Network Traffic Classification Using Machine Learning: A Step Towards Over-the-Top Bypass Fraud Detection

BY: TEWODROS HAILU

ADVISER: EPHREM TESHALE (PHD)

A Thesis submitted to
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology

in Partial Fulfillment of the Requirements for the Degree of Master of Science in Telecommunication Engineering

Addis Ababa University
Addis Ababa, Ethiopia
November 14, 2018

Declaration

I, the undersigned, declare that the thesis comprises my own work in compliance with internationally accepted practices; I have fully acknowledged and referred all materials used in this thesis work.

Tewodros Hailu

Name

Signature

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

This is to certify that the thesis prepared by **Tewodros Hailu**, entitled *Network Traffic Classification Using Machine Learning: A Step Towards Over-the-Top Bypass Fraud Detection* and submitted in partial fulfillment of the requirements for the degree of Master of Science Telecommunication Engineering complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Internal Examiner	_____	Signature	_____	Date	_____
External Examiner	_____	Signature	_____	Date	_____
Adviser	Ephrem Teshale (PhD)	Signature	_____	Date	_____
Co-Adviser	_____	Signature	_____	Date	_____

_____

Dean, School of Electrical and Computer Engineering

ABSTRACT

Over-the-Top (OTT) bypass is a type of Interconnect Bypass fraud where regular voice calls are rerouted through OTT network and terminated as an OTT call. These calls are terminated using OTT applications which need user's Mobile Station International Subscriber Directory Number (MSISDN) for authentication. Detecting OTT voice call packets through different network traffic classification techniques is one subtask in the detection of this fraud.

In this thesis, performance of three machine learning algorithms; Adaptive Booster (AdaBoost) + J48, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), and Support Vector Machine (SVM) is evaluated in detecting MSISDN-based OTT packets taking Viber, Tango, and Telegram as a sample. Detection of OTT traffic and voice call packets from the OTT traffic have been treated separately as classification tasks. Ten cross-fold and separate test data validation techniques together with 1.7 million labeled packets generated and captured in controlled laboratory environment are used in the evaluation process.

AdaBoost + J48 achieved the best accuracy on both classification tasks compared to the others while using ten cross-fold validation. However, an accuracy of 48.4% obtained in detecting voice call packets while using separate test data validation makes it less preferable in the classification task. Even if it takes longer time to train SVM, it was the best performer (95.35% accurate) in detecting voice call packets in separate test data validation. Considering accuracy attained by the algorithms in separate test data validation technique together with the detection rate of OTT voice call packets, SVM is preferable than the other two algorithms.

KEYWORDS

OTT bypass, MSISDN-based OTT, Network traffic classification, and Machine learning.

ACKNOWLEDGMENTS

Frist I would like thank God for being a source of courage for the whole time. My special gratitude goes to my advisor Ephrem Teshale (PhD) for his tremendous and genuine advises throughout this work. Thank you Dr. Ephrem for all those valuable comments and suggestions you have provided. I would also like to thank my evaluators Dereje Hailegebreal (PhD) and Murad Ridwan (PhD) for the feedbacks during the thesis progress presentations.

I would also like to thank my special friend Paulos Girma for all the assists he provided throughout this work. Thanks Bro!

Last but not least, special thanks goes to my wife Axumawit Tassew and little angel Selina. Axum, thank you so much. Little angel selina, now we will have plenty of time to spend together and have funs.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AdaBoost	Adaptive Booster
AI	Artificial Intelligence
ANN	Artificial Neural Network
CDR	Call Detail Record
CFCA	Communications Fraud Control Survey
DPI	Deep Packet Inspection
DNS	Domain Name Server
DRS	Domestic Revenue Share
EM	Expectation Maximization
FMS	Fraud Management System
GMM	Gaussian Mixture Model
GSM	Global System for Mobile communications
GUI	Graphical User Interface
IANA	Internet Assigned Numbers Authority
ID_3	Iterative Dichotomiser 3
IP	Internet Protocol
IQR	Inter Quartile Range
IRSF	International Revenue Share Fraud
K-NN	K Nearest Neighbor
MLP	Multi Layer Perceptron

MSISDN	Mobile Station International Subscriber Directory Number
OTT	Over-the-Top
PBX	Private Branch Exchange
PRS	Premium Rate Service
QoS	Quality of Service
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
ROC	Receiver Operating Characteristic
SIP	Session Initiation Protocol
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SPI	Security Parameter Index
SPID	Statistical Protocol Identification
SSH	Secure Shell
SSL	Secure Sockets Layer
SSDP	Simple Service Discovery Protocol
SVM	Support Vector Machine
TCG	Test Call Generation
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VAS	Value Added Service
VoIP	Voice over IP
WEKA	Waikato Environment for Knowledge Analysis

1

INTRODUCTION

Telecommunication fraud is any activity of obtaining a telecom service with an intention of not paying service fee. Revenue loss and Quality of Service (QoS) degradation are among the common impacts of telecom frauds. International Revenue Share Fraud (IRSF), Domestic Revenue Share (DRS), Interconnect Bypass, Premium Rate Service (PRS), Reselling of Device/Hardware such as cables and routers, and Wholesale fraud are among the different categories of telecommunication frauds [1]. IRSF, DRS, and PRS frauds deal with abusing carrier interconnect agreements and inflating traffics to high cost call termination areas while Wholesale fraud is about exploiting wholesale agreements. Interconnect Bypass fraud is an unauthorized insertion of traffic to a telecom carrier's network.

According to Communications Fraud Control Survey (CFCA) 2017 report [1], Interconnect Bypass fraud is the second most reason for revenue loss next to IRSF. In 2017, an estimated revenue of $4.27 B has been lost by telecom companies due to Interconnect Bypass frauds [1]. ethio telecom, sole telecom service provider in Ethiopia, is among the telecom companies impacted by Interconnect Bypass frauds, mainly SIM box fraud. Hence, interconnect bypass frauds mostly target international call interconnection/termination fees, not only to ethio telecom but the country is also impacted because its loosing revenue in foreign currency.

Even if telecom service providers and regulatory bodies are deploying different Fraud Management System (FMS) to detect these frauds, they still remain a main challenge for service providers. The changing behavior of the frauds from time to time is among the reason for the frauds to remain as a main challenge. Detection techniques for these frauds needs to be continuously assessed and improved so that telecom companies and regulatory bodies can go in line with the changing behavior of the frauds for efficient and effective fraud Mitigation [2, 3].

Currently, ethio telecom is using a rule-based FMS for the purpose of detecting and preventing telecommunication frauds. Rules with a predefined threshold values are manually supplied to the system so that the system can act accordingly on new data based on the the threshold given. The FMS is designed to detect and prevent nearly sixty two fraud types including IRSF, PRS, and Interconnect Bypass frauds. Currently, due to system integration and storage related issues, the system is only applicable to forty eight fraud types.

Despite the commonly known Interconnect Bypass fraud, there exist a paradigm shift; called OTT bypass fraud [4]. Fraudsters need to avoid being detected and earn more income, existence of large OTT user base, and advancement in telecommunication technologies are among the motives for this paradigm shift. OTT bypass fraud is a recent type of telecom interconnect bypass fraud where a call initiated as non-OTT call is rerouted through OTT applications and received as an OTT call at the receiver side [4]. Due to this, the call termination fee which is supposed to be paid for the telecom operator at the receiving end will be shared by the OTT service provider and the transit operator involved in the fraud. This fraud is usually done using OTT applications like Viber which require user's MSISDN during service registration.

Due to the difficulty in detecting and quantifying the impact of this fraud, telecom companies are losing a huge amount of revenue even without knowing the existence of this fraud [4]. Communication between telecom operators together with an intense analysis of data obtained from test call, Call Detail Record (CDR), data traffic, and audio fingerprinting is required to minimize the impact of OTT bypass fraud.

Analyzing the data traffic to detect fraudulent OTT voice calls is one of the detection technique for OTT bypass fraud. Net neutrality, which states data traffic operators need to treat all contents of data traffic without any restrictions and payment discriminations, and the use of encryption and proprietary protocols by OTT applications are the challenges in using data traffic analysis as a detection technique.

Detection of OTT packets from the data traffic using network traffic classification techniques is one of the steps in detecting fraudulent OTT calls. Network traffic

classification which is the process of identifying network applications or protocols that exist in a network, is usually done using port-based, payload-based, and machine learning-based techniques [5]. Among these techniques, machine learning-based techniques are less affected by encryption and use of proprietary protocols, which is one of the challenge in data traffic analysis.

This research focus on the use of machine learning algorithms in detecting MSISDN-based OTT voice call packets. The performance of three supervised machine learning algorithms is analyzed in detecting these packets taking three popular OTT applications; Viber, Tango, and Telegram.

1.1 STATEMENT OF THE PROBLEM

Telecom service provider's operations and revenues are highly impacted due to telecom frauds. As mentioned in the previous section, large amount of revenue is lost due to interconnect bypass frauds. OTT bypass fraud is one type of interconnect bypass fraud usually done with OTT services such as Viber which integrate the MSISDN of a user with the application during registration. It is a real threat for telecom companies due to the large user base OTT service providers have. As per our knowledge, there is no detection module for this fraud in the current ethio telecom FMS. Like the other frauds, OTT bypass threat also needs to be mitigated since the company is losing international call termination fee which is supposed to be paid in foreign currency.

Analyzing data traffic and detecting OTT voice traffic through different classification techniques is one step to mitigate this fraud. Data transport layer Port based and payload based traffic classification techniques which works well for non-encrypted traffic and for applications which use standard port are not efficient when the traffic is encrypted and dynamic port numbers used compared to machine learning techniques [6–8]. Most OTT services such as Viber and Tango use encrypted traffic and proprietary protocols for communication [9].

This research will focus on classifying a network traffic to detect OTT voice call traffic using machine learning as a classification technique and taking three MSISDN-based OTT services; Viber, Tango, and Telegram as a sample. In this research, per-

formance of machine learning algorithms in detecting OTT voice call traffic will be tested.

1.2 OBJECTIVE

1.2.1 *General Objective*

The main objective of the research is to evaluate the performance of three machine learning algorithms; AdaBoost + J48, RIPPER, and SVM in detecting MSISDN-based OTT voice call packets for the purpose of OTT bypass fraud detection.

1.2.2 *Specific Objectives*

The specific objectives of this thesis are:

- Explore different machine learning based network traffic classification algorithms and select three algorithms for implementation.
- Identify list of MSISDN-based OTT applications and select three of them as a sample.
- Generate and capture a network traffic consisting of MSISDN-based OTT and other data traffic, separately.
- Explore and select the best network packet attributes which can be used to detect OTT packets.
- Classify the captured traffic using the selected algorithms and packet attributes.
- Analyze the performance of the implemented classification algorithms.
- Deduce conclusion and recommendations based on the classification performance analysis.

1.3 SCOPE AND LIMITATIONS

The scope of this thesis is limited to generating, capturing, and detecting MSISDN-based OTT packets on a controlled laboratory environment. The laboratory environment is composed of network components (Router, Switch, and Wi-Fi AP) with an EPON connection and four users which are used for generating and capturing the data traffic. However, we are confident the generated traffic will model actual traffic on ethio telecom network with reasonable accuracy, and this factor does not significantly limit applicability of the research.

The packet detection process is done in two phases; first detecting MSISDN-based OTT packets from the given data traffic and then detecting voice call packets from the OTT packets.

- Only Viber, Tango, and Telegram have been taken as sample MSISDN-based OTT applications. Packets from all these three applications have been used in the first phase while only packets from Telegram have been used in the second phase.
- In the frist phase, packet detection is not done per application rather packets are classified either as "MSISDN-based OTT" or "Other" packets. Packets from YouTube, Facebook, Skype, Gmail, and Yahoo Messenger has been considered under the "Other" category.
- The second classification process categorizes Telegram packets as either "Voice call" or "Non-Voice call" packets. Other Telegram service flows such as "Video call", "Text message", and "Voice messages" are not considered as classification class labels in this research.

