ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

MACHINE LEARNING APPROACH FOR EXTERNAL FRAUD DETECTION

M.Sc. THESIS

Aji MUBALAIKE

Department of Applied Informatics

Information Security Engineering and Cryptography Programme

DECEMBER 2018

ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

MACHINE LEARNING APPROACH FOR EXTERNAL FRAUD DETECTION

M.Sc. THESIS

Aji MUBALAIKE
(707151002)

Department of Applied Informatics

Information Security Engineering and Cryptography Programme

Thesis Advisor: Prof. Dr. Ertuğrul KARAÇUHA
Thesis Co-Advisor: Prof. Dr. Eşref ADALI

DECEMBER 2018

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

DIŞ SALDIRILARIN BELİRLENMESİ İÇİN MAKİNE ÖĞRENİMİ YAKLAŞIMI

YÜKSEK LİSANS TEZİ

Aji MUBALAIKE
(707151002)

Bilişim Uygulamaları Anabilim Dalı

Bilgi Güvenliği Mühendisliği ve Kriptografi Programı

Tez Danışmanı: Prof. Dr. Ertuğrul KARAÇUHA
Eş Danışman: Prof. Dr. Eşref ADALI

ARALIK 2018

Aji Mubarek MUBALAIKE, a **M.Sc.** student of ITU Informatics Institute student ID **707151002**, successfully defended the **thesis** entitled "**MACHINE LEARNING APPROACH FOR EXTERNAL FRAUD DETECTION**", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Ertuğrul KARAÇUHA** ....
İstanbul Technical University

Co-advisor : **Prof.Dr. Eşref ADALI** ....
İstanbul Technical University

Jury Members : **Prof. Dr. İbrahim SOĞUKPINAR** ....
Gebze Technical University

Doç. Dr. M.Oğuzhan KÜLECİ ....
İstanbul Technical University

Dr. Öğr. Üyesi Şerif BAHTİYAR ....
İstanbul Technical University

Date of Submission : 16 November 2018
Date of Defense : 14 December 2018

To my family and friends,

FOREWORD

Foremost, I would like to thank my supervisor Prof. Ertuğrul Karacuğa for his supervision and fruitful discussions on the problem of learning in non-stationary environments during my research with Prof. Eşref Adalı as my co-supervisor.

A special thanks to my co-supervisor Eşref Adalı from deep of my heart. I would like to express my compete satisfaction and sincere gratefulness for his tolerance, enthusiasm, and encouragement. Words cannot verbalize how appreciative I am to have you as my mentor, for incentivizing and inspiring me to sustain myself to survive all the stress from my master journey and not letting me give up. I could not have envisioned having a better supervisor for my master study.

Finally, yet importantly, I would like to deliver exceptional gratitude to my family who always believed in what I was doing for their endless support and understanding. With blessings from my family stimulated me to reach this point in life. Showing appreciation for them is the tiniest I can do but I can venerate and extol their dedication and love by trying to convey my nonpareil best at every stage of life.

December 2018 Aji MUBALAIKE

TABLE OF CONTENTS

Page

ABBREVIATIONS

A/V	: Anti-Virus
ACL	: Access Control
Adam	: Adaptive Moment estimation
AI	: Artificial Intelligence
AIDS	: Anomaly-based Intrusion Detection System
ANN	: Artificial Neural Network
API	: Application Program Interfaces
CIA	: Confidentiality-Integrity-Availability
CIC	: Canadian Institute for Cybersecurity
CNN	: Convolutional Neural Network
CPU	: Centural Processing Unit
DBN	: Deep Blief Network
DDOS	: Distributed Denial of Service
DL	: Deep Learning
DMZ	: Demilitarized Zone
DNN	: Deep Neural Network
DNS	: Domain Name System
DOS	: Denial of Service
DR	: Detection Rate
DT	: Decision Tree
DVF	: Discrete Vector Factorization
ENIAC	: Electronic Numerical Integrator and Computer
FAR	: False Alarm Rate
FN	: False Negative
FP	: False Positive
FTP	: File Transfer Protocol
GBC	: Gradient Boosting Classifier
GNU	: Gnu’s Not Unix
GRU	: Gated Recurrent Unit
HIDS	: Host-based Intrusion Detection System
HTTP	: Hypertext Transfer Protocol
IBM	: International Business Machines
ICMP	: Internet Control Message Protocol
IDS	: Intrusion Detection System
IP	: Internet Protocol
IPS	: Intrusion Prevention System
IT	: Information Technology
KDD	: Knowledge Discovery and Data Mining
KNN	: K-Nearest Neighbors
LR	: Logistic Regression
LSTM	: Long-Short Term Memory
MIME	: Multipurpose Internet Mail Extensions
ML	: Machine Learning
MLP	: Multi-Layer Perceptron

NAN	: Not A Number
NB	: Naive Bayes
NIDS	: Network Intrusion Detection System
NMap	: Network Mapper
NSCS	: Neurosynaptic Core Simulator
PC	: Personal Computer
PCA	: Principal Component Analysis
PNN	: Probabilistic Neural Network
PSO	: Particle Swarm Optimization
R2L	: Remote-to-local Attack
RBM	: Restricted Boltzmann Machines
RF	: Random Forest
RNN	: Recurrent Neural Network
ROC	: Receiver Operating Characteristic
ROC-AUC	: Area Under the Receiver Operating Characteristic Curve
RSA	: Rivest–Shamir–Adleman
SAE	: Stacked Auto Encoders
SAINT	: Securiy Administrator’s Integrated Network Tool
SCADA	: Supervisory Control and Data Acquisition
SDN	: Software-Defined Networking
SMOTE	: Synthetic Minority OverSampling Technique
SMR	: Soft-Max Regression
SMTP	: Simple Mail Transfer Protocol
SQL	: Structured Query Language
SSH	: Secure Shell support
SSL	: Secure Sockets Layer
STL	: Self-Taught Learning
SYN flood	: Synchronize flood
TCP	: Transmission Control Protocol
TL	: Turkish Lira
TLS	: Transport Layer Security
TN	: True Negative
TP	: True Positive
U2R	: User-to-root Attack
URI	: Uniform Resource Identifier
UBPS	: User Behavior Prediction System
UDP	: User Datagram Program
UNIX	: Uniplexed Information and Computing System
URI	: Uniform Resource Identifier
URL	: Uniform Resource Locator
USB	: Universal Serial Bus
UUCP	: Unix-to-Unix Copy
WC	: Warezclient
WM	: Warezmaster
WNA	: Wireless Network Aubur

LIST OF TABLES

LIST OF FIGURES

MACHINE LEARNING APPROACH FOR EXTERNAL FRAUD DETECTION

SUMMARY

If we take a very brief timeline of noteworthy fraudulent incidents, there are hundreds of incidents over the last few decades. But one thing we may not be aware of is the first computer virus in 1971, after the invention of the first electronic general-purpose giant computer ENIAC in 1945, the very first computer virus known as the Creeper virus came to exist. From there, 1991, Michelangelo virus, which was designed to infect DOS systems, was perceived as digital apocalypse at that times. Fortunately, it did not have as much of an impact as people were kind of screaming about, but it was really one of the first widespread viruses that everyone started to kind of learn about and be a little bit fearful of. Then forge ahead a few years, we had the Melissa worm in 1999. That was an email-based worm targeting Microsoft Outlook spreading at an excessive speed. Let us fast forward one more year to the "ILoveYou" worm which was very similar in nature and scope and also one of the most damaging, quickly replicating and spreading fraudulent incident of all time. Again back in 2000, within the first few hours of that kind of worms release it infected and spread to millions of computers around the world. Forward another decade, the malicious computer worm uncovered in 2010, Stuxnet, which was generated to attack SCADA systems causing substantial damage to Isan’s nuclear program. It was initially intruded into the network via an infected USB drive, and from there it quickly mapped the internal network mapping out internal resources and so forth. But the operators never knew they were spinning out of control because the alarms were disabled. In these way, it destroyed a large piece of the infrastructure around that nuclear facility setting their program back many years. Fast forward a few more years to 2013, we had the advent of the Cryptolocker virus causing millions of dollars in loss to various companies. When Cryptolocker virus is activated, using cryptography name RSA public-key, malware encrypts definite categories of document files deposited on drives of a local network, with the private key deposited only on the control servers belonging to malware. Intruders can only decrypt the data if the payment they required is made by the expressed deadline, threatening victims to delete the private key if the deadline passes. And then lastly, fast forward to more or less present time, 2016 we had the Locky ransomware, which is very similar to Cryptolocker, and that has over 60 different derivatives of that specific piece of malware, exposing financial havoc on any number of systems, companies large and small, law enforcement agencies, and so forth. So fraudulent incidents as malware, whether it would be viruses, worms, and so forth, have been around for over four decades. So it is extremely necessary to detect or prevent these kinds of large-scale breaches efficiently.

Generally speaking, it is also necessary to consider the incredible cost of malware infections. In 2014, a few years ago, $491 billion were spent on the recovery of malware infections and $25 billion spent by the consumers as a result of security threats. That is an incredible number to try to wrap our arms around, but it has a huge

impact on the economy at large on a global scale. And something that is even perhaps a little more interesting is the fact the specialists spent 1.2 billion hours dealing with the after-effects of malware and malware infections. That is a lot of time obviously. So fraud is not just an annoyance. It is big business, and it costs a lot of money to companies and to consumers to combat malware infections. So it is not just hackers, and it is not just script codes trying to inconvenience people. It is actually criminal organizations; it is a very organized and intentional process. Whether it is Cryptolocker and ransomware, whether it is stealing information, proprietary secrets and competitive advantages from inside of companies, and so forth, it is a really big business and it costs a lot of money of companies and consumers to combat malware infections. So when it comes to positioning our organization or IT infrastructure and so forth, to be in the best position to ward off malware infections and to perhaps prevent from occurring in the first place, it is very important to understand how malware can affect the related PC or the security systems like IDS, how it can get into our network, how it can affect our organizations, and then the things we need to do. It is vital that everyone understands the nature true of this threat and take ideal measures to mitigate or minimize the risks.

In addition to these staggering cost of fraudulent incidents, detection and prevention of all kind of malware should be taken into consideration as fast as possible in an efficient way. As we all know, as far as anti-virus and anti-malware software is considered fairly effective. But when we install A/V software, it should be kept updated and also needs us to take the precautions. It couldn't prevent the security system from getting hacked or intruded like a firewall. Other cost-effective countermeasures designed to detect, prevent or block fraudulent malicious activities all over the network could be intrusion detection and prevention systems. After identifying abnormal traffics, IDS or IPS would write to log files when suspicious activity is detected, then would send event notifications taking preventative measures. However, some kind of destructive drawbacks, like misclassification of genuine traffics as anomalies, and incompetence to configure unknown attacks, make intrusion detection and prevention systems run inefficiently.

