signatures	Worms: Code-red II, Apache-knacker, ATPhttpd, CLET, TAPiON	up to 100%
StriD^{2}FA (Wang, Jiang, Tang, Liu, & Wang, 2011)	String-matching	not explicitly stated	Not specified
VS-DFA (Hua, Song, & Lakshman, 2009)	Snort & ClamAV pattern	Snort & ClamAV supported attacks scope	Not specified
LW-DPI (Fernandes, et al., 2009)	Number of packets, payload length	60 supported applications in L7-filter: P2P, web, chat. NM, streaming, mail, VoIP	up to 99.95% for *P2P*
SLTC (Park, Won, Kim, & Hong, 2008)	Time Correlational Metrics + LASER (Keralapura, Nucci, & Chuah, 2009)	P2P application: Gnutella, E-Donkey, BitTorrent, Skype, Kazaa	up to 99.61% for *P2P*
LASER (Keralapura, Nucci, & Chuah, 2009)	Packet count, minimum substring length, packet size	Limewire, BitTorrent, Fileguri, Others	up to 90%
Bernaille Early Application Identification (Bernaille, Teixeira, & Salamatian, Early Application Identification, 2006) (Bernaille, Teixeira, Akodkenou, Soule, & Salamatian, 2006)	Signature from Traffic Designer + statistical stream properties	NNTP, POP3, SMTP, SSH, HTTPS, POP3S, HTTP, FTP, Edonkey, Kazaa	up to 99.9%
BLINC (Karagiannis, Papagiannaki, & Faloutsos, 2005)	Graphlets behavior signature + heuristics	The Web, p2p, data, network management, mail, news, chat, streaming, gaming, non-payload	up to 99.9%
SVM (Quoc, D'Alessandro, Park, Romano, & Fetzer, 2015)	Not specified	40 unspecified applications	Not specified
Content Matching (Choi, Choi, Ha, & Ban, 2006)	Layer 4 header data	Not specified	Not specified
HMM sequence (Yuan, Xue, & Dong, 2013)	Payload bytes' sequences	Bittorrent, eDonkey, QQ, SSL, FTP, DNS, POP3, Skype	up to 100%
'Elaborative' Payload (Goo, Shim, Lee, & Kim, 2016)	Payload bytes'	Youtube, Facebook	up to 100%
API Discrimination (Dong, Guo, Li, Xu, & Wei, 2016)	Android API string matching	200 Androids Applications	Not specified

In summary, this thesis defines three criteria of consideration for extracting good quality signature to address the gap in current literature; this includes:

- Signature length – optimal signature uses the least number of bytes required to discriminate applications accurately. *Longest common substring* (LCS) and *Smith-Waterman algorithm* are two popular technique to extract a sequence of payload bytes that are unique towards some specific classes. In some cases, signature extraction begins at a random offset position within payload data if they are better behaviour descriptors.
- Correlational factor (CF) –signatures are bytes' sequence that are unique to a single class but are less correlated to other classes. CF is directly influenced by signature length; optimal signature length give higher CF and higher CF give more accurate classification. CF is likely to reduce when the number of classes involved increases.
- Scalable/self-learning – a common problem with DPI is the apparent decline in traffic classification accuracy over time. This is due to frequent application updates and that application behavioural changes induced overt changes in corresponding signature set. Accuracy decline when an existing signature no longer accurately describe the new behaviours of corresponding applications. Good signatures shows consistent performance across multiple application versions; or is self-learning capable and opportunistically recalibrate itself to stay optimal using reinforcement learning techniques. One of the main challenge here is to automatically detect when is new signature set is needed for accurate classification.

2.2.2 Flow Statistical Based Traffic Classification

Flow statistics are widely used in traffic classification for encrypted traffic. Classifier model trained with statistical features is capable to distinguish traffic simply by inferring to packet headers or flow statistical properties. Flow features are universal for all communication based on TCP/IP, is comparably less resources intensive and allow faster detection since no payload processing is involved (Moore & Zuev, 2005; Roughan, Sen, Spatscheck, & Duffield, 2004). Table 2.4 shows some most commonly used flow features in traffic classification.

Table 2.4 Flow Statistics Features Commonly Used to Discriminate Traffic Flows

Features	Description
Packet length	The size of IP packets given by IP HDR_LEN and IP Total_LEN
Packet length statistics	Min, max, mean, median, SD of *packet length*
Inter-arrival time (IAT)	The differences of arrival time/delta time/timestamp among packets
IAT statistics	Min, max, mean, median, SD of *IAT*
Packet counts	Number of packets in a defined period/interval/quartile/sample size
Bytes counts	Numbers of bytes in a defined period/interval/quartile/sample size
Flow counts	Numbers of flows in a defined period/interval/quartile/sample size
Timing intervals	Idle time, keepalive time, arrival time, connection time statistics
Flow size	The number of packets in a flow (average)
Flow duration	Length of flows in average per session/interval
TCP port	The port number as read by network monitor (as identifier)
Payload size	Number of bytes of message payload excluding headers
TCP window size	The advertised TCP sliding window size
Burst size	The burst intensity of packet in a defined interval
TCP count with PUSH	Number of TCP packets with PUSH set to 1
Packet arrival order	The pattern of packets arrival based on sequence number
Effective bandwidth (entropy)	Network utilization parameters
First n-packet size	The total bytes count of sum (first n packets)
Bytes flow	Number of bytes transferred from client to server; server to client
IP Header Parameters	The aggregation of IP header attributes values
TCP Header Parameters	TCP usage characteristics as conveyed in TCP header parameters such as flags (SYN, PUSH, FIN, ACK) and window size etc.

The problem with flow features is that they are susceptible to network noise, since these attributes are mostly defined by network conditions. For example, segment size and packet size are dependent on MTU unit and MSS value defined by operating system.

Packet arrival time and order is sensitive to network delay and congestion control mechanism that are in place. In short, flow features are only effective in a simulated environment where the variability of network conditions can be controlled.

One of the early work in statistical classification proposed by (Piskac & Novotny, 2011) looks at packet & flow vectors based on *Vector Space Model* (VSM). The algorithm computes the associations among flows based on *Root-Mean-Square* (RMS) distance, *Euclidean* distance and the angle between the vectors and it achieves true positive of 90% with 7% false positives. *Multivariate Gaussian Fitting* of Multi-Scale Traffic Characteristics (Rocha, Salvador, & Nogueira, 2011) technique is developed to account for real-time traffic analysis using highly correlated features. The author defines classification scope into legitimate and illicit traffic and evaluating the work using attack simulations gives 91% of accuracy. A semi-supervised algorithm proposed by (Risso, Baldini, & Bonomi, 2007) is a good showcase of the capability of statistical ML classification. It employs 6 parameters to describe flows including *flow start and stop timestamp*, the *total number of bytes, the total number of packets*, *average packet size*, *average packet/byte rate* and cumulative *TCP flags* for each flow. The author defines the classification scope to 8 protocol types; that includes *Web*, *FTP*, *P2P*, *Streaming*, *Database*, *Mail*, *Instant Messaging*, *VPN,* and *VoIP* traffic. The classification result achieved the best case of 90% accuracy for the five of the targets, but the result is less desirable for the rest of the traffic classes. The scope the work is among some of the most complete, as it supports both standard protocols and user application traffic classes.

There are methods that work at sub-flow levels that claim to improve existing statistical flow based methods in real time environments. Such classifications methods