1.4 CONTRIBUTIONS OF THE RESEARCH

As per our knowledge, there is no specific work done on the detection of MSISDN-based OTT applications using machine learning algorithms on network traffic data generated in a controlled laboratory environment. The output of this research will serve as an input to:

- Network administrators to manage the network and improve QoS by prioritizing applications running on the network.
- Anti-fraud personnel for implementing security policies related to OTT bypass fraud.
- Policy makers to understand the ratio of OTT applications bandwidth usage and design policies accordingly.
- Researchers who want to engage in OTT traffic related classification tasks since they can use the labeled OTT network traffic data generated and used in this research.

1.5 LITERATURE REVIEW

A number of researches have been conducted in implementing machine learning algorithms and other techniques for the case of network traffic classification. Among this researches, related works which focus on OTT traffic classification tasks are mentioned in this section.

1.5.1 Machine learning based network traffic classification

Machine learning based classification tasks focus on analyzing the packet behavior instead of the payload, they are hot research areas recently [6, 7, 10]. Hence machine learning algorithms are not tested in detecting MSISDN-based OTT packets previously, researches which are done on other OTT applications; Skype, GTalk, Google Hangout, and Yahoo Messenger have been discussed in this subsection.

Al-Naymat et al. [10] compared the performance of different machine learning algorithms in classifying and detecting three Voice over IP (VoIP) and two non-VoIP applications; Skype, YouTube, GTalk, Yahoo Messenger, and Paypal. The machine learning algorithms used for classification are AdaBoost, Random forest, J48, and Multi Layer Perceptron (MLP). They designed a test bed environment composed of computers installed with these applications and a router to simulate a real network traffic. Wireshark tool installed on a separate computer from the applications is used to capture the generated traffic while Waikato Environment for

Knowledge Analysis (WEKA) tool have been used to implement the machine learning algorithms.

Four packet features; namely, packet length, delta time (packet inter-arrival time), cumulative byte, and relative time; are used for classification claiming that these features have been used for the first time in internet traffic classification. The scholars have stated that AdaBoost achieves the best overall accuracy (98.3 %) while MLP is the least performer with an accuracy of 84.2 % in classifying the five applications. From the test results reported, AdaBoost is also the best classifier in detecting VoIP applications specifically compared to the others.

On other research conducted on Google Hangout, Datta et al. [6] used Naive Bayes, J48 decision tree and AdaBoost machine learning techniques to detect Google Hangout traffic. They used a dataset consisting of 2.5 million packets which are generated as a separate traffic for the purpose of training; 1, 984, 954 and 689,025 packets for non-Google Hangout and Google Hangout traffic, respectively. Wireshark and WEKA tools are used to collect and analyze the dataset, respectively. Packet length, protocol, source and destination port number, packet type, and Domain Name Server (DNS) reply from Google are used as packet features for the classification.

The researchers [6] evaluated performance of the three machine learning algorithms in terms of recall using 10 cross-fold validation. They conducted the evaluation using three techniques: two class (Google Hangout / non-Google Hangout), three class (Google Hangout / non-Google Hangout / Gmail), and four class classification (Google Hangout / Google Plus / Gmail / others). For the experiments done, the scholars stated that AdaBoost is the best performer in all of the 3 techniques with a recall of 99.99%, 99.99%, 100% for two, three, and four class classification, respectively.

Alshammari et al. [7] have also proposed a classification model based on machine learning algorithms in an attempt to classify encrypted network traffic as Secure Shell (SSH)/non-SSH and Skype/non-Skype. They compared the robustness of AdaBoost, SVM, Naïve Bayesian, RIPPER, and C4.5 algorithms in classifying the encrypted traffic without using IP address, source/destination port number, and payload information. The scholars used different training and testing datasets to

test the robustness while using Net Mate and WEKA tools with default parameters for feature extraction and analysis respectively. They stated that C4.5 achieved a better classification accuracy (83.7% detection rate) followed by RIPPER in classifying both SSH and Skype. The researchers stated that the proposed classification model can also be used to classify other encrypted applications.

The use of combination of publicly available datasets (AMP, MAWI, and Dalhousie trace) together with simulated traffic dataset (DARPA99) makes their work unique from previously mentioned related works [6, 10] which instead use only simulated traffic dataset for the purpose of classification. Usage of training dataset from one network and using test dataset from another network to test the robustness of the proposed model, is also another unique feature from this work compared to other related works.

From the above three works which are based on machine learning algorithms [6, 7, 10], it can be concluded that factors such as type of classification algorithm used, traffic attributes/features selected for training the classification algorithm, the labeling and source of training data, and the parameter tuning in the machine learning tools used has an impact on the classification performance. Network traffic classification model which is designed considering these factors is expected to achieve better classification accuracy.

1.5.2 Non-Machine learning based network traffic classification

In this subsection, research works which propose different classification techniques other than machine learning algorithms have been discussed. Network traffic from OTT applications such as Viber, Skype, GTalk, and Yahoo Messenger is used for implementation.

Sudozai et al. [11] proposed a framework to identify Viber traffic from an Internet traffic and classify its services as audio calls, video calls, text chats, voice chats, group messages, and file sharing. They designed a simulated environment consisting of wireless router, layer 3 switch, and single mobile phone. They captured the Internet traffic using Wireshark packet analyzer which is configured to the layer 3 switch and used Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) port-based filtering techniques to detect viber traffic from other traffic.

For instance, the researchers claimed Viber usually uses UDP port numbers 7985, 7987, 5243, and 9785 for voice service while TCP ports 5242 and 4244 are usually used for message chats.

Once the detection of Viber traffic is completed, they used different behavior analysis methods such as payload size, byte patterns, and peculiarities in client-server/server-client responses for classifying Viber traffic to its service flows. For instance, the scholars claimed payload sizes between 18 and 350 bytes are assumed to be Viber call with 100% confidence while payload between 750 and 1400 bytes is considered as video call with a confidence of 50-70%. Even if the researchers claimed the proposed framework is reliable and works for the latest Viber version (August 2016), the evaluation process and result of the proposed framework is not included in the paper. Taking to account the use of dynamic port and port masquerading by OTT applications, it would have been better if the accuracy of this port based viber detection framework known.

Statistical Protocol Identification (SPID) which analyze statistical values of some traffic attributes and application layer data features is also used in detecting and categorizing encrypted TCP Skype traffic as voice, video, chat, file upload, and file download [12].They used only payload features to detect Skype traffic from data traffic while using combination flow and payload features to classify the Skype traffic to specific service flows. The scholars have further classified the detected file sharing and voice/video service flows using only flow features. Packet size, byte frequency, byte re-occurring, direction change, and byte offset are among the flow features used for classification.

The researchers in [12] generated a data traffic which consisted of both TCP Skype traffic and non-Skype traffic separately for the purpose of designing and evaluating performance of the classification model. While generating the Skype traffic, they have considered different Skype versions, operating systems, and wired/wireless connections. Experiment on the dataset show that proposed framework achieves precision of 100% and 92.75% for detecting and classifying the Skype traffic respectively while achieves lower precision in classifying service flows as voice and video (64.2%). The proposed classification model by the researchers is not generic, it is only applicable in detecting Skype traffic.

Rathore et al. [13] proposed an algorithm which generate rules based on statistical analysis of traffic features to classify network traffic as VoIP and Non-VoIP. The algorithm uses source IP address, destination IP address, and Security Parameter Index (SPI) features for IPsec traffic while it uses source /destination IP address and source / destination port number features for other type of traffic. The researchers collected DNS, frame relay, and VoIP applications such as Skype, GTalk, and Yahoo Messenger data from home computers, university laboratory, two telecom authorities' gateway, and Skype test data repository.

The researchers [13] claimed classification accuracy of 97.54% and 97.78% on an overall test data and Skype dataset specifically respectively. They have stated that the proposed algorithm is generic to all VoIP applications and used for real-time detection in high speed networks. However, the detection accuracy of the algorithm per VoIP applications is not mentioned in the paper while most VoIP applications use proprietary protocols and encryption techniques to avoid detection.

From the work which is based on application signature [11], it can be concluded that it is difficult to detect application using signatures since application signatures are dynamically updated by application/service providers. Even if it is reported that 100% precision is achieved while using payload based approach used initially in detecting Skype traffic [12], issues can be raised on the reported accuracy considering encryption challenge poised on payload based classification; which is applicable on most OTT applications including Skype. Compared to the proposed models in [11, 12], proposed algorithm in [13] is claimed to be generic to all VoIP applications.

From both category of related works mentioned in this section, it can be seen that type of algorithms, dataset, and attributes/features used in the process have significant impact on classification accuracy. In addition that, the existing publicly available training datasets doesn't consider MSISDN-based OTT applications, which are main players of OTT bypass fraud. As per our knowledge, the algorithms are also not tested in detecting MSISDN-based OTT traffics. Considering these factors, selected machine learning algorithms have been tested in detecting MSISDN-based OTT packets using manually labeled training dataset.

1.6 METHODOLOGY

The methodology used in this research include:

- Conduct extensive literature review to identify and select MSISDN-based OTT applications, explore different machine learning algorithms, and explore different network packet attributes.
- Generate a network traffic data in a controlled laboratory environment and capture the traffic using Wireshark; an open source network traffic capturing tool.
- Use WEKA workbench as tool to train the machine learning algorithms and analyze performance of the classification algorithms using confusion matrix, accuracy, F-Measure, and Receiver Operating Characteristic (ROC) curve as evaluation metrics.

1.7 THESIS ORGANIZATION

In this thesis, performance of machine learning algorithms is evaluated in detecting MSISDN-based OTT packets from a given data traffic. The detailed outline for the remaining part of the thesis is given below:

- Chapter two discusses about OTT bypass fraud by demonstrating the fraud scenario and detailing the different detection and prevention techniques proposed for mitigating the fraud. Different network traffic classification techniques are also discussed in the Chapter.
- Another subject matter of the research; machine learning, has been discussed in Chapter three. In this Chapter, the application areas together with categories of machine learning algorithms are stated.
- Chapter four details the experimental analysis used in this research. The tasks done under each sub module of the system model; network traffic generation and capturing, data preprocessing and feature selection, training, and evaluating the algorithms, are also discussed separately in the chapter.

- In Chapter five, the results obtained from the performance evaluation of the algorithms are presented and discussed.
- Finally, conclusion and future research works are stated in Chapter six.

2

OTT BYPASS FRAUD

2.1 INTRODUCTION

Interconnect Bypass fraud is the act of fraudulently finding unintended entrance (grey routes) to telecommunication service provider's network avoiding the legal interconnect gateways. The fraudsters use the infrastructure deployed by the telecom operator illegally without paying the necessary payment which result in revenue loss to the telecom operator. This is usually associated with an economics term "free rider" problem [14], which states enjoying the benefits of deployed infrastructure without paying to support it.

The high interconnection fee demanded by telecom service providers for terminating international call is the main enabler for this fraud. Even if different mitigation techniques are proposed and implemented to prevent interconnect bypass frauds, they still remain a challenge for telecom operators.

The common type of interconnect bypass fraud is SIM boxing, in which the bypass is done by connecting incoming VoIP calls to the telecom operator's cellular network using VoIP Global System for Mobile communications (GSM) gateways. A collection of SIM cards and cellular radios are used to make international calls terminated as a local call [14]. SIM boxing have negative impacts to customers and telecom operators such as reducing availability and reliability. It also reduce QoS rates of legitimate customers since it injects a huge number of simultaneous audio calls to the provisioned cells which are designed to handle provisioned number of calls. The other type of interconnect bypass fraud is done using Compromised Private Branch Exchange (PBX) to terminate international calls as local calls [4].

In addition to the above mentioned bypass frauds, there exist a recent type of interconnect bypass fraud; OTT bypass fraud. OTT bypass fraud also referred as OTT hijack is a recent type of telephony interconnect bypass fraud where a phone

call initialized as regular telephone call from a caller is terminated on the OTT applications installed on the device of the user at the receiving end [4]. This call rerouting is done by telephone transit operators in coordination with the OTT service provider, without the consent of the caller and telephone service provider at the receiver side [4]. OTT bypass fraud scenario is shown in Figure 2.1.