All mentioned striking evidence lends support to the view that we determine to use the combination of two different type of IDSs, identifies as network-based IDS and anomaly-based IDS for the methodology of this research. Network-based IDSs are positioned within the network to mainly detect abnormal malicious traffics by examining passing network transactions. Anomaly-based IDSs is also responsible for the unknown attack traffics, it could detect unknown external frauds, developing non-signature-based IDSs. A number of factors of this combined IDSs could contribute to the success in detecting external unknown frauds that have not been identified previously and minimizing false positive rate. It is indisputable that, machine learning technique which is the subset of artificial intelligence have gained significant awareness in the past few decades. With the contributions of machine learning techniques, we can analyze a tremendous amount of network traffic data with high performance in a short time, and generate reliable external fraud detection and classification model. Taking into account all these factors, we safely plan to present a comprehensive review of external fraudulent attacks and corresponding detection systems and also demonstrate a set of experimental works analyzing the execution of supervised machine learning techniques.

DIŞ SALDIRILARIN BELİRLENMESİ İÇİN MAKİNE ÖĞRENİMİ YAKLAŞIMI

ÖZET

İlk bilgisayarlar kendi başlarına çalışan bilgisayarlardı ve dış dünyaya bağlantıları yoktu. Bu nedenle güvenlik ile ilgili bir sorunları yoktu. Zaman içinde bilgisayarlar ağları, İnternet ve telsiz ağlar bilgi ve bilgisayar güvenliği için tehlikeli tehdit olmaya başladılar. Özellikle İnternet bilgi ve bilgisayar güvenliğinde önemli açıkların doğmasına neden olmuştur. Bunun başlıca nedeni, İnternet'in kapalı bir ağ yapısının (ARPANET) herkese açık biçimde uygulanmaya konmasıdır.

Bilgisayar sistemlerine yapılan saldırılar genel olarak iki sınıfa ayrılmaktadır: İç saldırılar ve dış saldırılar. Bu tez kapsamında dış saldırıların belirlenmesi üzerinde çalışılmıştır.

Bilgisayarlara ilk saldırı 1971'de Creeper virüsü ile yapılmıştır. 1985'li yıllarda bireysel bilgisayarların yaygınlaşmaya başlamasıyla, bu bilgisayarlara yönelik saldırılar görülmeye başlamıştır. 1991 yılında geliştirilen Michelangelo virüsünün amacı DOS işletim sistemini bozmaktı. Daha sonra üretilen Melissa (1999) I Love You (2000) çok yayılmış ve etkili olmuş saldırı örnekleridir.

Saldırılar yalnızca sunucu ve bireysel bilgisayarlara yönelik olmamakta, sistemlere karşı da yapılmaktadır. Örneğin İran'ın nükleer çalışmalarını engellemek amacıyla üretilen Stuxnet SCADA sistemine ciddi hasarlar vermiştir (2010). USB bağlantısı üzerinden SCADA sistemine bulaştırılan virüs sistemin bütün kaynaklarını ele geçirmiş ve alarm sistemlerini devre dışı bıraktırmıştır. İşletmenler durumdan haberdar olmadıkları için ne olup bittiğini anlayamamış sonuç olarak tesiste ciddi zararlar oluşmuştur.

2013 yılında ortaya çıkan Cryptolocker virüsü sunucularda tutulan dosyaları şifreleyerek kullanımını engellemiştir. Engelin kalkması gereken anahtar daha sonra para karşılığı veriliyordu.

Rus savaş uçağının düşürülmesine misilleme olarak Türk kurum ve kuruluşlarını hizmet veremez duruma sokmak için yapılan saldırılar yakın zamanda görülmüştür.

Bilgi sistemlerine zarar vermeye yönelik olan bu programlardan korunmak için harcanan paranın 2014 verilerine göre 25 Milyar ABD Doları ve bu yazılımların verdiği zararın 491 Milyar ABD Doları olduğu göz önüne alındığında bilgi sistemlerine yapılan dış saldırıların ne denli önemli bir konu olduğu açıktır.

Dış saldırılar amaçları açısından sınıflandırıldığında;

1) Sisteme zarar vermek,
2) Sistemin çalışmasını engellemek ve
3) Menfaat sağlamak olarak sınıflandırılabilirler.

İran SCADA sistemine yapılan saldırı birinci sınıfa girmektedir. Türk kurum ve kuruluşlarına karşı yapılan DDos saldırıları ikinci sınıfa girmektedir. Üçüncü sınıfa giren saldırılar için çok sayıda örnek verilebilir. Bunların içinde bankalara yönelik saldırılar, müşteri hesaplarından para çalmak en yaygın görülenlerdir.

Bu tez çalışmasının kapsamı dışarıdan gelebilecek tehdit ve olası saldırıların incelenmesi ve ortaya çıkarılmasıdır. Bu hedefe ulaşmak üzere öncelikle dış saldırılar incelenmiş ve bunların verebileceği zararlar nitelik ve nicelik açısından değerlendirilmiştir. İkinci aşamada, dış saldırıların nasıl belirlenebileceği üzerinde durulmuştur. Tehdit ve saldırıların belirlenmesi amacıyla geliştirilmiş yöntem ve algoritmalar incelenmiştir. Üçüncü aşamada, dış saldırılara ilişkin veri kümesi oluşturulmaya çalışılmıştır. Dış saldırılara ilişkin olarak önce PaySim mobile Money simulator veri kümesi üzerinde çalışılmıştır. Ardından NSL-KDD veri kümesi üzerinde çalışılmıştır. Ancak bu iki veri kümesi yeterli görülmemiştir ve Canadian Institute for Sybersecurity'nin hazırladığı veri kümesine geçilmiştir. Her üç veri kümesi üzerinde altı algoritma denenmiştir. Bu algoritmalar, K-Nearest Neighbor (KNN), Random Forest (RF), Adaboost, Logistic Regression (LR), Multinominial Naive Bayes (MNB), Stochastic Gradient Discent (SGD). Denemelerimizin sonucunda RF algoritmasının en başarılı sonucu verdiği görülmüştür. Karşılaştırmalar doğruluk ve F1 ölçüsü hesaplanarak yapılmıştır. Aynı veri kümesi için doğruluk değerleri söz konusu yöntemler için 100 üzerinden RF: 100, AdaBoost: 99,99, KNN: 99,97, LR: 98,1, MNB: 96,79 ve SGD: 96,87 olarak bulunmuştur. F1 ölçüsüne değerleri ise 100 üzerinden şöyle bulunmuştur: RF: 99,97, AdaBoost: 99,85, KNN: 99,64, LR: 62,35, MNB: 49,03 ve SGD: 32,80.

En iyi algoritmanın belirlenmesinin ardından, algoritmanın daha hızlı çalışmasını sağlamak amacıyla özellik seçimine geçilmiş. Veri kümesinde 79 olan özellikler 14'e indirilmiştir. Seçilen özellikler şunlardır:

- Hedef adresi - Destination Port,
- İlk_Pencere_byte_ileri_- Init_Win_bytes_forward,
- İlk_Pencere_byte_geri_- Init_Win_bytes_ backward
- Akış IAT Enk - Flow IAT Min,
- İleri IAT Enk - Fwd IAT Min,
- Geri IAT Enk - Bwd IAT Min,
- Ortalama Paket Boyu - Average Packet Size,
- Geri Paket Uzunluğu Std - Bwd Packet Length Std,
- İleri Paket Uzunluğu Std - Fwd Packet Length Std,
- Paket Uzunluğu Std - Packet Length Std,
- Toplam Geri Paketler - Total Backward Packets,
- Toplam Geri Paketlerin Uzunluğu - Total Length of Bwd Packets,
- İleri_Enk Seg_Boyu - Min_seg_size_forward,
- Etiket - Label

Bu işlemlerin sonunda seçilmiş özelliklere kullanılarak, değişik algoritmaların başarımları bulunmuş ve sonuçlar diğer araştırmacıların bulduğu değerler ile karşılaştırılmıştır. Karşılaştırmalar sırasında her yöntem için Bulma, Tutturma ve F1 olcusu hesaplanmıştır. Değerler yüzde cinsinden verilmiştir: Diğer Araştırmacıların Sonuçları DAS ve Tez Çalışmasının Sonuçları TÇS olarak kısaltılmıştır:

Tutturma - DAS: KNN: 96, RF: 98, AdaBoost: 77, NB: 88, MLP: 77 ve ID3: 98

TÇS : KNN: 99,9, RF: 99, AdaBoost: 99,9, LR: 95,8, MNB: 66,9 ve SGD: 96,8

Bulma - DAS: KNN: 96, RF: 97, AdaBoost: 84, NB: 04, MLP: 83 ve ID3: 98

TÇS : KNN: 99, RF: 99,4, AdaBoost: 97, LR: 97, MNB: 66,8 ve SGD: 89

F1 olcusu- DAS: KNN: 96, RF: 97, AdaBoost: 77, NB: 04, MLP: 76 ve ID3: 98

TÇS : KNN: 99,6, RF: 99,9, AdaBoost: 99,8, LR: 62,3, MNB: 49 ve SGD: 32,8

Sonuç olarak, bu tez çalışmasında elde edilen sonuçların, diğer araştırmacılar tarafından yapılan çalışmalara oranla daha başarılı olduğu gösterilmiştir.

1. INTRODUCTION

When asked about fraud, the vast majority of people claim that it is a hackneyed subject that has been existing for a long time until now to restrict the burgeoning of informatics and pose hideous threats to the security all around the world. The general definition of fraud is an intention to accumulate wealth or gain personal reputation lawbreaking, like cash, intellectual property or any vital information, that dishonestly perpetrated by one or more individuals [1]. There is a general discussion nowadays over the computer fraud that can be considered as an adulteration or counterfeiting activities by the employee, colleagues or any third party with the fraudulent aim to possess detrimental benefit to violate against the integrity, availability, and confidentiality of vital knowledge of data.

Computer security is the confidentiality, integrity, and availability of information alluded to as the CIA triad in Figure 1.1 or information security triad [2]. All the principles, mechanisms and standards we will encounter in the security domain are dedicated to these three abstract but prohibitively fundamental goals of information and information processing resources. The CIA triad is described in more detail as follows.

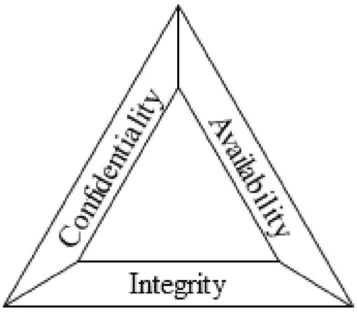


Figure 1.1 : CIA triad of information security.

Confidentiality: Detection of unaccredited disclosure of vital information. In other words, the security organizations ensure unaccredited users do not expropriate or duplicate the private information [3]. A Handsome example of encryptions, that ensure only authorized people can access the information, would be SSL or TLS.

They are prominent security protocols for internet communications to ensure security. Social security information like credit card number is the most common case that hackers can easily steal and disclose.

Integrity: Detection of uncertified modification of perishable key information. The most frequent example is that if we were transferring 100 TL but the related information was changed to 10,000 TL, it could be a big cost. One of the common countermeasures is hashing the original data by using advanced hybrid-encryption techniques, like GNU Privacy Guard.

Availability: Detection of unauthorized withholding of needed information. It ensures authorized organizations can gain access to the valued information when they needed. Some attacks like Denial of service (DoS) is the most widespread network attacks that interrupt the normal use of system's resources, targeting the availability of CIA triad and make system information assets unavailable to the legitimate users as is depicted following in Figure 1.2.

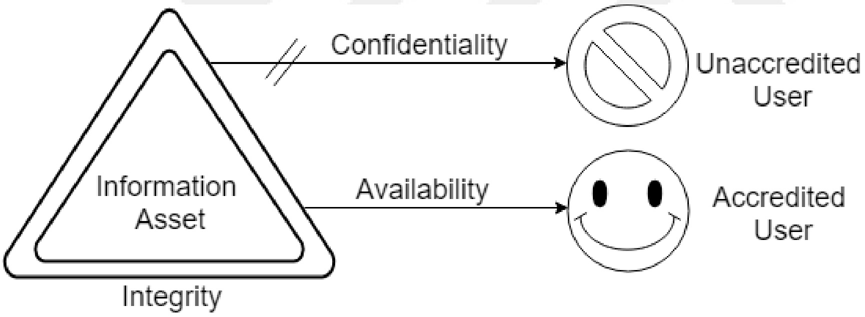


Figure 1.2 : A graphical presentation of CIA triad.

In general, fraud can be classified as an external and internal fraud based on the relation of the perpetrator to the organizations [4]. As is illustrated from the figure of fraud taxonomies in Figure 1.3, more specifically, fraudulent acts or attacks can be originated from within an organization or from outside of the organization. An internal user, such as employees, ex-employees that are retired or resigned, contract staff or trusted partners, can accidentally or intentionally tamper confidential data and also threaten the operations of internal servers or mishandle network infrastructure devices. While external frauds perpetrated range from by amateurs to skilled attackers, like the cybercriminals, activists, terrorists or hackers, are causing prohibitively high priced costs for different kind of organizations or countries, like identity theft, the space race, mass transactional incidents, network intrusions, cybercrime, and so forth. For instance, they facilitate outside attacks by connecting infected USB media into the corporate computer systems, based on the historical virus SCADA logics, or accidentally invite malware into the network through malicious email or websites. One nationwide investigation concluded that hackers are generally categorized by three types, blackhat, grey hat, and white hat. There are two premier aspects that determine the category of hacker we are dealing with: their incentives, the other is whether or not they are disturbing the law.

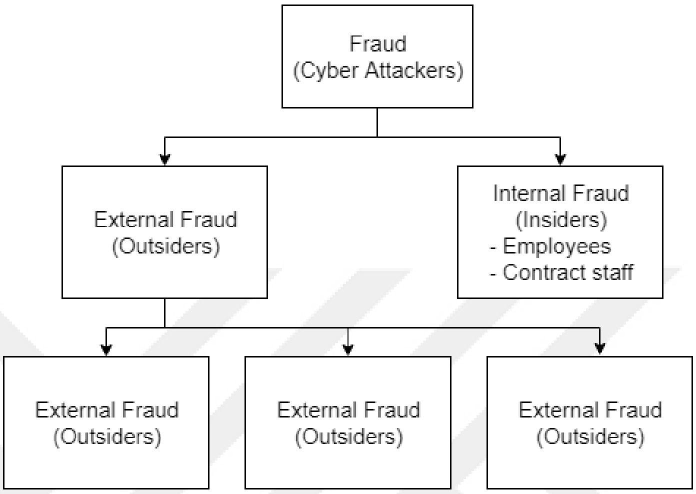


Figure 1.3 : Fraud taxonomies and their perpetrator.

Blackhat hackers: Black hat hackers, who range from amateurs to experienced hackers, usually have considerable knowledge about writing malware to steal, modify or destroy crucial data like financial information, personal information or login credentials from the security systems, also bypass the security protocols. Their hostile intentions are usually for personal or financial gain, 12 pt (before) and 6 pt (after) paragraph spacing must be set. Table captions must be ended with a full stop. A table and its caption must be on the same page.

Grey hat hackers: Grey hat hackers can be defined as a conglomeration of both black and white hat demeanors. In spite of considering illegal, they still strike to the vulnerabilities of a system without the owner's permission or authorization [5]. After destroying to the intentioned a piece of information, they will report them to the owner, requesting a gigantic amount of money to fix the issue, just a bit like ransomware. If the owners refuse to comply, gray hat hackers threat them by exploiting the private information online for the world to see.

White hat hackers: White hat hackers described ethical or moral hacker, are known as specialists who use their hacking bits of knowledge for good to find security holes.

Getting permission from the authorized members of the system makes the process completely legal [6].

In this research, we focus on detecting and preventing external frauds mainly on network traffic attacks that posing extremely considerable threats day by day. Figure 1.4, illustrates the building process of a fraud detection model. Firstly, all original raw data goes to show that could be of prospective usefulness. All that collected data will then be gathered and cleaned in a data warehouse. Some prominent exploratory analysis using a different kind of machine or deep learning techniques are used. Taking into account all above process, a methodical model will then be meticulously approximated from the preprocessed and engineered data. Once the model has been generated, fraud experts will elucidate and explicate the proposed classification model.

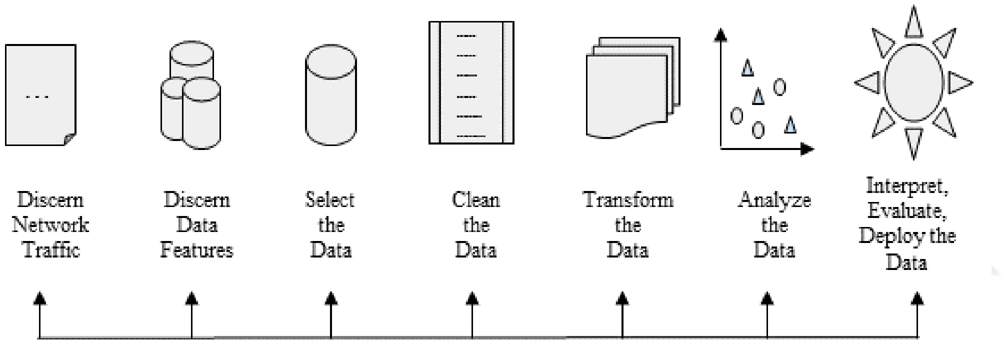


Figure 1.4 : The process of fraud analytics.

1.1 Literature Review

With the expeditious pervading of external fraudulent transactions in the network environment, fraud researchers in cybersecurity domain are trying to exert themselves to dissertate the challenges of cyber and network security using multifunctional techniques like various type of machine learning and data mining algorithms efficiently. Let us review some contemporary and pre-eminent researches to acquire some noteworthy contributions with their pros and cons.

In a study by Jihyun [7], the authors evaluated IDS classifier on which apply the deep learning algorithms name LSTM-RNN. After doing some feature engineering to the KDD Cup 99 dataset, they extracted a new dataset trying the find ideal size of hidden layer and learning rate. After comparing carefully the advantages and disadvantages of other classifieries, they discovered that LSTM-RNN classifier can detect network attacks with the highest performance in detection accuracy.

McLernon [8], tried to detect network traffics and estimated a NIDS model on which implementing deep learning algorithm. After differentiating the experimental results with other classifiers, authors came to the conclusion that flow-based anomaly detection system can run in high performance using deep learning. By the deep learning intrusion detection module, a small amount of network traffics test data can still be detected successfully.

Nataraj, Karthikeyan [9], authors executed a new viewpoint that is about an analysis of malware by the images with which visualizing and processing malware. Striking preliminary results are obvious that among 9,458 samples with 25 different malware families in a dataset, they received high classification accuracy of approximately 98%. Undoubtedly, authors came to the remarkable conclusion that computer vision techniques could contribute unmistakably for the analysis of malware.

Research, represented by Wang [10], uses deep learning techniques including deep belief and deep coding methods to detect anomalies and identify fraudulent traffics. Although the used a gigantic sized of a deeper dataset, they still get the higher accuracy than other machine learning techniques did.

Raffie [11], authors of this research focused on the false positive rate in network intrusion detection systems, approaching a different type of machine learning techniques. Clearly all the analysis they made leads to unshakable consequences that the machine learning algorithms contribute to maximizing the false positive rate of NIDS. Finally, the network traffics are prevented and protected successfully from the intrusions.

Guangzhen [12], the writer presented an intrusion detection model based on DBN and PNN, after experiencing a copy of obstacles like irrelevant information in a big quantity of data, prolonged training time, undemanding local optimum that quickly fall into. He reduced the dimensions of original raw data to low-dimensional data using PNN algorithm. Furthermore, the author applied a PSO algorithm motivated by increasing the accuracy of the DBN network model and optimizing the number of hidden layer nodes. Considering all the inspirations deduced, he safely draws the conclusion that incorporation of deep learning, PSO, and PNN algorithms are considerable efficient, also supply reliable solutions for the mentioned obstacles.

According to research presented by Zahangir [13], a new approach of deep learning techniques, for detecting and classifying network intrusions, is demonstrated on the dataset of KDD Cup 99. All aspect of protecting cybersecurity implements on a hardware platform related to energy efficient neuromorphic. From what has been proved on all experimental tests, it is clearly observed that the empirical tests resulted in the accuracy of approximately 81.31% and 90.12% for purpose of detection and classification of network intrusions.