OTT services work on top of data links without the control of internet service providers to deliver voice, text, and video contents using packet switching technique [15]. Viber, Skype, WhatsApp, and Google messenger are among the different applications which are used to deliver OTT service contents; which are provided by different OTT service providers. Most features of OTT services are free of charge and the service provider's revenue are based on advertisements, in-app purchases, and subscription charges [16].

Currently, OTT services have a large user base around the world mainly due to the increased adoption of smartphones and availability of fast mobile internet access [4]. Due to the user friendliness, content flexibility, and usage charge of the OTT applications, users are preferring OTT services over the traditional carriers. This brings a significant threat to the traditional telecommunication service provider's revenue loss related to the traditional voice, Short Message Service (SMS), and multimedia services, in addition to the data traffic congestion the OTT services bring on the operators network; compared to the revenue gain from the data usage.

Due to the threats poised from this OTT services, telecom operators and regulators are taking actions such as blocking or slowing down the speed of OTT traffic depending on the network neutrality policies they adopt, launching packaged services (bundling OTT with their services), and developing their own OTT applications [16] to minimize the impacts.

Some OTT applications require user account for registrations while others need user's MSISDN for service registration [9]. Skype and Yahoo messenger are among the common OTT which need user account while applications such as Viber and Tango need user's MSISDN for registration. OTT bypass fraud is usually done with OTT applications which need user's MSISDN during service registration.

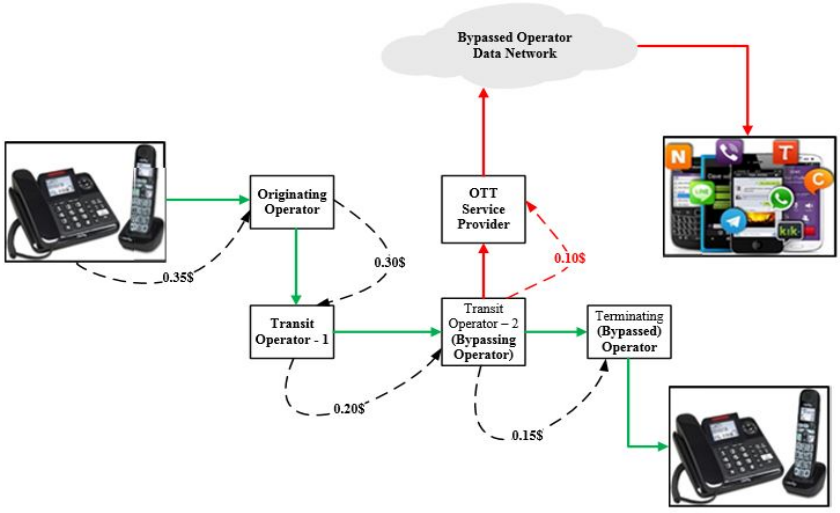


Figure 2.1: OTT bypass fraud scenario [4].

As shown in Figure 2.1, the caller initiates a regular call paying a certain fee to the originating operator assuming that his/her calls will pass through the normal telephony call route. But, the calls are rerouted to OTT network (shown in red line) instead of the regular cellular network (shown in green) with the help of a transit operator. OTT service providers will check whether the receiver is on-line or off-line using MSISDN and terminate the calls on the OTT applications at receiver end if the customer is on-line. The bypassed terminating operator will lose the call termination fee since the calls are not coming through the normal telephony route. In the mean time, OTT service provider and the transit operator involved in the call rerouting share the termination fee of the call. In this fraud, the terminating operator will only receive data usage fee from the receiver.

In addition to the revenue loss to the telecom operators, impacts of OTT bypass fraud include QoS degradation, charging of the receiver for the data service used, and inaccessibility of Value Added Service (VAS) such as voice mail and call forwarding [4]. Hence the calls are delivered as OTT calls, the caller will not have access to these VAS configured at the receiver end.

2.2 MITIGATION TECHNIQUES

There are different techniques that are proposed and implemented for the detection and prevention of OTT bypass fraud. However, they are not fully fledged by

themselves as stated in [4]. Cooperation and communication between telecommunication service providers is strongly recommended together with the following techniques.

2.2.1 *Test Call Analysis*

Making international test calls using different commercially available Test Call Generation (TCG) platforms is also another technique in detecting OTT bypass fraud. Telecom operators can make test calls to phone numbers available in their network from another operator's network using these platforms, and can understand whether the calls made are bypassed or not. By making repetitive calls, the operators can identify the transit operators involved in redirecting the calls to OTT network . Once the fraudulent transit operators are identified, corrective actions can be taken by the bypassed telecom operator.

TCG platforms are also used for QoS assessment and available with a cost of $1-10 per test call. However, they are not designed for specifically detecting OTT bypass fraud. The bypassed calls identified in test calls may not be only from OTT bypass fraud but also from other Interconnect Bypass frauds. In a recent study, [4] have conducted 15,872 test calls in European country operator using two TCG platforms and observe 83% of the calls are subject to OTT bypass fraud.

2.2.2 *Network Traffic Analysis*

Analyzing the data traffic using Deep Packet Inspection (DPI) tools is also used to detect OTT bypass fraud. However, this method have some challenges. The first is network neutrality issue where operators are expected to treat all data traffic the same, irrespective of the content [15]. The second challenge is the incapability of DPI tools to capture only OTT bypass traffic. The different encryption methods and proprietary protocols used by the OTT service providers are the other challenges of network traffic analysis.

2.2.3 *CDR Analysis*

Analyzing international incoming CDR and observing a decrease in received international call will indicate there is some interconnect bypass fraud. However,

this alone does not indicate there is OTT bypass fraud, therefore integrating this analysis with tracking the on-line/off-line status of users on the OTT application will help to detect bypass traffic. OTT providers will decide whether to terminate the calls on the OTT network or not after cross checking user's status on the applications using address matching techniques. If the user is on-line, the call is terminated on the OTT network otherwise not.

With regard to CDR analysis, [17] proposed fraud detection framework on Session Initiation Protocol (SIP) based VoIP networks analyzing SIP CDR. The proposed framework consisted of three modules; parsing module where data preprocessing is done, training module where Bayesian belief network is used for training, and detection module; to detect common attacks on SIP protocol such as Call tracking, Call hijacking, SPAM over Internet telephony, and Caller-id spoofing.

2.2.4 *Audio Fingerprinting*

Calls terminated on OTT network are believed to have some distortion or particular fingerprints on the audio channel compared to calls terminated on the normal telephony network. Therefore, the bypassed operator can fingerprint calls that pass through its network and identify those fingerprints on the audio channel and detect OTT bypass fraud. Audio fingerprnting technique have been tested in detecting the other Interconnect Bypass fraud, SIM box fraud [14] and an accuracy of 87% have been reproted. The behaviors of calls recorderd while using the traditional GSM cellular and SIM box calls have been compared for the fraud detection. However, this method is expensive since a lot of resources required to fingerprint all audio calls in the channel.

2.2.5 *Commercial Solutions*

There are commercially available solutions to detect and block OTT bypass fraud offered by different companies such as ARAXXE, Purgefraud, and Sigos. These commercial solutions are usually believed to use the concept of test call for detection and DPI to block the traffic.

Among the above mentioned mitigation techniques, this research focuses on data traffic analysis. In analyzing the data, first task is to detect MSISDN-based OTT voice

call packets from whole data traffic. The different classification techniques used to classify and detect network packets are discussed in the next section.

2.3 NETWORK TRAFFIC CLASSIFICATION

Network traffic classification plays a crucial role for network operators to identify applications and protocols running on a network for the purpose of network planning, application prioritizing for QoS, and security policy deployment. There exist different network classification techniques which are classified under three categories; Port-based, Payload-based, and Statistical-based classification approaches [5, 18].

2.3.1 *Port-based Techniques*

It is a classification technique which is based on the TCP or UDP port numbers of applications at the transport layer. It works well for applications which use standard ports assigned by Internet Assigned Numbers Authority (IANA). Classification speed, resource consumption, and not compromising user's privacy are among the advantages of port-based classification technique. However, its incapability in detecting applications which use dynamic port numbers remains the main drawback of this technique. Applications can avoid detection by port-based classification through port masquerading; taking port numbers of other applications [19].

2.3.2 *Payload-based Techniques*

It is also refereed as DPI technique and is based on inspecting the actual payload content instead of checking port numbers. It compares the payload content with a predefined feature set and claimed to have slow classification speed and consumes higher processing power [19]. Besides violating the user privacy, it lacks support for many applications/services due to the use of encryption and frequent signature change. PACE, OpenDPI, NDPI, Libprotoident, and Cisco NBAR are among the popular open-source DPI tools [19].

2.3.3 *Statistical-based Techniques*

This technique is also referred as machine learning or flow-based technique due to the use of machine learning algorithms and packet flow features during classification. It is based on analysis of payload independent traffic attributes such as byte frequencies, packet length, and packets inter-arrival time. classification is done based on analyzing protocol communication patterns instead of payload content. Due to this, statistical-based techniques are recommended classification techniques for encrypted traffic since they don't need to know details of the encryption protocol used.

Statistical based classification techniques gives promising accuracy results compared to port-based and payload-based techniques. This is due to their capability to perform the traffic classification independent of port number and encryption technique used. In addition to accuracy, faster computation time compared to port-based technique and preservation of data privacy are the advantages of statistical-based classification techniques.

Machine learning algorithms are usually used to analyze the packet features used in this classification technique. A set of training data labeled with a pre-defined classification class labels is used to train the algorithms for classification. The algorithms will associate packet feature values with the given class label during training phase. The feature - class label association is done using different metrics depending on the type of algorithms used. The different types of machine learning algorithms together with metrics used in the association are discussed in the next Chapter. After the training is done, the algorithms are capable of predicting class labels for new traffic instances.

In addition to the research works mentioned in Section 1.5, machine learning algorithms have been applied in different network traffic classfication tasks. SVM, C4.5 decision tree, and Naive Bayes are among the machine algorithms implemented in the detection of packets from World Wide Web (WWW), DNS, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), DNS, Hypertext Transfer Protocol (HTTP), Peer to Pear (P2P), and telnet applications [5, 20, 21]. Hence, the methodol-

ogy of this research is machine learning, the details of these and other algorithms is given in the next Chapter.

3

MACHINE LEARNING ALGORITHMS

3.1 INTRODUCTION

Machine learning is the process of gaining knowledge or expertise from a prior experience, and considered as subset of the wider Artificial Intelligence (AI) discipline [22, 23]. Machine learning algorithms take example data as an input and provide some expertise in the form of computer program that can perform some task as an output. The inputs are provided as concepts, instances, and attributes while the outputs are displayed using different knowledge representation styles such as decision tree, decision table, classification rule, association rule, or clusters [24]. Incorporation of prior knowledge is inevitable for the success of machine learning algorithms [22].

The need for machine learning arises due to different reasons. The need to perform complex tasks, both currently performed by human beings and beyond the capabilities of human beings, is the main reason why machine learning is required. Driving and face/speech recognition are among the tasks performed by human beings while analysis of big and complex datasets are tasks which are beyond human capabilities [22]. In addition to these complex tasks, the adaptive nature of machine learning algorithms to environmental changes (the environment they interact with) makes them preferable compared to other programming codes. Fraud detection from transaction records and prediction are among the tasks where machine learning is preferred instead of other programing codes due to its adaptive nature.

Machine learning is applied in different domains. The common application areas of machine learning algorithms include [24, 25]:

- **Web Page Ranking :** It is a scenario where search engines return web pages in precedence of relevance based on user's query.

- **Collaborative Filtering:** It is a situation where systems predict user's need without the need for users to have explicit query. This query prediction is done using users previous experience. Good examples for collaborative filtering can be on-line book stores recommending users to buy additional books, and query prediction in Google search.

- **Automatic Document Translation:** Manual translation of documents from one language to other is tedious and error prone task. However, machine learning can be used for this task just by training the algorithms providing language translation examples.

- **Named Entity Recognition:** Is a process of identifying entities, such as places, titles, names, and actions, from a given document. Good example for this task is Apple's mail application where addresses are extracted from email and filled to the address book.