When it comes to a research, proposed by Nguyen [14], the majority of all implementation is relating to anomaly-based NIDS using deep learning techniques that include stacked autoencoders and restricted Boltzmann machine. The advantages of SAE carry more weight than RBM with the performance on accuracy and precision. The main idea is trying to detect network intrusions and classify the results in four groups according to their accuracy rate. Since SAE includes too much perplexing computations, although SAE performs an enormous advantage on performance matrix of accuracy, it can not be compared with RBM in the total length of consumed time for generating detection models. RBM classifiers consume less time than the other classifiers.

Vinayakumar [15], proposed a number of factors that could contribute to the effectiveness of CNN, which is one branch of deep learning techniques. IDS endeavor to achieve and construct the events of network traffic according to the time series of TCP/IP packets. Intrusions on ICT networks are variegated and constantly advancing without interruption. The author conducted a survey evaluates the effectiveness of different kind of deep learning methods, making convolutional neural network be the first layer instead of a recurrent neural network. From what has been examined, it is obviously clear to observe the result that the accuracy of CNN outweighed than other experiment results, like RNN.

A recent study conducted by Norbert [16], researched how to successfully discern abnormal transactions in a network environment using NNIDS detection system. The traffic attack types consist of UDP flood, SYN flood, nmap scan, also including genuine network transactions. The preferred detection system can, fortunately, distinguish identified traffic attacks, not only a single but also concurrently intruded several attacks.

Valentina [17], demonstrated a comparative research on the contemporary machine learning methods for skewed datasets named UNSW-NB15. Applied machine learning techniques are used to generate the most reliable intrusion detection classifier, that includes AdaBoost, LogitBoost, BaggedTree, RUSBoost, and GentleBoost. Taking into account all accuracies of mentioned classifiers, the author had stipulated certain results that RUSBoost outperformed all other detectors, while Bagged tree and GentleBoost classifiers have fairly high performance as well.

Chuanlong [18], presented deep learning based intrusion detection model, identified as RNN-IDS. A contrast of other traditional machine learning classification techniques, just like naive Bayesian, decision tree J48 and random forest, the accuracy of RNN possesses much better performance and detection rate to distinguish the categories of network intrusion.

Ishita [19], emphasized with clarity that prospective hackers are strongly tending to launch network attacks, like identity theft or DDoS in back of Tor network, which is becoming a practical appliance for malignant users. Moreover, with the guidance of Tor, the users are assured not to deny the authenticity of computer security. In this research, one kind of deep learning techniques, deep recurrent neural network abbreviated as DRNNs, is implemented to estimate user behavior in Tor environment, with which assembling of a deep web browser and Tor server. Judging from all evidence offered, the author came to wield WNA to acquire data and DRNNs to make a prophecy of user manners. Obviously, all the evidence confirms the undoubted result that the combining model of UBPS-DRNN has reached a reliable estimation for almost all behaviors of users.

Among the most convincing contributions cited by the author, Georgi A. [20], one should be emphasized that flow-based IDS for SDN is adept to detecting abnormal traffic with superb performance. Since previously trained supervised classifiers do not need payload information, so they possess the authority of classifying encrypted flows. By comparison with other algorithms, the RF ensemble was adept at identifying different types of intrusions from real-time intrusions as well as detecting them successfully.

Dinh [21], demarcated to solve the imbalanced dataset problems for NIDSs using standard NSL-KDD dataset. Several stereotyped deep learning methods of SAE and DBN are demonstrated in a Tensorflow environment and provided improvements in the accuracy of detecting network attacks. Last but not least, graphical demonstrations have illustrated that the generated detection model combined with introduced deep learning methods could significantly detect and classify the most predominant attacks, named R2L and U2R, with a brilliant performance.

Kopelo, Devi [22], researched some outstanding surveys tending to Host-based intrusion detection and prevention systems. The testing data excluded from training data is originated to be processed after two engines, misuse detection after that anomaly detection phase. So-called HIDPS designated the most effective algorithms individually, one type of decision tree algorithm of C4.5 for misuse detection and support vector machine algorithms for anomaly detection. HIDP could detect and prevent all kind of fraudulent attacks, despite the resources came from external or internal.

Nathan [23], outlined the NDAE method that previously undiscovered for unsupervised machine learning. Two type of datasets, NSL-KDD and KDD 99, are applied combining with the RF and stacked NDAEs classification algorithms. They achieved fairly promising results offering decreased training time combined with high degrees of precision, accuracy, and recall. Particularly, they have differentiated the mainstream DBN technique against a stacked NDAE model. The contrasts between the mainstream DBN and stacked NDAE model have revealed that the generated NDAE model achieved a much greater result of accuracy rate about 98.81% rate and shorter time on training.

1.2 Contribution

From what has been discussed above about a brief history of frauds, we could safely come to the essential contribution of this research. All mentioned striking evidence lends support to the view that we determine to maneuver the combination of two different type of IDSs, network-based and anomaly-based IDS respectively, for the methodology of this research. Network-based IDSs are positioned within the network to mainly detect abnormal malicious traffics by examining passing network transactions. Anomaly-based IDSs is also responsible for the unknown attack traffics, it could detect unknown external frauds, developing non-signature-based IDSs. A number of factors of this combined IDSs could contribute to the success in detecting external unknown frauds that have not been identified previously and minimizing false positive rate. It is indisputable that, machine learning technique which is the subset of artificial intelligence have gained significant awareness in the past few decades. With the contributions of machine learning techniques, we can analyze a tremendous amount of network traffic data with high performance in a short time, and generate reliable external fraud detection and classification model. Taking into account all these factors, we safely plan to present a comprehensive review of external fraudulent attacks and corresponding detection systems and also demonstrate a set of experimental works analyzing the execution of supervised machine learning techniques. Following are the brief list of the contribution of this research:

1) To decrease the rate of false positives.
2) To have the potential to detect unknown or unidentified external fraudulent attacks.
3) To automatically detect network external attacks and fraudulent behaviors.
4) To initiate the integration of anomaly-based and network-based IDS methods.
5) To work on intrusion prevention as well as intrusion detection methods.
6) To react efficiently in order to maintain the highest possible level of security.
7) To apply real-time external traffic data but synthesized.
8) To generate classification detecting model combined with preeminent ML techniques.
9) To reduce the consumed training and testing time during the generating and evaluation phase.

1.3 Thesis Organization

This thesis is ordered as follows:

Chapter 2: The Second chapter starts with the discussion of different taxonomies of external fraud and network attacks, then address the theoretical information of countermeasures for external frauds. After that, some fraud analyzing tools will then be introduced with their strengths and weaknesses. Different detection approaches, like intrusion detection and prevention systems, that are contemporarily reliable and efficient to distinguish fraudulent activities like network transactions, are also described in detail.

Chapter 3: This chapter is about the theoretical foundation of deep machine learning approaches. According to this conceptual knowledge, we will try to select the best reliable detection technique with high accuracy. Four types of supervised machine learning techniques to intrusion detection research area are introduced in detail. Moreover, the relative cross-validation and evaluation metrics are also described.

Chapter 4: In this chapter, we initiate with the establishment of the dataset that we use, and demonstrate all the experimental results, like data preprocessing and feature engineering. At last, we will display the demonstration of different kind of external fraud detection classification models using most preeminent and efficient deep machine learning techniques.

Chapter 5: The last chapter configures the best classification model of detecting network traffics, then conclude our results and compare them with other generated classification models. Finally, we will also compare our IDS classifiers with the related previous work, then conclude the result and display future work.

2. BACKGROUND INFORMATION

As we move toward the digital economy, malefactors and lawbreakers exert themselves to discovering innovative and heterogeneous ways to execute fraud against any kind of information systems. The deprivation due to unaccredited credit card transactions alone is approximated to be trillion of lira's each year. Hence, it is necessary to consider the incredible cost of malware infections measured in dollars here. In 2014, a few years ago, $491 billion were spent on the recovery of malware infections and $25 billion spent by the consumer as a result of security threats. That is an incredible number to try to wrap our arms around, but it has a huge impact on the economy at large on a global scale [4]. In addition, something that is even perhaps a little more interesting is the fact the specialists spent 1.2 billion hours dealing with the after-effects of malware and malware infections. That is a lot of time obviously. So fraud is not just an annoyance. İt is big business, and it costs a lot of money to companies and to consumers to combat malware infections. So it is not just hackers, and it is not just script codes trying to inconvenience people. İt is actually criminal organizations; it is a very organized and intentional process. Whether it is Cryptolocker and ransomware, whether it is stealing information, proprietary secrets, and competitive advantages from inside of companies, corporate espionage, and so forth, it is a really big business and it costs a lot of money of companies and consumers to combat malware infections. So when it comes to positioning our organization or IT infrastructure and so forth, to be in the best position to ward off malware infections and to perhaps prevent from occurring in the first place, it is very important to understand how malware can affect the related PC or the security systems like IDS, how it can get into our network, how it can affect our organizations, and then the things we need to do [24]. It is vital that everyone understands the nature true of this threat and take ideal measures to mitigate or minimize the risks.

This chapter discusses general taxonomies of network attacks that fraud researchers encounter with constantly, including the precise explanation of some of the more widely known attacks, perpetrated by external and internal fraudulent activists, then introduce some existing detecting techniques and their types one by one in detail. Figure 2.1 depicts explicitly the general picture of network intrusion prevention and detection systems and their classifications.

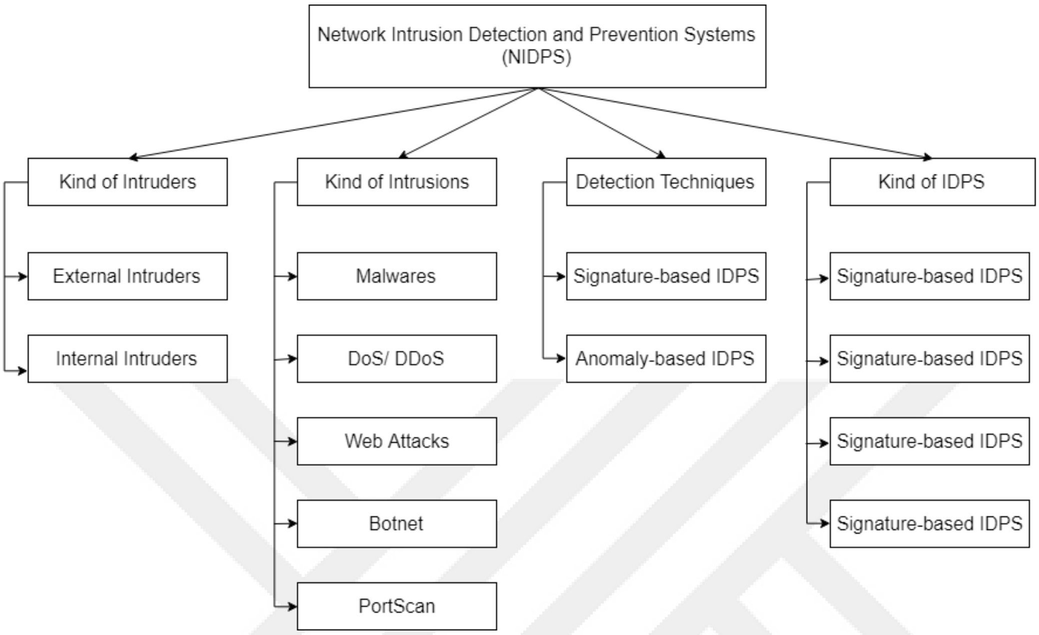


Figure 2.1 : Classification of NIDPS..

2.1 External Frauds

A malicious software, that has a fraudulent purpose to exploit a vulnerability, belongs to a malware [25]. There are different taxonomies of malware described below, Figure 2.2 depicts the brief timeline of all noteworthy malware and network attacks.

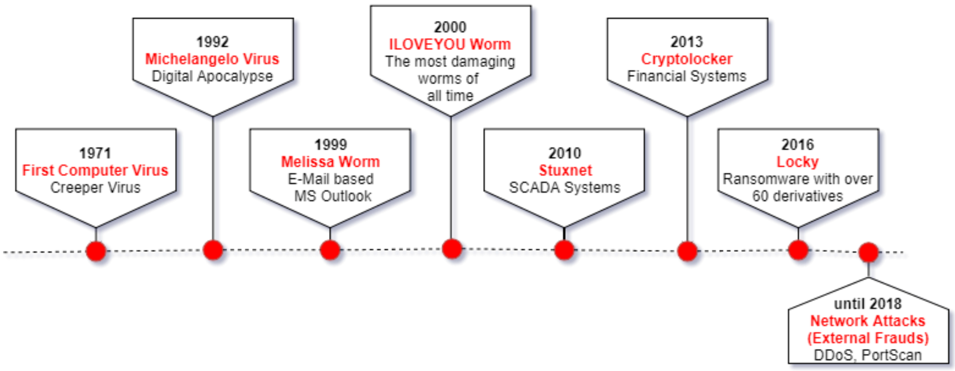


Figure 2.2 : Brief timeline of noteworthy network attacks.

Infiltration: Infiltration provides a concrete instance of malicious software, including viruses, adware, spyware, worms, trojan horses, logic bombs and ransomware, that endeavour to invade or devastate the administrator's computer. The intruders implement network infiltration by devastating vulnerable computer softwares, like Adobe Reader, Apple iTunes and Internet Explorer, etc. After successfully executing the intrusion of infiltration, a backdoor will be conducted on the victim's computer and can execute various network intrusions on the victim's network such as port scanning, IP sweep and network enumerations using Nmap, which will be intruduced deeply later.

Viruses: A virus can be a malicious code appending to a host engine. It can be easily distributed when the impaired program is implemented. Viruses can be replicated and implemented by itself. A virus can communicate in a widespread way. For example, it can be downloaded to local host indirectly as part of files downloaded as e-mail attachments under the internet.

Adware: An adware is ignored by most anti-malware applications since it generally does not mistreat the asset or operating system. Adware bothers any normal users by snatching their screen for.

Spyware: A spyware starts out as a companion to adware, harvesting information about user's web browsing habits to sell to advertisers leading to identity theft.

Worms: Unlike from viruses, worms do not append with a local file, however, spread expeditiously through computers and the Internet. A well known denial-of-service attack can be generated by spreading worms, overburdening email servers.

Trojan Horses: A Trojan can be considered as a programming code that intentionally contains various type of abnormal functions, putting itself in a hiding place of a useful program. Trojan horses can make copies of themselves and can be launched by intruders to exploit attacks on a system.

Logic Bombs:This kind of malicious code is placed hiddenly in an application and instigated by a logical incident, such as a specific date or time period.

Ransomware: A ransomware is a malicious software that limits access to some part of our system then demands a ransom to get it back. The first ransomware masquerades as antivirus software falsely reporting problems that they could fix if victims buy their product. Recently, ransomware became bolder as they render common document types useless and demand ransom payment to restore access to these files, a ransom increases in value if victims do not pay quickly enough.

Probes: Generally probes accumulate information by scrutinizing and scanning computer networks, they can be the reason that future attacks come to exist. The main aim of this accumulated information is discovering services and computers exist in a network to distinguish that kind of attack based on known vulnerabilities. On this stage. Let us briefly introduce some available scrutinizing tools applied for network probing.

IPSweep/ PortSweep: The Portsweep is manipulated to scan the port address of a specific computer which is disclosed in a subnet, while The IPSweep attack ascertains the host address which is opening on a subnet through a sweep of a close observation. After acquiring the types of service and the processed host, the accumulated information can be exploited by intruders to search for unprotected computers. If the intrusion is performed in a linear single-sourced style, Detecting PortSweep or IPSweep attacks in progress is fairly effortless than in a non-linear multiple-hosts-sourced style.

Nmap: Nmap generally displays IP, firewall, and port scans and operating system fingerprinting which is exploiting raw IP packets directed at victim computers. All these utilities are open sourced and free, that ports can be scanned orderly or haphazardly. Some detrimental factors, like the multiple-sources distribution and slow-scheduled intrusions, makes the probing conspiratorial over a long time period. Hence, detecting scans conducted by NMap is considered as strenuous.

Mscan: MScan employs DNS brute force scanning and zone transfers over the entire domains and whole scaling for IP addresses to probe distinguished computers on processing for familiar vulnerabilities of different network services such as imapandfinger, pop3, statd and cgi-bin programs. Different key signatures are existing for MScan attack detection, based on being probed of target computers and flaw.

Saint: SAINT is not an attack device. SAINT accumulates a big volume of networking information, such as FTP, telnet, finger, statd, tftp, and some other services. SAINT supplies three modes of non-compulsory behaviors, identified as light, normal, and heavy modes. Within the light mode, SAINT scrutinizes the objective computer for distinguishing DNS vulnerabilities and in addition unsafe NFS mount points. While in normal mode, it distinguishes vulnerabilities of boots, displays port scans on some common TCP ports such as FTP, UUCP, UDP, and HTTP. A heavy mode is fairly comparable to normal mode, excluding many other ports, could also be scrutinized.

DoS: Abbreviated from Denial-of-service, attacks are the main type of attack that demolishes or hogs the normal use of system’s deposits, and are really demanding to distinguish them from normal attacks. Unlike others, it targets the availability of the CIA triad and changes a local resource not obtained to authorized users. In other words, by sending a server with tremendous contemporaneous requests make the server could not answer while making it incapable to respond to any legal requests. One of the breakthrough issues is that, after making requests to big amounts of bandwidth, that would be handicapped the server needs a tremendous network connection. The second issue is that they are uncomplicated to block. when the victim discovered that they are of intrusions, they can quickly block the IP addresses of the offenders. Then the offender discovered to make a network resource unavailable temporarily.

DDoS: Another related attack, which is the main type of network intrusions, sending copies of files to pervade all memory space of all local hard drive. DDoS attacks are more arduous to identify and distinguish from authorized requests than DoS attacks do. It generally arises in The software potentially include DDoS attacks, are placed and inaugurated from a large number of host computers, then activated concomitantly to destroy the target machine using botnets [26]. Denial of Service attacks is a consequential threat to system managers who can easily demolish any network by illegal traffic. To reserve against DDoS attack, security researchers need to learn comprehension knowledge like demonstrating blocking technology that capable of identifying all kind of vulnerable traffic.

Smurf: The Smurf Attack identified as an amplified attack which is one amazing branch of DDoS attack. The intruders send echo requests using a forged source address to the broadcast of third-party servers. In point of fact, proposed forged source address indicates the confirmed and authentic IP address of the victim. After receiving the third-party server’s request, they misleadingly believe that requisitions came from the victim by sending an echo respond. The victim's total Internet and network connection become demolished with replies received from all over the place. In a fundamental DDoS attack, the restricted factor was a bandwidth. In this kind of amplification attack, the network attacker meticulously selects requests that possess extremely enormous responses.

Brute Force FTP/SSH: This type of ubiquitous attack not only be used for decrypting passwords, but also for detecting concealed content and web page in a web application. While the main function of Brute-force attack runs only in attemption of decrypting the encrypted message, trying a great quantities of passwords to find the security administrator's parole or password.

Heartbleed: The increase in heartbleed bug mainly is owing to a serious vulnerability among the widespread library of OpenSSL cryptograph. Using this disadvantage, intruders try to read the moemory of the local system and steal the defended informations preserved by SSL/TLS encryption. It is generally demonstrated by forwarding a malformed request to a defenceless server in order to trigger the intruder’s response and detect it.

Web Attack: This attack types are very common in our daily life because the people now from all walks of the world are taking security seriously. We may cite a common example of web attacks. SQL Injection attack mainly derives benefit from security vulnerability, as an assailant can generate a string of SQL commands and then deploy them to oblige the database to respond the required information. Another type of web attack is Cross-Site Scripting (XSS) which is occuring when supervising instructor or developers don’t test their generated code meticulously to prevent the probability of malicious script injection.

Botnet: The terminology of botnet is generally perceived as malignant implication. Some copies of Internet-connected devices controlled by a botnet malevolent owner to perform various malicious activities, such as delivering spam, executing DdoS attacks, stealing protected data or permitting intruders acquiring access to the local devices and their connection.

Warezmaster: This category of attack, which appears in a circumstance where write authorization is allowed improperly, is abbreviated as WM. More specifically, the WM attack makes use of a misconfiguration on the FTP server. Almost all FTP servers support any unidentified FTP procedure which supply users gain access to document files without requiring to identify themselves to local server [27]. unidentified FTP is generally deployed to download or make use of publicly attainable files. After enabling unidentified login to the server, users can enter to the server with the username “unidentified” and provided password from the server itself. Conventionally, the FTP server is considered as unidentified users are never allowed for write permissions. Unfortunately, because of misconfiguration on FTP server, even such malicious users or attackers have received the write permission and log in to the server with the identification of “unidentified”, then build hidden directories, finally upload tremendous, even unexpectedly illegal files “Warez” on the server.