- **Telecommunication:** Machine learning algorithms are applied in different tasks such as failure prediction of telecom Business Support System (BSS) [26], detecting telecom frauds [3, 27–29], designing customer churn prediction model where satisfaction level of a customer is analyzed and potential customers who may leave for another telecom service providers are identified [30–32], and intrusion detection systems where detection of anomalous traffics is done [33].

Classification is one of the tasks done using machine learning algorithms. It uses a set of classified/labeled examples for learning so that new instances are classified in to a set of predefined classes. Association, clustering, and numeric prediction are the other tasks performed using machine learning algorithms [24]. Association deals with learning association among the different features/attributes of a data while in clustering group of examples that belong together are clustered. The main difference between classification and clustering is a certain predefined class is predicted in classification while no predefined class will be predicted in clustering. Numeric prediction deals with predicting a numeric quantity instead of discrete class. The focus of this research is on the use of machine learning algorithms for classification task.

There are a number of machine learning algorithms mainly classified as supervised and unsupervised algorithms. Supervised machine learning algorithms need a set of labeled examples as a training dataset to construct a classification model. Unlike the supervised ones, unsupervised algorithms doesn't need labeled examples and usually used in clustering tasks instead of classification. Details of common supervised and unsupervised machine learning algorithms are discussed in the next sections.

3.2 SUPERVISED ALGORITHMS

In supervised machine learning, a labeled training data or examples are needed to train the algorithms. Once learning is done, the built models can predict class labels of new data/instances which are not used in learning phase. Naïve Bayes, SVM, Decision Trees, Neural Network, and Nearest Neighbor algorithms are among the common algorithms categorized under supervised machine learning algorithms [24, 25, 34]. Each of these algorithms are discussed separately here.

3.2.1 *Naïve Bayes*

Naive Bayes algorithm is considered probabilistic classifier, hence, it is based on Bayes probability theorem. It's output is the posterior probability of being a certain class label given the instance features/attributes as denoted in Eq. (3.1) [23].

The algorithm will calculate the probability of each class label given an instance features, and a class label with the highest probability value will be assigned to the instance. $P(x|w_c)$ and $P(w_c)$ are calculated from the training data while $P(x)$ will be normalized since it's value is the same for each class label.

Naïve Bayes classifier is preferable when there is a dataset which has small size and non-correlated/independent attributes. The time it takes to calculate posterior probabilities becomes higher when size of the dataset is increased. The algorithm also works well when the data type of attributes in the dataset is a combination of numeric and text.

$$P(w_c|x) = \frac{P(x|w_c)P(w_c)}{P(x)} \tag{3.1}$$

Where, $P(w_c|x)$ Posterior probability of being a class label "w_c" given the attribute value "x"

$P(x|w_c)$ Probability of the attribute value "x" when the given class label is "w_c"

$P(w_c)$ Prior probability of the class label "w_c"

$P(x)$ Prior probability of the attribute value "x"

Robustness and low computing cost are considered as advantages of Naïve Bayes algorithm. However, it's low performance makes it less preferable in the task of classification [35]. The performance achieved by other algorithms such as SVM is higher than that of Naïve Bayes.

3.2.2 *SVM*

Unlike Naïve Bayes algorithm, SVM is a non - probabilistic classifier. It maps input training data in to 'n' dimensional space as points and draws 'n-1' dimensional hyperplane to separate the data points in to groups. SVM draws a linear hyperplane using Eq. (3.2) with a target of maximizing the distance between support vectors and the hyperplane [34]. Support vectors are data points or instances which are closer to the hyperplane. Given two predefined class labels C_1 and C_2, instances are mapped to C_1 if $S(x) > 0$ and C_2 if $S(x) < 0$.

$$S(x) = w^T x + b \tag{3.2}$$

Where, $S(x)$ Linearly discriminant function

x Feature vector selected for classification

w Weighting vector which is orthogonal to the hyperlane and controls its direction

b Bias which control position of the hyperplane.

SVM is used for both binary and multi class classification tasks. When the inputs on feature space are not linearly separable, SVM use kernel trick to separate inputs [34]. Kernel trick maps a given input to a higher dimension feature space. SVM

can also be used for regression and clustering tasks. In regression, the class label to be predicted is continuous instead of discrete value.

3.2.3 *Decision Tree*

Decision tree is a hierarchical data structure which can be used for classification and regression tasks by implementing divide and conquer strategy [36]. It is composed of decision and leaf nodes in which each decision node implements a function to label the branches with discrete outcomes. Entropy and information gain are statistical measures which are used to construct the tree in decision tree algorithm. Entropy deals with calculating homogeneity of information using Eq. (3.3). Total entropy of a given dataset is obtained by first splitting the dataset based on different attributes and calculating entropy for each branch using Eq. (3.3), and adding the entropy of each branch proportionally.

$$H(x) = -\sum_{i} P(x_i) \log_2^{p(x_i)} \tag{3.3}$$

Where, $H(X)$ Entropy of the dataset, and

$P(x_i)$ Probability of an instance being labeled with certain class given a specific feature

Information gain of an attribute is calculated by subtracting entropy of the dataset from entropy of the target attribute/feature as shown in Eq. (3.4). It is used in common decision tress such as Iterative Dichotomiser 3 (ID3).

$$Info(A) = H(X) - H_A^{(X)} \tag{3.4}$$

Where, Info (A) Information gain of a specific attribute 'A'

H (X) Entropy of the dataset given in Eq. (3.3), and

$H_A^{(X)}$ Entropy of the specific attribute in the given dataset

The attribute with the highest information gain value will be the decision node in decision tree algorithm. However, attribute biasness problem in the decision node selection process is the drawback of information gain. An attribute which

have different unique values will attain higher information gain value. Though this attribute may not worth to be a decision node it will be selected as a decision node.

To avoid this biasness problem, latest decision tree algorithms such as C4.5/J48 use gain ratio instead information gain. Gain ratio normalizes information gain values by introducing split info concept as shown in Eq. (3.5). Through this process, an attribute with the highest gain ratio value wi be selected as a root/decision node. Nodes which have an entropy of zero are considered to be leaf node while nodes with entropy greater than zero will further split until the entropy is zero [37].

$$GainRatio(A) = \frac{Info(A)}{SplitInfo_A^{(x)}} \tag{3.5}$$

Where the split information is given by Eq. (3.6). $\frac{|V_j|}{|V|}$ stands for the ratio of unique values of a given attribute to total number of instances for the specific attribute A.

$$SplitInfo(A) = \sum_{j=1}^{n} \frac{|V_j|}{|V|} * \log_2^{\frac{|V_j|}{|V|}} \tag{3.6}$$

3.2.4 *Artificial Neural Network (ANN)*

ANN is an algorithm which is inspired by the structure and function of biological neural networks, central nervous system in human's brain [34]. It is used for both classification and regression tasks. In ANN, perceptron is used to classify linearly separable classes while MLP is used for classes which can't be separated using linear function. Weighted sum and certain activation functions are used in the learning process of neural networks.

3.2.5 *K Nearest Neighbor (K-NN)*

K-NN is a non-probabilistic algorithm used for both classification and regression tasks [23]. It is considered the simplest algorithm compared to the other machine learning algorithms. It is an instance based learner or lazy classifier since it takes

less amount of time in training phase while taking longer in classification. Classification in K-NN is based on similarity measurement among the K- nearest neighbors while putting the training dataset on a feature space. The similarity measure among the neighbor points is based on either of the distance functions Euclidean, Manhattan, Minkowski, and Hamming distance [37]. Among these measures, Euclidean distance is the commonly used distance measure in K-NN algorithm.

End users are expected to set the number of neighbors (K) so that K-NN will select those k number of instances among the training dataset based on the distance measure for further process. Class label of the selected instances will be used to predict class of an input instance. Majority vote of the k neighbors is used in assigning a class label to an input instance in classification while average calculation of the instances will be used in regression. K-NN is preferable when no prior knowledge about the distribution of the data is known and where a single sample can have multiple class labels. It has a high computation time since all instance features are used in distance calculation.

3.2.6 *RIPPER*

RIPPER is a rule based sequential covering algorithm where a rule is generated sequentially by learning one rule at a time while at the end set of rules are generated that cover all instances of a training set [34]. The algorithm takes a training dataset and starts by selecting the less prevalent class, a class label which have less number of instance/example. The next step for the algorithm is to select and create a combination of rules that cover all instances of the less prevalent class. During this process, information gain measure given in Eq. (3.7) is used and the other class (higher prevalent class) will be the default class. Rules are selected from the rules which consider all combination of attribute values for each attribute in the dataset.

$$Gain(R', R) = S * (\log_2^{\frac{N'_+}{N'}} - \log_2^{\frac{N_+}{N}}) \tag{3.7}$$

Where,	Gain(R', R)	Information gain measure used in comparing candidate rules
	R	Original rule
	R'	Candidate rule for selection
	N	Number of instances covered by R
	N'	Number of instances covered by R'
	N_+	Number of true positives in R
	N'_+	Number of true positives in R', and
	S	Total number of true positives in R and R'

3.2.7 *AdaBoost*

AdaBoost is an ensemble classifier which is based on iteratively correcting the misclassification made by other classification algorithms, usually weak classifiers. It takes a training dataset consisted of labeled instances and initially gives them an equal weight $w = \frac{1}{n}$, where n is the number of instances in the training dataset. After making initial classification using the selected weak algorithm, it will increase the weight of wrongly classified instances calculating the classification errors made by the algorithms. Repeating these process T times which is dynamically set on the algorithm, AdaBoost will assign a certain predefined class label for a given instance minimizing the classification error [38].

AdaBoost is among the popular machine learning algorithms due to its simple implementation and less susceptibility to over-fitting problem. The algorithm have been applied to many areas such as pattern recognition and intrusion detection.

3.3 UNSUPERVISED ALGORITHMS

In unsupervised machine learning, the algorithms don't need a labeled training dataset or examples for the purpose of learning. Unlike the supervised ones, these algorithms are mainly used for clustering task instead of classification and regression. K-Means and Gaussian Mixture Model (GMM) are among the common

algorithms categorized under unsupervised machine learning algorithms [24, 25, 34].

3.3.1 *K-Means*

K-Means is a popular clustering algorithm which partitions 'n' observations or instances into 'k' clusters through a number of iterations. The number of clusters is defined by end users and each instance in a dataset belongs to a cluster with the nearest mean. Mean values serve as a prototype of a cluster in K-Means algorithm. K-Means have loose relationship with K-NN, supervised machine learning algorithm. By setting the value of k to 1 in k-NN (Nearest centroid classifier or Rochhio algorithm), 1-NN classifier can be applied to clusters obtained by k-Means to classify new data into existing clusters.

K-Means algorithm performs clustering task first by selecting random 'k' means and grouping data points or instances in to 'k' clusters based on the distance between the means and data points. In each iteration, the mean values for each cluster are updated according to the data points inside the clusters and data points are grouped to a cluster with the closest mean. This process continues until there is no change in cluster data points and mean values. Different clusters and means can be obtained every time we run k-Means algorithm due to the random selection of initial k means but it assures each data point only belongs to exactly one cluster.

3.3.2 *GMM*

GMM is a clustering model for a mixture of 'M' Gaussian distributions with a goal of finding three model parameters that best fits the given dataset. Gaussian representations Mean and Covariance, and weight of each gaussian are the parameters for the model. Once the model that best fits the data obtained, GMM Compute posterior probability of data instances using each component and assign each instance to a cluster based on calculated likelihood. GMM with 'M' components is represented by Eq. (3.8).

$$P(x|\theta) = \sum_{k=1}^{M} w_k P(x|\theta_k) \tag{3.8}$$

Where,	w_k	Weight of the k^{th} component
	$P(x \mid \theta_k)$	Covariance of the k^{th} component
	θ_k	Mean of the k^{th} component

GMM Uses Expectation Maximization (EM) algorithm where initial estimates of mean and covariance are required for execution. The problem with EM is that if these initial estimates are poor, the algorithm can stuck in local optima. Mean and covariance can be obtained from K-Means algorithm and provided to EM as an input to avoid this problem. Among the various application areas of GMM, speaker identification and biometric verification are the common ones.

Different machine learning tools and libraries have been designed for data mining tasks. WEKA, Matlab, Encog, IBM SPSS modeler, KNIME, LIONsover, Mlpy, SAS enterprise miner, and oracle data miner are the common tools and libraries [39].WEKA; an open source tool with an option of both Graphical User Interface (GUI) and a command line, is used in this research.

A collection of state-of-the-art machine learning algorithms and data preprocessing tools have been integrated in WEKA for the standard data mining tasks. These tasks include classification, clustering, association rule mining, regression, and attribute selection. The tool provides extensive support in input data preparation, statistical evaluation of learning schemes, and visualization of both input and output of the learning process [24].

4

EXPERIMENTAL ANALYSIS

In this chapter, the overall experimental process followed in conducting this research is discussed. The experimental process shown in Figure 4.1 is composed of three modules; Data collection, Data Preprocessing & Feature Selection, and Classfication. Details of tasks done under these modules are described separately in the next sections. Network traffic generation and capturing, data preprocessing and feature selection, training and evaluating the algorithms are tasks which will be described in the coming sections.

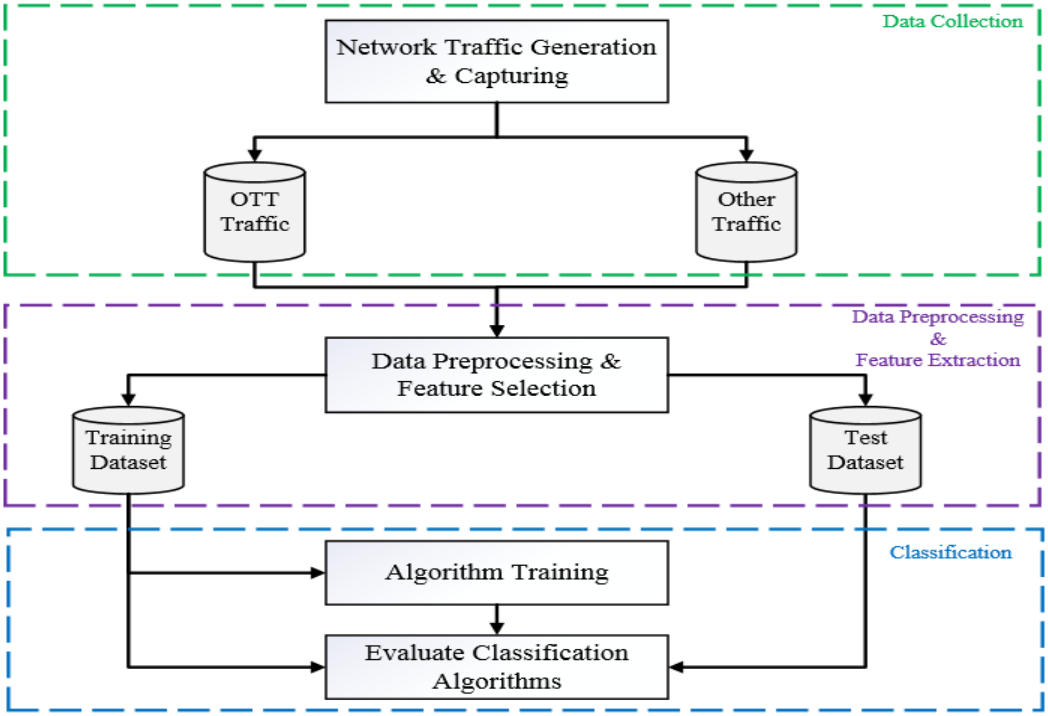


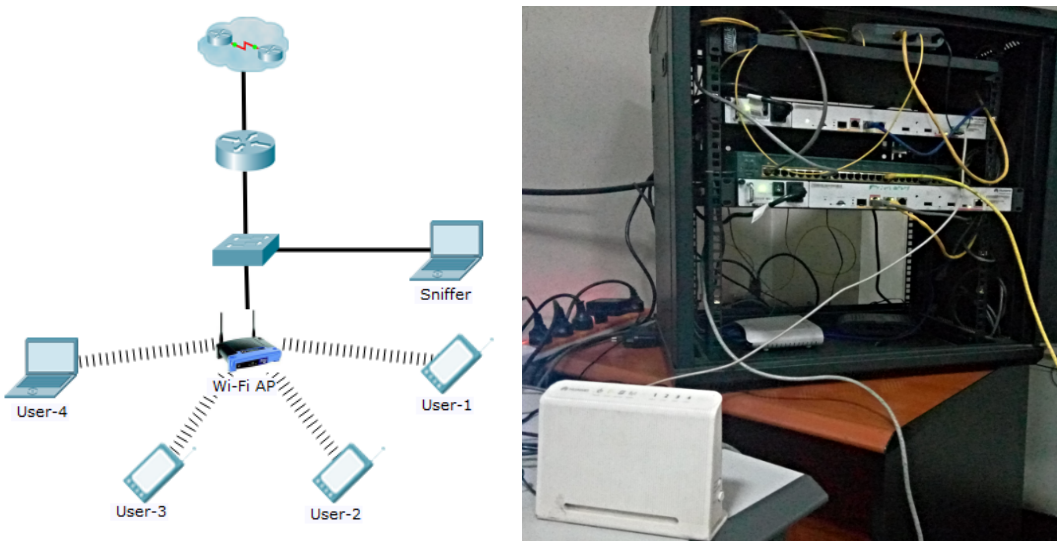
Figure 4.1: Overall experimental process [5].

4.1 NETWORK TRAFFIC GENERATION AND CAPTURING

Hence, a labeled network traffic dataset is needed to train the classification algorithms, the first task done is to generate and capture network packets consisting

of both MSISDN-based OTT and other traffic. This task could have been avoided if there is a publicly available labeled network traffic dataset which consist of the traffics related to this research. However, the existing publicly available datasets doesn't include traffics from MSISDN-based OTT applications as per the need for this work [7, 8]. Due to this, generating and labeling of the packets as per the need is a mandatory task as it is done in other related researches such as [6, 10].

Generation and capturing of the network packets is done using the topology shown in Figure 4.2. As it is seen from the topology, sniffer software is directly connected to the switch so that each packets passing through the Wi-Fi access point are captured. Details and specifications of the network and other components used in the traffic generation, and capture process are mentioned in Table 4.1.



(a) Traffic capturing architecture (b) Hardwares in the capturing architecture

Figure 4.2: Traffic generation and capture environment.

Network traffic generation and capture is done in two phases. The first phase focus on generating packets only from MSISDN-based OTT applications. OTT applications selected for this research are Viber, Tango, and Telegram. Viber and Tango are among the popular applications which use user's service number for authentication compared to the other popular OTT (Mobile VoIP) applications; Skype, Google Talk, ICQ , Nimbuzz, Yahoo, Fring, Vonage, and WeChat. Voice communications in Tango are encrypted while scrambling technique is used in Viber [9]. Telegram is also widely used OTT application among different users in Ethiopia recently.

Hence these applications are compatible with android and windows OS, the following versions of the applications,

- Viber 7.5.0.26
- Tango 4.8.226852
- Telegram 4.8.11

are installed on the smart phones and computer used for traffic generation.

The second phase focus on generating network traffics other than MSISDN-based OTT traffic. Packets from OTT applications Skype and Yahoo messenger together with other web based packets YouTube, Gmail, and Facebook are used in the second phase. During these two phases of traffic generation, attention is given not to contaminate packets with one another by closing all unrelated connections since it affect the classification performance. Network packets generated from MSISDN-based OTT application are stored to 'OTT Traffic' while traffics generated in second phase are stored to 'Other Traffic' data store as shown in Figure 4.1.

Through these processes, a total of two million packets have been captured and ready for the next step as shown in Table 4.2. Out of these packets, 794,333 (37%) of them are from the MSISDN-based OTT applications while the remaining packets are from other traffic category as shown in Figure 4.3.

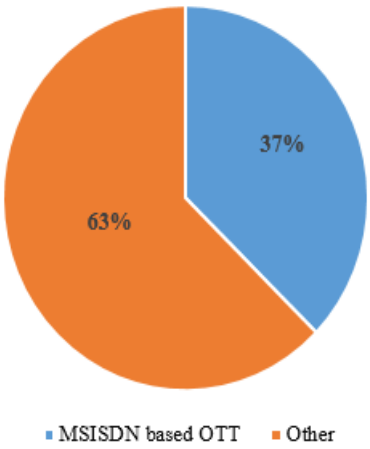


Figure 4.3: Pie chart for initially captured packets.

Table 4.1: Infrastructure used for network traffic generation & capture

Type/Device	Quantity	Purpose
Dell Laptop (4 GB RAM and 4 CPUs with 2.9 GHz clock rate)	2	Traffic Generating & Capturing
Samsung Smart-phone	2	Traffic Generation
Huawei Smart-phone	1	"
Huawei Router	1	"
Cisco Switch	1	Traffic Generating & Capturing
Wi-Fi Access Point	1	Traffic Generation
EPON Internet connection	100 Mb/s	"
Wireshark 2.4.3	-	Traffic Capture

4.2 DATA PREPROCESSING AND FEATURE SELECTION

Since the dataset used in this research is generated manually as per the need, tasks such as handling of missing data, and integration of multiple data sources is not done as part of data preprocessing. However, packet attribute/feature selection and handling of noisy data (outliers) is performed as a data preprocessing task as shown in Figure 4.4. Details of tasks done under this module; User's Internet Protocol (IP) based filtering, Manual attribute selection, Attribute worthiness evaluation, and Outlier removal, is discussed below.

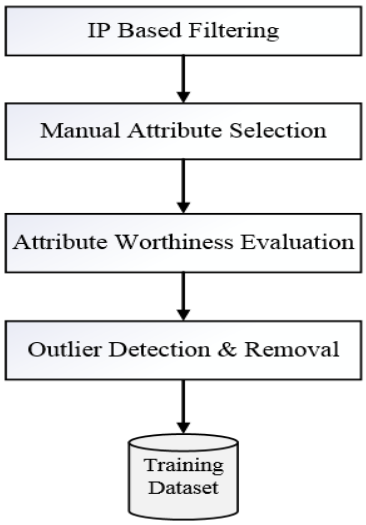


Figure 4.4: Data preprocessing and feature selection process.

4.2.1 IP based filtering

While capturing network traffic packets using the topology mentioned in section 4.1 and Wireshark tool, the packets captured are not only from the user's IP. Packets from other network transactions such as DNS server communications have been also captured. These packets are not relevant for the classification process and only packets directly related to the four users are needed to be used for further process. Using IP filtering command, a total of 172,188 are identified and removed from the dataset as shown in Table 4.2.

Table 4.2: Packet size before and after IP based filtering

Status	Packet type		**Total**
	MSISDN-Based OTT	Other	
Before filtering	794,333	1,334,675	2,129,008
After filtering	626,675	1,330,145	1,956,820

Out of the total 2,129,008 initially captured packets , only 1,956,820 of them are filtered to be used in the next process. From the removed packets, the larger share was from the MSISDN-based OTT applications; 167,658 packets.

4.2.2 Manual attribute selection

Wireshark 2.4.3 tool is used to capture packets and more than 50 packet features can be captured using it. Out of these features, only five of them namely length, protocol, delta time, relative time, and cumulative byte have been selected as initial features for the classification task. Selection of these attributes is based on assessing related works and all the features that can be captured using the tool.

Short description of the selected attributes is given below [10]:

- **Packet length:** It is length of each packet crossing the real network and controlled by type of Hardware and Software the network is using. It's data type is numeric.
- **Delta time:** It is the arrival time between two successive packets. In other words, delta time is the time since the previous packet have arrived and can

be used to measure network roundtrip and server response time. It's data type is numeric and is sometimes referred as packet inter arrival time.

- **Relative time:** It is the elapsed time between first and the current packet. In other words, it is the total time it take to capture the last packet starting from capturing the first packet. It's data type is numeric and also referred as cumulative time.