Warezclient: The abbreviation of this attack is WC consists of downloading illegal software previously uploads during a warezmaster attack. The WC attack is the analytical inheritor of the previous WM attack. Once the document files have been uploaded intentionally by the intruder on the server, any unidentified/legal user can download these malicious/illegal files [27]. The WC attack can have devastating effects on the host machine which depends on the type of Warez that had been uploaded.

2.2 Detecting Measures

The most traditional and credible solution to warrant the safety of an organization's devices can be to abstain from the internet. However, that is not a very feasible and efficient idea, especially on the contemporary age. Nowadays, computers are nothing without connecting to a network. There is a general controversy that how can we keep our network-connected computers safely from external frauds? Detection and prevention of all kind of malware should be taken into consideration as fast as possible in an efficient way. As we all know, as far as anti-virus and anti-malware software is considered fairly effective. But when we install A/V software, it should be kept updated and also needs us to take the precautions. It couldn't prevent the security system from getting hacked or intruded like a firewall. Other cost-effective countermeasures designed to detect, prevent or block fraudulent malicious activities all over the network could be intrusion detection and prevention systems. After identifying abnormal traffics, IDS or IPS would write to log files when suspicious activity is detected, then would send event notifications taking preventative measures. However, some kind of destructive drawbacks, like misclassification of genuine traffics as anomalies, and incompetence to configure unknown attacks, make intrusion detection and prevention systems run inefficiently.

When it comes to external frauds or network attacks against the computer security, the vast majority of related news on the Internet come to exist every day. Now, the malicious codes are commonly recognized as various kind of network attacks, like DoS, Web attack, and Warezmaster discussed in the previous section. Budding attacks either for getting access to the local server, aiming of demonstrating a great damage to websites or deleting the vital piece of information, such as bank accounts or credit card numbers, are getting more pervasive through almost every corner of security systems. In conclusion, all the evidence justifies a remarkable consequence that indentifying and distinguishing the external frauds or network malicious attacks are one of the principal requirements that information security researchers do need. Detecting tools are capricious according to capricious conditions. Let us review some principle different types of detection methodologies.

2.2.1 Firewalls

However, in addition to that staggering cost of fraudulent incidents mentioned in the previous chapter, detection, and prevention of all kind of malware should be taken into consideration as fast as possible in efficient ways. Firewalls essentially allow or deny traffic into a host or an entire network. There are mainly two types of firewall. The host-based firewalls which controls traffic coming into a particular host, like a server, and also control the traffic leaving the host. While network-based firewall controls traffic coming into and leaving the network based on the rules named ACL [28].

The firewall has long been used as the most essential device in dealing with safeguarding network security. Nevertheless, firewalls have become largely incapable in its utility in monitoring activity on the internal network and it increasingly recognizes that the necessity to monitor internal networks driving from the fact that the majority cases of all attacks and losses incorporate insiders.

2.2.2 A/V tools

Anti-Virus system can either be a software or hardware device which supervisorily control the network of a system for prospective fraudulent activities [29]. Virus scanning and virus prevention techniques that A/V software wield are primarily utilized to detect or distinguish viruses from incriminating precious network accumulations.

Virus scanning: Pattern-matching algorithms, that can scrutinize innumerable various signatures simultaneously, preferred by Virus scanners. Furthermore, proposed algorithms have capabilities of scrutinizing either known or unknown worms or Trojan horses. In particular, these virus scanners have the competence to be able to keep hard disks clean by removing viruses. Moreover, this software can also possess auto-update functions that can cope with downloading signatures of new malwares or viruses into the virus-scanning database.

Virus prevention: Virus prevention software conventionally domiciles in computer memory and supervisory control the system's network demonstration, moreover also monitor filters incoming runnable programs and individual file types. Once malicious or fraudulent virus obtains a boot sector or a program, then the whole system will be stopped and the users and operators are instigated to eliminate that special sort of malicious code.

Snort: Snort belongs to one variety of A/V system which is established for IP based networks. Snort can be best dealt with by analysis of network traffic and configuring viruses, and prevent other inherent dexterous transactions. Functions applied for Snort is classified into three distinct ways: (1) sniffer mode: discovering the Internet packets and dispose on a console board; (2) logger mode: logging and rescuing the packets to the disk; (3) intrusion detection mode: analyzing and evaluating the network traffic against elucidated rule sets. Snort regulations can also be assumed by operators and checks heterogeneous features of packets whether the network traffic should be qualified and permitted or blocked directly.

Bro: Bro is a submissive network A/V system which supervises disbelieving traffic link in depth, recording the network transaction activities along with the requested URI, headers, DNS request responses, SSL certificates, SMTP sessions, and so on. Bro plays a crucial role in network traffic analysis and is intimately customizable, and newborn analysis functions can effortlessly be inserted by the means of scripts. Bro comes with a preidentified measured library and subsidizes a multiple amounts of attributes for detecting as well as preventing intrusions.

From what has been discussed above, we could safely summarize that anti-virus and anti-malware software are contemplated as fairly reliable and effective. But when we install A/V software, it should be kept updated and also needs us to take the precautions. In the contemporaneous Internet-connected circumstance, more proactive methods are required to audit or monitor networks and systems. A/V software couldn’t prevent the security system from getting hacked or intruded like an effective intrusion prevention and detection systems.

2.3 IDS/ IPS

One of the major challenges in this approach has been the number of false positives, there tend to be too many cases for a human operator to review, and a significant number of them turn out to be normal transactions anyway. Therefore, improving the accuracy of fraud detection is a key to success in this case. To prevent that kind of typical problems, we will determine to use network-based and anomaly-based IDSs for the methodology of our research.

Defense in depth, one of the key essence of information and network security, generally applies in the case of network security. The principle of defense in depth declares that organizations should demonstrate multifarious and overlapping security controls to achieve the identical control objective. This kind of layered approach detects opposed to the failure of any isolated security control. If one single control fails, there is still another isolated control established to achieve the identical security objective standing in it's place.

In order to understand what the protocols do, a direct comprehension of the OSI model and the encapsulation procedure is essential to do efficacious packet investigations. The seven layers OSI model systematize the functions of data transformation by seperating it into isolated layers, as presented in Figure 2.3. We will review the main function of each layer by describing about some common protocols in that layer and mention the protocol data unit, which defines the shape of the data positioning in that layer, and we will also conclude about any addresses that are needed, such as MAC address or IP address.

1) Application layer: Application layer is positioned in layer seven initiating contact with a network, generally a user who would initiate perhaps getting a web page. The protocols that are used are HTTP, FTP or SMTP. The protocol data unit at this layer is simply data, and there's no addresses that are needed.

2) Presentation layer: Presentation layer will format the data, providing selective encryption and compression, and the protocol data unit at this point is simply data.

3) Session layer: This is all about instigating, maintaining and tearing down a session. The protocol data unit is also data.

4) Transport layer: The transport layer is responsible for transporting data, initiating the encapsulation process. Based on how we require it transported, we might select a connectionless protocol such as UDP or connection-oriented protocol such as TCP. The protocol data unit at this point is a segment, requiring a port address. The source and destination port addresses are also required, which will logically related with a suitable application.

5) Network layer: The third layer designated as network layer, supplying routing, addressing, and different protocols existed in this layer. As we are all concerned, there is a need for IP, ICMP, and Address Resolution Protocol which is placed between layer two and layer three. There's no routing involved, it's simply resolving an IP address to a MAC address. The protocol data unit at this point is a packet and the address is an IP address.

6) Data link layer: The protocol is Ethernet two, which is the most extensively used protocol on a local area network, and the protocol data unit at this point is a frame, the address is a MAC address. It's encapsulated with proper frame formation obtaining all the prerequisite addresses.

Once we have already not been familiar with the OSI model, got a better apprehension of each isolated layers of the OSI model, the protocol data units and the addressing.

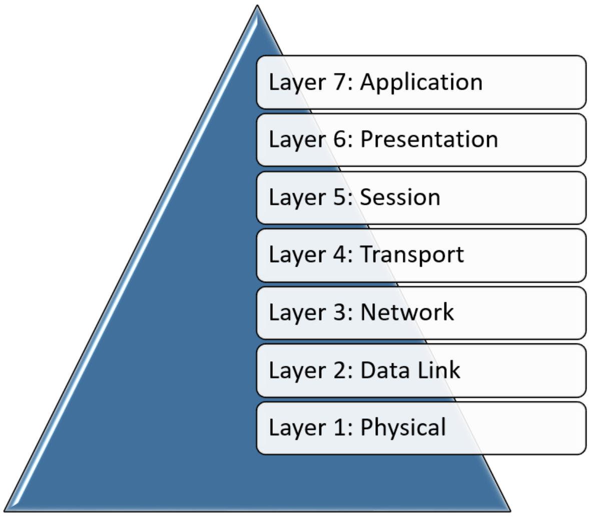


Figure 2.3 : Seven layers of the OSI model.

In conclusion, these understandings will lend a help to investigate traffic better. When architecting a secure network, we should definitely follow proposed defense in depth and OSI model principles. Let's take a look at how we can apply the defense in depth layered security approach to our external fraud detection environment.

Network-based IDSs are located at a pivotal point inside the network to discriminating passing network traffics for predicting and saving the record of fraudulent malicious activities to deposit, as depicted in Figure 2.4. The IDS and IPS sensors can be placed in front of the firewall or behind the firewall in the network of an organization. Routers on a DMZ network shared segment may filter traffic before it even reaches the firewall. Similarly, an intrusion prevention and detection systems might sit in front of or behind the firewall, filtering out potentially malicious traffic that manages to pass through the firewall before it reaches the eternal network. However, they could not be placed in the firewall, since the monitoring process of firewall is faster than the other detection systems, while IDS and IPS engines taking a lot of time by generating and testing the network intrusion detection classifiers. Defense in depth is a time-tested security principle and it certainly applies to network security. It is actually the process not a product, and is a proactive approach to thinking about security from the inside out. The major supremacy of anomaly-based IDSs is the capability to distinguish undefined malicious traffics. We will work on this two types of IDSs more in detail with some real experimental studies in the next chapter.