- **Cumulative byte:** It is amount of data that can be transmitted between the sender and receiver when a large block of data crosses over the network. It is considered the scale that measures the total bytes that are transmitted in the time interval from the captured traffic. Its data type is also numeric.

- **Protocol:** It describe type of protocol used in each packet. It's data type is nominal and TCP, UDP, DNS, SMTP, Simple Network Management Protocol (SNMP), Simple Service Discovery Protocol (SSDP), and Secure Sockets Layer (SSL) are among the common protocols captured by wireshark.

4.2.3 Attribute worthiness evaluation

Once the five packet attributes are selected manually, the next task is to evaluate the worthiness of each attribute in relation to the classification class; how much a given attribute is related to target classification class. Selection of attributes/features based on a certain evaluation techniques will optimize the required processing resources in addition to improving the classification performance [40].

There exist various attribute worthiness evaluation methodologies that can be used depending on the attribute data type. Among these techniques, two of them namely correlation and information gain ratio have been used here. These selection is done considering the data type of the attributes in this research; four numeric and a nominal. Details of how these techniques perform the evaluation and results obtained are discussed here.

- **Correlation:** It evaluates the worthiness of an attribute by measuring the correlation between an attribute and the classification class. Pearson correlation

coefficient of the attribute which is calculated in Eq. (4.1) is used for this research.

$$\frac{N(\sum xy) - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}} \tag{4.1}$$

Where, N Number of instances in the given attribute
x Instance values of the given attribute and
y classification class label

Coefficient values are always with in the range [-1,1] and the value '1' shows that the attribute is in strong positive relation with the target classification attribute [41]. On the other hand, the value '-1' shows the attributes are in strong negative relation while 'o' indicates that the attributes are not correlated.

- **Information Gain Ratio:** Evaluates the worthiness of an attribute by calculating the information gain ratio of an attribute with the predefined classification class. Gain ratio is recommended to avoid the attribute biasness problem in information gain as it is already stated in Chapter 3.

The result obtained through these two attribute worthiness evaluation techniques for the five attributes is given in Table 4.3. The attribute 'delta time' scores the

Table 4.3: Attribute worthiness evaluation results

Attribute	Correlation	Information Gain Ratio
Cumulative byte	0.51	0.156
Relative time	0.47	0.063
Packet length	0.29	0.08
Protocol	0.26	0.18
Delta time	0.12	0.045

least value both in correlation and gain ratio compared to the other attributes. Therefore, since it is the least predictor attribute compared to the other attributes, it is removed from the attribute list and the remaining four attributes; protocol,

packet length, cumulative byte, and relative time; have been selected as attributes for the classification task. Totally, five attributes including classification class 'Is OTT traffic?' which is a 'nominal' data type with values 'yes' and 'No' have been used as final attributes for the classification task.

4.2.4 Outliers detection and removal

An outlier is a value which is not consistent with the remaining dataset and also considered as noisy data. These values in a dataset need to be detected and removed to enhance classification performance of algorithms [42].Inter Quartile Range (IQR) method is used for detecting outliers in this research. IQR is suitable for detecting outliers in a numerical data; which is the case here in our dataset.

IQR frist sort the values of an attribute and divide the dataset in to four equal parts. Once the three quartiles (Q1,Q2,Q3) are identified; where Q2 is the median, an IQR value is calculated by subtracting Q1 from Q3. An outlier factor of '1.5' is used to calculate upper and lower boundary values using Eq. (4.2) and (4.3), respectively. Values greater than the upper boundary and less than lower boundary are treated as outliers.

$$Upper - boundary = Q1 - (1.5 * IQR) \tag{4.2}$$

$$Lower - boundary = Q2 + (1.5 * IQR) \tag{4.3}$$

Through this process, 52,524 outliers are detected and instances associated with these outlier values have been removed from the dataset. In other words, 3% of the total instances are treated as outliers. The detection is made per attribute and number of outliers in each numerical attributes is shown in Table 4.4.

After performing the above mentioned data preprocessing tasks on the initial data, a training dataset which consists of 1.7 million packets have been obtained. Out of this, 25.1 % of the packets are from MSISDN-based OTT applications while the

Table 4.4: Number of outliers per attribute

Attribute	Number of outliers
Cumulative byte	41,062
Relative time	11,445
Packet length	17
Total	**52,524**

remaining were from other traffics as show in Table 4.5. WEKA workbench is then used to train and evaluate the algorithms using '.ARFF' file format of the dataset. The files captured from Wireshark were in '.CSV' format and conversion to '.ARFF' was done using WEKA as shown in Figure 4.5.

Table 4.5: Size of dataset after data preprocessing task

Packet Type	No of Packets	Share
MSISDN-based OTT	431,920	25.1%
Other	1,289,083	74.9%
Total	**1,721,003**	

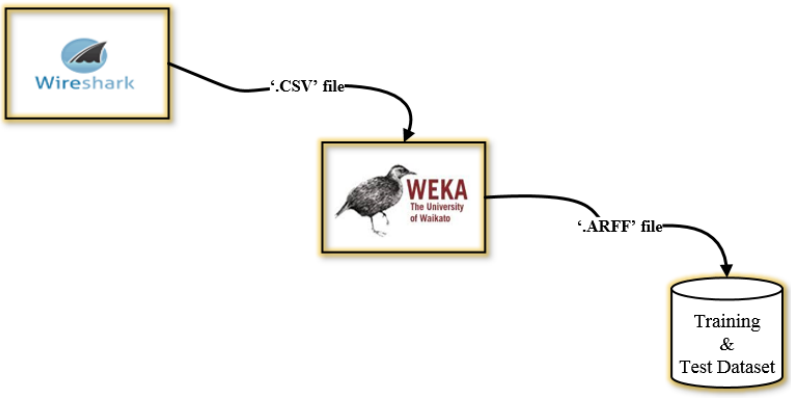


Figure 4.5: File formats used.

4.3 ALGORITHM TRAINING

Once the data preprocessing and feature selection task is completed, the next step is to train the selected algorithms and building classification model. Three supervised machine learning algorithms;AdaBoost + J48, RIPPER, and SVM; have been selected and used to create the models. A server having 24 GB RAM and

8 CPUs each with 2 GHz clock rate, and WEKA with default parameter setting is used in training the algorithms.

Two separate training datasets are used for the two classification tasks performed as shown in the process flow in Figure 4.6. In the first classification task, MSISDN-based OTT packets are detected from a network traffic dataset. However, the objective of second classification model is to detect voice call packets from MSISDN-based OTT traffics. On both datasets, 'Yes' and 'No' labels are used to label the packets which refers either 'Is it MSISDN-based OTT traffic?' or 'Is it Voice call traffic?' questions.

In the first classification task; MSISDN-based OTT | other traffic; the training dataset is consisted of traffics from Viber, Tango, and Telegram. But, for the second classification; Voice call | Non-Voice call ; only packets from Telegram are included in the dataset. The overall time taken to construct the classification models for each algorithms is shown in Table 4.6.

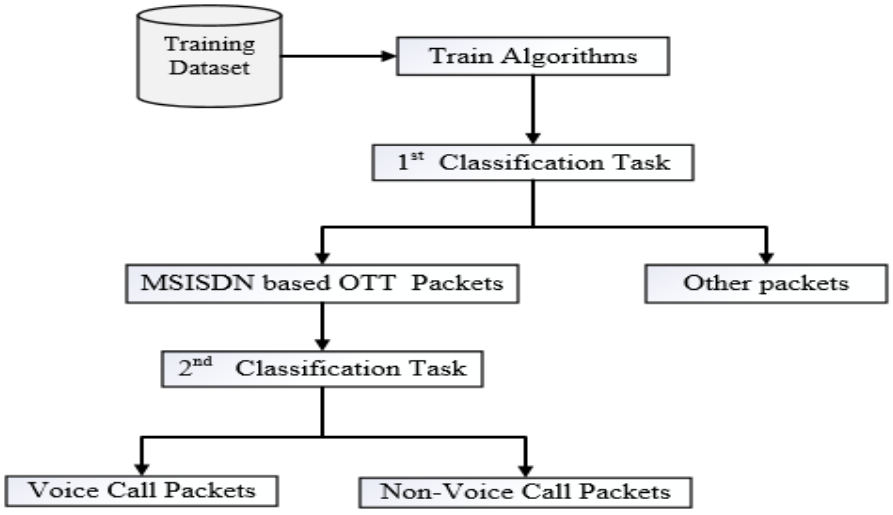


Figure 4.6: Overall packets classification process.

Table 4.6: Time taken (in minutes) to train algorithms

Algorithm	Classification Task	
	MSISDN-Based OTT | Other	Voice call | Non-Voice call
AdaBoost + J48	37.9	0.79
RIPPER	314.6	4.69
SVM	1,594.2	26.62

4.4 ALGORITHM EVALUATION

Evaluating and comparing the classification performance of the algorithms; how well they detect target packets; is the main objective of this research. Two techniques; Validation through ten cross-fold and separate test data; have been used to evaluate performance of the algorithms.

In ten cross-fold validation, the given training dataset is partitioned to ten equal parts. Then only one partition is used as test dataset at a time while the remaining nine partitions serve as a training dataset. This process is repeated ten times until each partition serve as both test and training dataset. This evaluation technique is usually used when there is no sufficient amount of test data to evaluate the algorithms performance.

The other evaluation is done by supplying separate test data to the classification model which is constructed using ten cross-fold validation. These test data are not used in training the algorithms and are labeled with the predefined classification classes. The model will predict class label for the test data instances and displays the label in a separate column (sample is shown in Annex). The evaluation is done by comparing the class label of an instance in the test dataset with the predicted label.

The size of separate test dataset used in this research is shown in Table 4.7. A total of 206,219 and 120,792 instances have been used in the test dataset for the first and second classification tasks respectively.

Table 4.7: Number of instances in test dataset

Class label	Classification task	
	MSISDN-Based OTT | Other	Voice call | Non-Voice call
Yes	180,519	110,739
No	25,700	10,053
Total	**206,219**	**120,792**

As an evaluation metrics, confusion matrix and classification accuracy which are common evaluation metrics in classification tasks such as [6, 10, 13], are used to-

gether with F-measure [12] and ROC curve [10]. Details for each of these evaluation metrics is discussed below.

4.4.1 *Confusion Matrix*

It is a 2X2 matrix which contains True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values for the classification class labels. Confusion matrix for two class labels (Class 'A' and 'B') is shown in Table 4.8.

Table 4.8: Confusion Matrix

	Class 'A'	Class 'B'
Class 'A'	TP	FP
Class 'B'	FN	TN

Where, TP Number of instances correctly classified as class 'A'

FP Number of instances that belong to class 'A' but classified as class 'B'

FN Number of instances that belong to class 'B' but classified as class 'A'

TN Number of instances correctly classified as class 'B'

4.4.2 *Classification Accuracy*

The accuracy denoted by Eq. (4.4) measures the ratio of correctly classified instances (both class 'A' and 'B') with respect to the whole test data. In other words, it shows the percentage of correctly classified instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

4.4.3 *F-Measure*

It is the harmonic mean of Precision and recall calculated using Equation 4.5.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.5}$$

Precision for class 'A' denoted by Eq. (4.6) measures the ratio of instances correctly classified as class 'A' with respect to total number of instances classified as class 'A'. Precision for class 'B' is calculated in the same way.

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

Recall for class 'A' denoted by Eq. (4.7) measures the ratio of instances correctly classified as class 'A' with respect to total number of instances available (both class 'A' and 'B'). Recall for class 'B' is calculated in the same way.

$$Recall = \frac{TP}{TP + FN} \tag{4.7}$$

4.4.4 *ROC curve*

ROC curve; which is also referred as threshold curve; is a graphical representation of True Positive Rate (TPR) against False Positive Rate (FPR) at different threshold values. TPR represents total number of positives in the sample while FPR is for total number of negatives. TPR is displayed in the Y axis while FPR is displayed in the X axis. Both TPR and FPR values are with in the range [0,1].