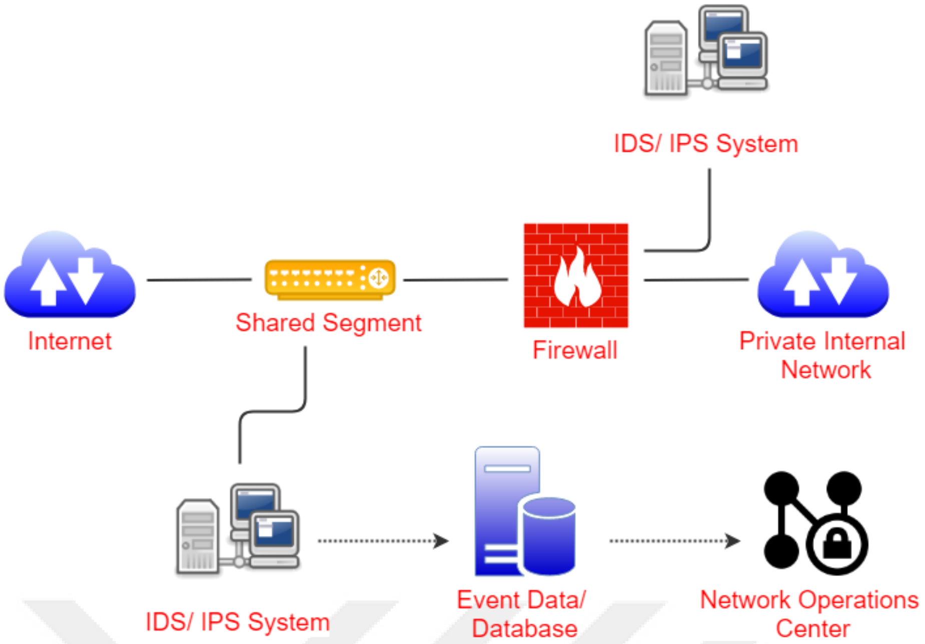


Figure 2.4 : Placement of IDS/ IPS sensors in the network of an organization.

Intrusion detection or prevention systems predominantly focus on distinguishing or prohibiting malicious or suspicious network traffics and irregular transactions originating from both inside or outside of the corporation. Intrusion detection systems are generally split into two distinct problems, like host-based and network-based IDS. Both include rarely complicated datasets having characteristics that usher themselves to solve statistical problems.

Host-based intrusion detection system: Host-based IDS supervisorily monitors either inbound or outbound traffics that are running from individual devices or hosts on the network. After finished all mentioned processes, it accumulates the whole traffic data on a single host. The advantage of host-based agents is that they can supervise every tiny change to critical system files and changes in user privileges [29]. Although host-based IDS has a distinct advantage of requiring no auxiliary hardware, for the reason that they keep running on the system itself only. If we just want to analyze a single system, the total cost of host-based IDSs is regularly underneath than those for their network-based correlatives. The disadvantages to the host-based approach are that to analyze the entire network, it is compulsory to load the IDS to every computer. For the main reason that host-based IDSs do not monitor packet headers, so they, unfortunately, cannot detect denial-of-service attacks.

Network-based intrusion detection system: We must recognize the undeniable fact that network-based IDS detects or distinguishes network traffics to and from all apparatus, positioning at pivotal points within the network. It routinely supplies effective, contemporary information not having the benefit of predicting host or network resources. Since network-based IDSs can also supply superior controls on event logs monitoring packet headers, hence they can detect and prevent network DoS attacks. Moreover, because this kind of IDS is mainly supervising intrusions in real time, it can retaliate to an ongoing attack to limit destruction. The vital disadvantage of network-based IDSs is the evidence that they become less powerful as network traffic proliferates, working in high efficiently and perfectly on a vacant or unoccupied network. This is a major deficiency while scrutinizing contemporary tremendous amount of transaction volumes, as increasing of switched Ethernet and fast Ethernet.

There is another rudimentary proposition to network intrusion detection.

Signature-based detection system: It is noteworthy to pay attention to the new concept of miuse-based or knowledge-based detection system which is the alias of the signature-based detection system. Most existing systems rely on signatures of attacks. There is a robust superiority of this method is that signatures are uncomplicated to prosper and discern when we acquire the knowledge includes the properties of our network behavior [30]. They are more productive based on the identified known intrusion by doing updates of the signatures constantly, receiving low false alarm rates. This contribution is the reason for the actuality that they usually distinguish extremely particular patterns, strings, and signatures. As opposed to widely accepted advantages, there still exists some astonishing drawbacks that knowledge-based IDSs are inefficacious against new techniques having no pattern in the base of knowledge. So it is a necessary prerequisite to keep IDS up-to-date with new patterns of environments and vulnerabilities. It would be exceedingly taking too much time for analyzing each newborn vulnerability to update the signatures of IDS. So creating a new signature for every attack would be time-consuming on solving big amount of network traffic data, resulting in the IDS not to find out the novel or unexisting network attacks. The proposed model of anomaly and misuse intrusion detection system is given in the Figure 2.5 below.

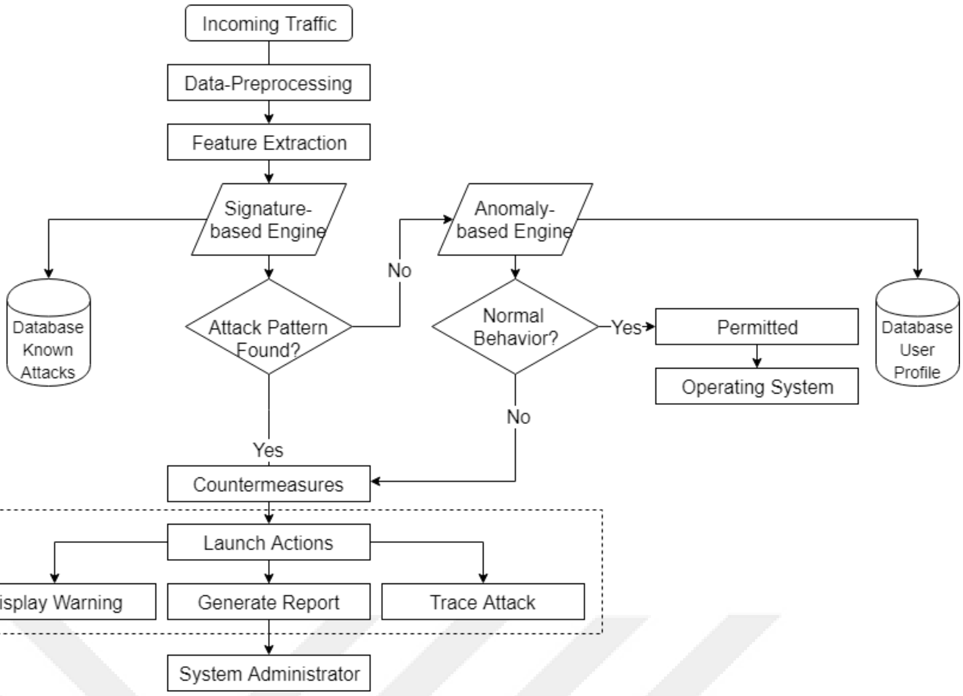


Figure 2.5 : Proposed model of anomaly and misuse signature detection system.

Statistical anomaly-based detection: When it comes to statistical anomaly-based detection, the vast majority of researchers would deduce the fact that it supervises network traffic comparing in opposition to an pre-installed baseline. Different from misuse detection, anomaly detection would be able to find out the existing concealed risks, devoting to the establishment of genuine traffic profiles for the system. The superiority of anomaly-based IDS outweigh any benefit we conclude from signature-based IDS that could only detect fraudulent traffics for which a signature has heretofore been identified. The AIDS can detect any novel intrusive or malicious activities falling out genuine traffic patterns for which a signature does not create, according to the presumption that all fraudulent traffics are automatically anomalous. The main process begins with acquiring first-step knowledge on what the geniuine features for the perceived objects are, after that should determine and classify what category of traffics should be labelled as anomalous or genuine. For instance, the malicious transfer of financial funds, like credit card transactions, from one account to another could go off alarms if TL amount was remarkably aberrant what was normal for the innocent, or if one individual did not ordinarily access proposed account he or she wants, or if the act of transaction was processed at an very unexpected time.

Firewall, IDS and IPS are not the same creatures or engines. As narrated, defense in depth is the efficacious measure we are looking to allocate. A Firewall is a control mechanism applied to obstruct and restrict the protocols traversing between two networks at layers 3 named network layer, and layer 4 named transport layer. In some cases, the firewall can perform restricted inspection of layers 5, 6 and 7, named session layer, presentation layer, and application layer. But in those instances, it's straightforwardly attempting to do complementary commission, which it may not perform well because firewalls don't have the processing muscle to do protocol analysis. Eventually, we can think of a firewall as a tool to control protocols. An IDS is not a control mechanism. It has much more processing power than a typical firewall, but generally less throughput since it is performing much more work by taking much more time. An IDS can detect intrusions but it cannot control them. It cannot function as a firewall and it cannot function as an IPS. An IDS can perform detection in layers 2 through 7 in proposed OSI model. An IPS is a control mechanism, while an IDS is being with the ability to control frames and packets in the same layers 2 through 7. Moreover, they have much more processing power than a firewall, but generally less throughput due to the inspection it must perform. An IPS may have the ability to perform many firewall-like functions, while the IPS is generally more difficult to administer when deployed for that function since it's designed to detect exploits and prevent attacks, not act like a firewall.

3. MACHINE LEARNING TECHNIQUES FOR EXTERNAL FRAUD DETECTION SYSTEM

Machine learning is a subfield of artificial intelligence where humans teach machines how to solve real-time problems without explicitly programming them to do so. In general, there are three types of machine learning based on the application of labeled data – a) Supervised learning, b) Unsupervised learning, and c) Reinforcement learning. Since our aim is to classify the unseen traffic data to normal or fraud, is kind of a classification problem that obtains categorical outputs. The commonly used classification algorithms in supervised learning include the k-nearest neighbor, Naive Bayes, Decision tree, and Random Forests

3.1 K-Nearest Neighbor Classifier

K-nearest neighbor is the most wide-ranging applied nonparametric classification method. In K-nearest neighbor algorithm, the appropriate K value is a principal attribute that affects the detection performance and accuracy, while an inappropriate value of K directly results in high detection error rate [31].

One of the most frequently applied distance metrics is Euclidean distance. Let us assume that there are two factor matrices, X = [x1, x2, x3, …, xn] and Y = [y1, y2, y3, …, yn], then Euclidean distance can be defined as (3.1):

$$d(X,Y) = \sqrt{(\mathrm{x1} - \mathrm{y1})^2 + (x2 - y2)^2 + \cdots + (xi - yi)^2} \tag{3.1}$$

In order to explain more explicitly, let us take a small experimental example for a dataset, having a complex structure. To generate a KNN fraud detection classifier, we need to group reasonable clusters using Euclidean distance. After processing ten times of iterations, as is illustrated in Figure 3.1, three types of traffic cluster has been generated.