ROC Curves which are nearer to the upper 90 degree (0,1) of the graph are considered as perfect classifiers while curves which tend to lie below the linear line (y=x) are considered as low level classifiers. Perfect classifier ROC curve will go straight up the Y axis and then along the X axis.

In addition to the above mentioned evaluation metrics, the time taken by the algorithms to classify the given dataset is also considered in comparing performance of the algorithms.

5

RESULTS AND DISCUSSION

Ten cross-fold and separate test data validation techniques have been used for both MSISDN-based OTT | other traffic, and Voice call | Non-Voice call traffic classification experiment. Size of the separate test data and description of the evaluation metrics; confusion matrix, F-Measure, ROC curve, and the time taken to perform classification; is given in Section 4.4. The same machine mentioned in Section 4.3 which is used for algorithm training is also used for performance evaluation.

While reviewing performance of the three algorithms on both classification tasks, different results have been obtained in ten cross-fold and separate test data validation as shown in Table 5.1 and 5.2 (on page 47 and 49 respectively) . These performance variations are because the algorithms are familiar with the class labels of instances using ten cross-fold, while new data (instances never used in training) is used in the other validation technique.

Use of test data instances for algorithm training in cross-fold validation is the other reason for this performance variation. Unlike cross-fold, test data instances will not be used for algorithm training in separate test data validation. Though the experiment is done in stratified cross-fold; each instance is used as training and test dataset; the algorithms are somehow familiar with the test data in cross-fold validation. Due to this, results obtained in cross-fold are usually better compared to using separate test data validation.

Hence the main target of classification models is to classify traffic packets in real environment, good performance in separate test dataset validation is a must. In real environment, classification models are expected to perform detection whether the packets are new or used in building (training) the models.

Ten cross-fold validation is usually used when there is a lack of sufficient separate test data. That is the reason why it is also used here as a validation technique since size of the separate test data is not proportional to size of packets used

in the training dataset. In this chapter, results obtained from evaluation of the algorithms on each classification tasks is discussed separately.

5.1 MSISDN-BASED OTT PACKET DETECTION

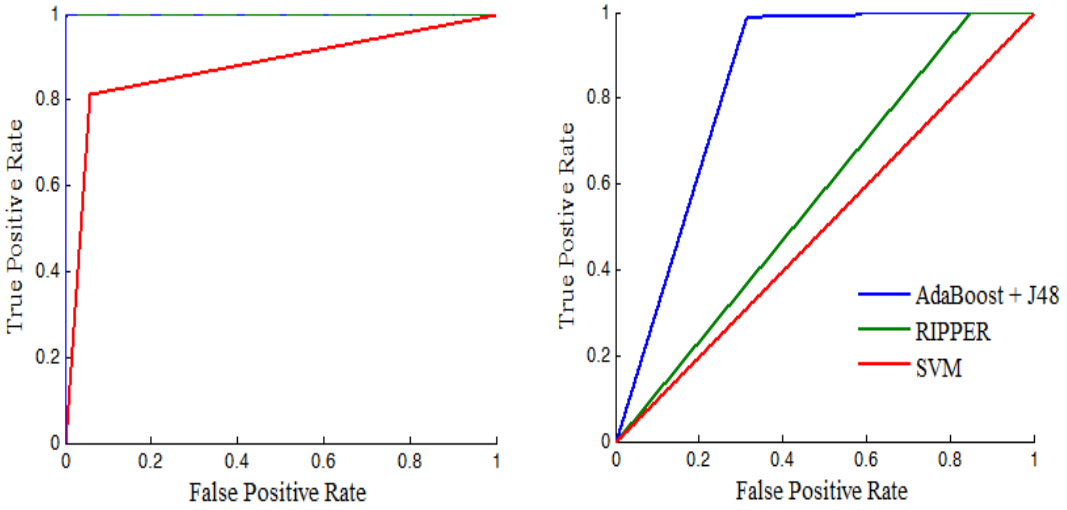
In the detection of MSISDN-based OTT packets, AdaBoost with J48 is the best performer compared to RIPPER and SVM on both ten cross-fold and separate test data validation. It achieved an overall classification accuracy of 99.98% and 89.74% in ten cross-fold and separate test data validation respectively as shown in Table 5.1. RIPPER is the second best algorithm in detecting MSISDN-based OTT packets. It achieves better classification accuracy compared to SVM in both validation techniques.

An accuracy of 99.96% and 89.43% have been recorded by RIPPER in ten cross-fold and separate test data validation respectively. AdaBoost with J48 and RIPPER have attained nearly the same overall classification accuracy; 0.02% and 0.31% of performance variation in cross-fold and separate test data validation respectively. SVM is the least performer in both techniques achieving an accuracy of 90.97% and 87.36% in ten cross-fold and separate test data validation respectively.The performance of SVM in ten-cross fold is much lower compared to the other algorithms.

As shown from Table 5.1 and discussed earlier, classification accuracy achieved by the three algorithms while using separate test data is much lower than the performance achieved in ten cross-fold validation. In this classification task, 206,219 packets which is 0.12% of the data used in training have been used as test dataset. Increasing size of this test data is a recommended solution to enhance classification performance.

In classification tasks like the one used here, algorithm training is a one time task even if the time to train algorithms varies from algorithm to algorithm as demonstrated in Chapter 4. The resources used in algorithm training are free once the classification models are constructed. Once the models are ready, they are expected to perform detection every time an end user provide the dataset to operate. Due to this, the time it takes to perform the classification is used to compare the algorithms instead of training time.

SVM takes much longer time compared to the other two algorithms to perform the classification in ten cross-fold validation. It takes 60 hours for SVM to categorize the packets which is nearly 11 times the time taken by AdaBoost with J48. However, this performance gap is much reduced and the algorithms took nearly the same time to perform the classification in separate test data. RIPPER takes relatively less amount of time ; 0.02 second; in separate test data validation.



(a) Ten cross-fold

(b) Separate test data

Figure 5.1: ROC curve for MSISDN-based OTT | Other packet classification

Performance comparison for the three algorithms is also graphically represented using ROC curve as shown in Figure 5.1. As it is seen from Figure 5.1a, AdaBoost with J48 and RIPPER have attained almost the same curve in ten cross-fold validation. Their curves lie on the upper 90 ^{0} at (0,1) which shows these two algorithms were accurate in ten cross-fold validation. In the meantime, ROC curves of these two algorithms differ while using separate test dataset as shown in Figure 5.1b. AdaBoost with J48 remains best classifier even if it's performance deviates from what achieved in ten cross-fold validation. In both Validation techniques, SVM is the least performer.

Table 5.1: Classification performance of the algorithms

Validation Technique	Algorithm	Confusion Matrix (OTT Traffic?)			Accuracy	F-Measure	Time (Minutes)
10 cross fold	AdaBoost + J48		**Yes**	**No**	**99.98%**	1.00	332.1
		Yes	431,774	146			
		No	141	1,288,942			
	RIPPER		**Yes**	**No**	**99.96%**	1.00	490
		Yes	431,495	425			
		No	199	1,288,884			
	SVM	1	**Yes**	**No**	**90.97%**	0.91	3,600
		Yes	350,966	80,954			
		No	74,319	1,214,764			
Separate test data	AdaBoost + J48		**Yes**	**No**	**89.74%**	0.864	0.05
		Yes	180,511	8			
		No	21,139	4,561			
	RIPPER		**Yes**	**No**	**89.43%**	0.858	0.002
		Yes	180,507	12			
		No	21,786	3,914			
	SVM	1	**Yes**	**No**	**87.36%**	0.816	0.005
		Yes	180,163	356			
		No	25,700	0			

5.2 MSISDN-BASED OTT VOICE CALL PACKET DETECTION

Considering the fact that OTT bypass fraudulent calls are delivered as OTT voice calls, detection of MSISDN-based OTT voice calls from other service flows such as video call and voice/text message is a crucial step. For this reason, the other experiment done in this research was to detect voice call packets from Telegram service flows; text and voice messages including attachments.

Like the previous classification task, this experiment is also done using both validation techniques.AdaBoost with J48 achieved best classification accuracy (99.6%) in ten cross-fold while achieving the lowest accuracy in separate test data validation as shown in Table 5.2. SVM have the highest classification accuracy when provided with separate test data. It achieved an accuracy of 95.35% which is nearly double of the result achieved by AdaBoost with J48; 48.4%.

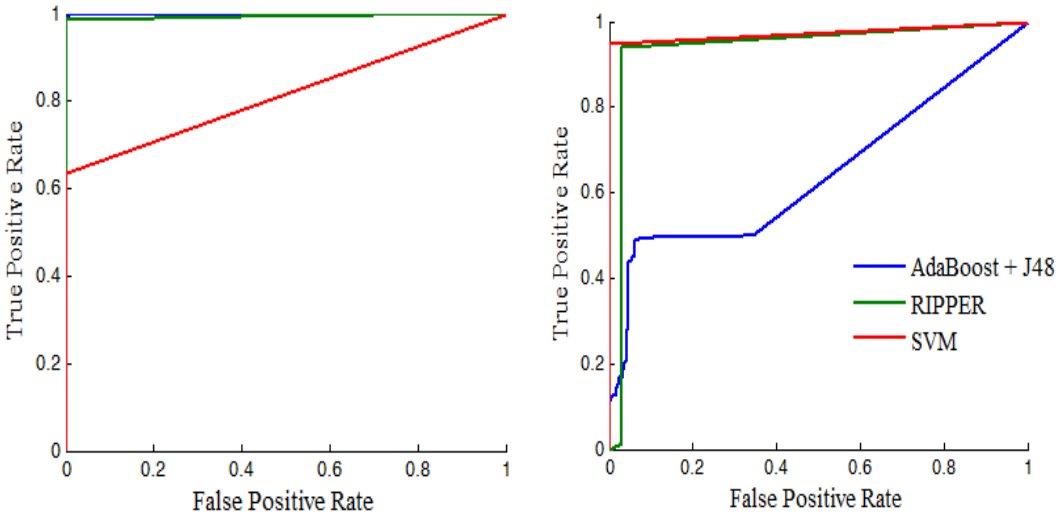
With regard to classification time, SVM takes longer time to classify the datasets on both cross-fold and separate test data validation compared to the other algorithms. Like in the detection of MSISDN-based OTT packets, all of the three algorithms take longer classification time in ten cross-fold validation compared to the time they take in separate test data validation.

Classification time is dependent on factors such as size of the data to be classified and the machine used in the classification process. Number of folds used in cross-fold validation is also another factor. The time achieved by these algorithms in this research could have been different if the number of folds and size of the dataset is changed.

ROC curve for the three algorithms while detecting voice call packets is shown in Figure 5.2. AdaBoost with J48 have attained an accurate ROC curve in cross-fold validation followed by RIPPER as shown in Figure 5.2a. However, ROC curve obtained by AdaBoost with J48 in separate test data is much lower than the other algorithms as shown in Figure 5.2b. ROC for SVM is the better one in separate test data validation.

Table 5.2: Classification performance of the algorithms

Validation Technique	Algorithm	Confusion Matrix (Voice call Traffic?)			Accuracy	F-Measure	Time (Minutes)
10 cross-fold	AdaBoost + J48		**Yes**	**No**	**99.68%**	0.997	6.92
		Yes	28,297	213			
		No	101	70,964			
	RIPPER		**Yes**	**No**	**99.49%**	0.995	35
		Yes	28,120	390			
		No	108	70,957			
	SVM	1	**Yes**	**No**	**89.47%**	0.887	131
		Yes	18,075	10,435			
		No	44	71,0214			
Separate test data	AdaBoost + J48		**Yes**	**No**	**48.4%**	0.58	0.05
		Yes	48,935	61,804			
		No	515	9,538			
	RIPPER		**Yes**	**No**	**94.12%**	0.947	0.03
		Yes	104,000	6,739			
		No	357	9,696			
	SVM	1	**Yes**	**No**	**95.35%**	0.958	0.05
		Yes	105,132	5,607			
		No	12	10,041			



(a) Ten cross-fold (b) Separate test data

Figure 5.2: ROC curve for Voice call | Non-Voice call packet classification

Considering objective of the research, where algorithms are provided with new data traffic and they are expected to detect voice call packets, results obtained in separate dataset are more useful. So, the low performance achieved by AdaBoost in detection of voice calls makes it less preferable compared to the other two algorithms. Even if the time it takes to perform the classification is almost the same, RIPPER and SVM have relatively achieved better in separate test data validation. RIPPER achieves best in the first classification task while SVM is the better one in the second.

However, the algorithms performance in detecting the target packets; MSISDN-based OTT and voice call in the first, and second classification task respectively; varies. In other words, the TP value from confusion matrix can be used to compare the performance of the algorithms. In the first task, RIPPER detect 180,507 of the MSISDN-based OTT packets while SVM detect 180,163 packets. Unlike the first classification task, SVM detects better number of voice call packets in the second classification task. It detects 105,132 of the voice call packets which is better compared to 104,000 packets detected by RIPPER. Based on these performances, SVM is recommended for the network traffic classification task in this research.

6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Interconnect bypass frauds are still a main challenge for service providers and regulatory bodies even if different detection techniques are implemented to tackle them. Their impact ranges from customer dissatisfaction to revenue loss and security. Telecommunication service providers with higher call interconnection fee are the main targets of interconnect bypass fraud. The focus of this research is on OTT bypass which is considered as paradigm shift in interconnect bypass fraud. Detecting and blocking fraudulent OTT calls is one of the recommended solution of this fraud. In this research, the applicability of machine learning algorithms in detecting OTT voice call packets; one step in the detection process; has been investigated.

Performance of three supervised machine learning algorithms is evaluated in the OTT voice call packet detection process. These algorithms are trained on the labeled dataset and the evaluation is done using ten cross-fold, and separate test data validation techniques using predefined metrics. Labeling of packets with the corresponding class is done during the traffic generation phase while using the controlled laboratory environment.

In the performance evaluation, AdaBoost with J48 is the best performer compared to the other algorithms in the detection of MSISDN-based OTT packets. It scores higher values on all evaluation metrics and validation techniques except the classification time recorded in separate test data validation. The algorithm is also the best performer in detecting voice call packets from OTT traffic using ten cross-fold validation.

However, this performance domination is not seen in detecting voice calls while separate test data validation technique is used. AdaBoost with J48 achieves an over-

all classification accuracy of below 50% while SVM is the best performer in this category. SVM achieves a classification accuracy of 95.35% which is nearly double of a result obtained by AdaBoost. RIPPER was the second best classifier in most of the metrics while using both validation techniques.

Results recorded in separate test dataset validation technique are preferred as performance indicators compared to the results achieved in ten cross-fold validation. In reality, classification models are expected to face new network traffic type which are not used during algorithm training. Hence the objective of this thesis is to detect OTT voice call packets for the purpose of OTT bypass fraud detection, the classification models to be proposed here are expected to detect the packets while facing different network traffics. For this reason, focus is given for results obtained in separated test data validation technique while comparing performance of the algorithms.

Even if size of the test data used affects classification accuracy in separate test data validation, the much lower accuracy attained by AdaBoost in detecting voice call packets makes it less preferable in this classification task. However, SVM and RIPPER algorithms recorded better results on both classification tasks. TP value from confusion matrix which shows how many of OTT voice call packets have been detected is finally used to recommend the algorithm for this work. SVM scores better TP value and recommended for classification task in this thesis work.

6.2 RECOMMENDATIONS FOR FUTURE WORK

The future works proposed from this research include:

- Researches on how to detect fraudulent OTT calls (initiated as regular calls but delivered as OTT calls) from the general MSISDN-based OTT voice calls packets detected here. Detection of fraudulent OTT calls will lead to proposing full mitigation technique for OTT bypass fraud.
- Retesting performance of machine learning algorithms proposed here on actual operator network by increasing the dataset size. In addition to that, incorporating additional MSISDN-based OTT applications on both classifica-

tion tasks is recommended since it may impact the performance reported in this research.

- Besides increasing size of the dataset, preparation of adequate labeled MSISDN-based OTT voice call packets is recommended as future work. Accomplishing this task will help other researchers in using supervised machine learning algorithms for classification tasks related to OTT traffic.

REFERENCES

[1] CFCA. (2017). Global fraud loss survey, [Online]. Available: https://www.cfca.org.

[2] L. Cortesao, F. Martins, A. Rosa, and P. Carvalho, "Fraud management systems in telecommunications: A practical approach," in *12th Int. Conf. on Telecommun. (ICT)*, 2005, pp. 167–182.

[3] E. Tarmazakov and D. Silnov, "Modern approaches to prevent fraud in mobile communications networks," in *Conf. of Russian Young Researchers in Elect. and Electron. Eng. (EIConRus)*, IEEE, 2018, pp. 379–381.

[4] M. Sahin and A. Francillon, "Over-the-Top bypass: Study of a recent telephony fraud," in *Proc. SIGSAC Conf. on Comp. and Comm. Security*, ACM, 2016, pp. 1106–1117.

[5] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *2nd Int. Conf. on Comput. and Commun. (ICCC)*, IEEE, 2016, pp. 2451–2455.

[6] J. Datta, N. Kataria, and N. Hubballi, "Network traffic classification in encrypted environment: A case study of google hangout," in *21th Nat. Conf. on Commun. (NCC)*, IEEE, 2015, pp. 1–6.

[7] R. Alshammari and N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying ssh and skype," in *Symp. on Comput. Intell. for Security and Defense Applicat. (CISDA)*, IEEE, 2009, pp. 1–8.

[8] R. Alshammari, N. Zincir-Heywood, and A. A. Farrag, "Performance comparison of four rule sets: An example for encrypted traffic classification," in *World Congr. on Privacy, Security, Trust and the Manage. of e-Business*, IEEE, 2009, pp. 21–28.

[9] A. Azfar, K. R. Choo, and L. Liu, "A study of ten popular android mobile VoIP applications: Are the communications encrypted?" In *47th Hawaii Int. Conf. on System Sci. (HICSS)*, IEEE, 2014, pp. 4858–4867.

[10] G. Al-Naymat, M. Al-Kasassbeh, N. Abu-Samhadanh, and S. Sakr, "Classification of VoIP and non-VoIP traffic using machine learning approaches," *J. of Theoretical and Appl. Inform. Technol.*, vol. 92, no. 2, p. 403, Oct 2016.

[11] M. Sudozai, N. Habib, S. Saleem, and A. Khan, "Signatures of viber security traffic," *J. of Digital Forensics, Security and Law*, vol. 12, no. 2, pp. 109–120, Jun 2017.

[12] M. Korczyński and A. Duda, "Classifying service flows in the encrypted skype traffic," in *Int. Conf. on Commun. (ICC)*, IEEE, 2012, pp. 1064–1068.

[13] M. Rathore, A. Paul, A. Ahmad, M. Imran, and M. Guizani, "High-speed network traffic analysis: Detecting VoIP calls in secure big data streaming," in *41th Conf. on Local Comput. Netw. (LCN)*, IEEE, 2016, pp. 595–598.

[14] B. Reaves, E. Shernan, A. Bates, H. Carter, and P. Traynor, "Boxed out: Blocking cellular interconnect bypass fraud at the network edge.," in *in USENIX Symp. on Security*, 2015, pp. 833–848.

[15] T. K. Sawe, "Emergence of OTT communication services and sustenance of revenue among kenya telcos.," *Int. J. of Innov. Sci., Eng. and Technol.*, vol. 3, no. 8, pp. 377–381, Aug 2016.

[16] J. Sujata, S. Sohag, D. Tanu, D. Chintan, P. Shubham, and G. Sumit, "Impact of Over-the-Top (OTT) services on telecom service providers," *Indian J. of Sci. and Technol.*, vol. 8, no. S4, pp. 145–160, Feb 2015.

[17] T. Kapourniotis, T. Dagiuklas, G. Polyzos, and P. Alefragkis, "Scam and fraud detection in VoIP networks: Analysis and countermeasures using user profiling," in *50th Congr. of FITCE*, IEEE, 2011, pp. 1–5.

[18] Z. Shaikh and D. Harkut, "An overview of network traffic classification methods," *Int. J. on Recent and Innov. Trends in Comput. and Commun.*, vol. 3, no. 2, pp. 482–488, Feb 2015.

[19] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *J. of IEEE Commun. Surveys and Tutorials*, vol. 10, no. 4, pp. 56–76, Oct 2008.

[20] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Comput. Commun. Review*, vol. 36, no. 5, pp. 5–16, Oct 2006.

[21] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS Performance Evaluation Review*, ACM, vol. 33, 2005, pp. 50–60.

[22] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press, 2014.

[23] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," in *8th Int. Conf. on Inform., Intell., Syst. and Applicat. (IISA)*, IEEE, 2017, pp. 1–8.

[24] I. Witten, E. Frank, M. Hall, and C. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2016.

[25] A. Smola and S. Vishwanatha, *Introduction to Machine Learning*. Cambridge: Cambridge University Press, 2008.

[26] J. Chen, C. Huang, and C. Shih, "The exploration of machine learning for abnormal prediction model of telecom business support system," in *19th Asia-Pacific Network Operations and Manage. Symp. (APNOMS)*, IEEE, 2017, pp. 211–214.

[27] A. H. Elmi, S. Ibrahim, and R. Sallehuddin, "Detecting sim box fraud using neural network," in *IT Convergence and Security*, Springer, 2013, pp. 575–582.

[28] X. Min and R. Lin, "K-means algorithm: Fraud detection based on signaling data," in *World Congr. on Services*, IEEE, 2018, pp. 21–22.

[29] S. Qayyum, S. Mansoor, A. Khalid, Z. Halim, A. R. Baig, *et al.*, "Fraudulent call detection for mobile networks," in *Int. Conf. on Inform. and Emerging Technologies (ICIET)*, IEEE, 2010, pp. 1–5.

[30] A. Mishra and S. Reddy, "A comparative study of customer churn prediction in telecom industry using ensemble based classifiers," in *Int. Conf. on Inventive Computing and Informatics (ICICI)*, IEEE, 2017, pp. 721–725.

[31] P. Dalvi, S. Khandge, A. Deomore, A. Bankar, and V Kanade, "Analysis of customer churn prediction in telecom industry using decision trees and logistic regression," in *Symp. on Colossal Data Anal. and Networking (CDAN)*, IEEE, 2016, pp. 1–4.

[32] U. Yabas, H. C. Cankaya, and T. Ince, "Customer churn prediction for telecom services," in *36th Annu. Conf. on Comput. Software and Applicat. (COMPSAC)*, IEEE, 2012, pp. 358–359.

[33] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," in *15th Annu. Conf. on Comput. Security Applicat. (ACSAC)*, IEEE, 1999, pp. 371–377.

[34] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine Learning: Algorithms and Applications*. New York: CRC Press, 2017.

[35] H. Bhavsar and A. Ganatra, "A comparative study of training algorithms for supervised machine learning," *Int. J. of Soft Comput. and Eng. (IJSCE)*, vol. 2, no. 4, pp. 2231–2307, Sep 2012.

[36] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Massachusetts: MIT Press, 2010.

[37] R. Choudhary and H. K. Gianey, "Comprehensive review on supervised machine learning algorithms," in *Int. Conf. on Machine Learning and Data Sci. (MLDS)*, IEEE, 2017, pp. 37–43.

[38] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Machine Learning (ICM)*, Bari, Italy, vol. 96, Jul 1996, pp. 148–156.

[39] D. Pop and G. Iuhasz, "Overview of machine learning tools and libraries," Inst. e-Austria Timisoara.

[40] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. of machine learning research*, vol. 3, pp. 1157–1182, Mar 2003.

[41] D. Lane, "Online statistics education: A multimedia course of study," in *EdMedia: World Conf. on Edu. Media and Technol.*, AACE, 2003, pp. 1317–1320.

[42] N. Schwertman, M. Owens, and R. Adnan, "A simple more general boxplot method for identifying outliers," *Computational statistics and data analysis*, vol. 47, no. 1, pp. 165–174, Aug 2004.