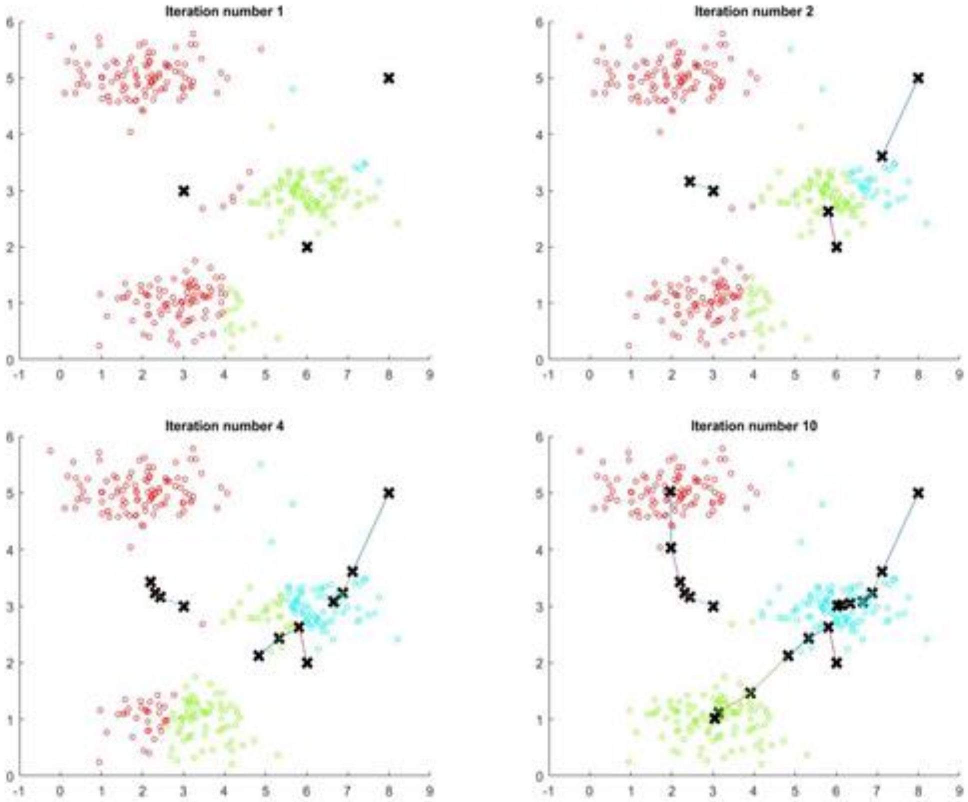


Figure 3.1 : The schematic diagram of the KNN fraud detection algorithm.

3.2 Naive Bayes Classifier

If input values $\boldsymbol{x}$ are independent with each other, Naïve Bayes' classifier assumes independent inputs, ignoring correlations and possible dependencies between them. By this way, Naïve Bayes classifier reduces a multivariate problem, as (3.2), to a bunch of univariate problems, as formula (3.3), making all calculations easier. $p(C|x)$ means a conditional probability, where given input values $\boldsymbol{x}$ to predict the class variable C.

$$p(C|x) = \frac{P(C)p(x|C)}{P(x)} \tag{3.2}$$

$$p(x|C) = \prod_{j=1}^{d} p(\mathrm{x_j}|C) \tag{3.3}$$

3.3 Decision Tree Classifier

One of the preeminent examples of a hierarchical model for supervised learning can be easily considered as decision tree classifier. A decision tree is composed of internal decision nodes and terminal leaves, as illustrated in Figure 3.2. Rounded rectangle nodes represent decision nodes and rectangles are leaf nodes, also named as classified nodes.

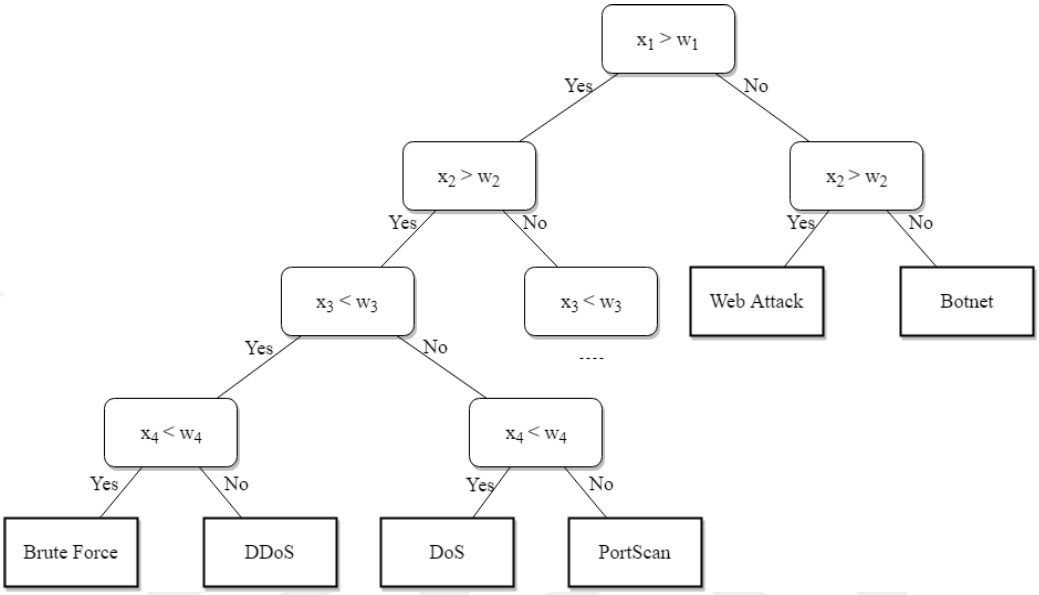


Figure 3.2 : A graphical model of a decision tree classifier.

In the example for the classification problem of decision trees, each decision node m displays a test function, the performance of a branch is evaluated by the measure of impurity [32]. When we assume for a node m, N_m is the number of training transactions reaching node m. The estimation for each probability of class C_i is calculated as (3.4) below:

$$P(C_i|x, m) = \frac{N_i}{N_m} \tag{3.4}$$

where N_i of N_m belongs to class C_i, m is the amount of total training instances.

Node m is pure not splitting any further if the estimated probability $P(C_i|x, m)$ is either 1 or 0. If the value equals to 1, all such instances belong to class Ci, if it equals 0, none of the instances belong to Ci. One kind of measure to calculate impurity of a node is named entropy, as formula (3.5). If the split is pure, taking 1 or 0, we are not required to split any more but can add a leaf node labeled with the class of each group of instances, as mentioned in the previous figure.

$$\text{Entropy} = \sum_{i=1}^{k} \mathrm{P}(\mathrm{C}_i|\mathrm{x}, \mathrm{m})\log \mathrm{P}(\mathrm{C}_i|\mathrm{x}, \mathrm{m}) \tag{3.5}$$

3.4 Random Forest Classifier

RF algorithm is a high-level branch of decision tree algorithms. Using multiple trees could reduce the risk of overfitting while training data, obtaining less training time and high accuracy in generating a classification model. Since it does not last long in training, hence random forest algorithm runs efficiently on a large database.

3.5 K-Fold Cross-Validation Classifier

K-fold cross-validation, the aboriginal dataset is capriciously seperated into k equal-sized subsets. Among of proposed partitioned subsets, a single subset is utilized as the validation data for testing the future model, and the other k-1 subsets are retained as training data. With the similar functions, the total operation is then repeated k times, with each of the k subsets used only once as the validation data. Considering all preocess above, the k results can then be averaged to generate a single approximation like (3.6):

$$\mathrm{E} = 1/10 \sum_{i=1}^{10} Ei \tag{3.6}$$

Each observation is used for validation precisely once. We use the most common category of cross-validation named 10-fold cross-validation, as depicted in Figure 3.3 below.

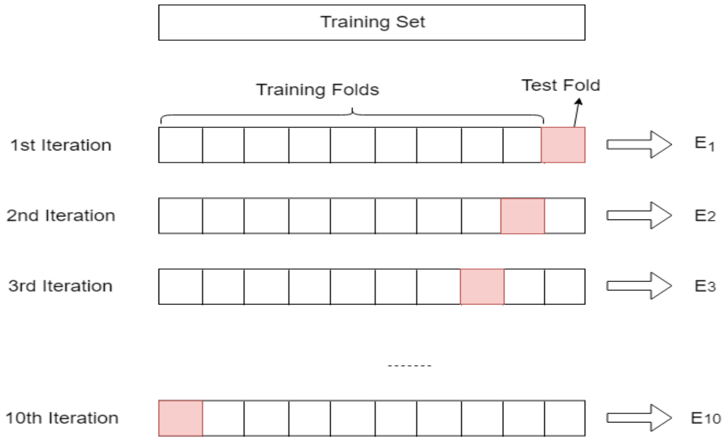


Figure 3.3 : An example of a 10-fold cross validation.

3.6 Evaluation Metric

Using some kind of different evaluation metrics, like accuracy, precision, and f1-score, we can evaluate the performance result of our classification models and deduce which model performs the best [32]. We can learn how well our model performs by using the mentioned evaluation metrics.

In a confusion matrix, the accuracy is calculated in a tabular form as depicted in Table 3.1 below. Where true positives and negatives represent the correct operation of the detector, false positive, as well as negatives are the events that undermine the detection performance when IDS is not verified. Hence, the values on left diagonal of confusion matrix should be as infinitesimal as possible approximating to zero.

Table 3.1 : Confusion matrix in a tabular form.

Confusion Matrix	Actual Value (as confirmed by experiment)	
Predicted Class (predicted by the test)	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Accuracy is the measure of how many data points or observations are predicted correctly out of all number of instances. Accuracy, a formula like (3.7), works best if receiving a similar value of false positives and false negatives result.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad (3.7)$$

F1 score (3.8) is more functional than accuracy principally in the case where we have an unequal amount of class distribution like in our case. It's the weighted average of Precision and Recall. For that reason, this score will take both false negative and false positives into consideration. If their values are exceptionally dissimilar, it is more preferable to calculate both Precision and Recall or F1 score.

$$F1\ score = 2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \qquad (3.8)$$

where Precision (3.9) is calculated by:

$$Precision = \frac{TP}{TP + FP} \tag{3.9}$$

where Recall (3.10) is calculated by:

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

4. EXPERIMENTAL WORK

We have selected Anaconda Jupyter notebook to perform the experiments as it provides enough machine learning and deep learning libraries to visualize and analyze the network traffic data. Implementation environment of our experiments, like hardware and software that we used, are listed below:

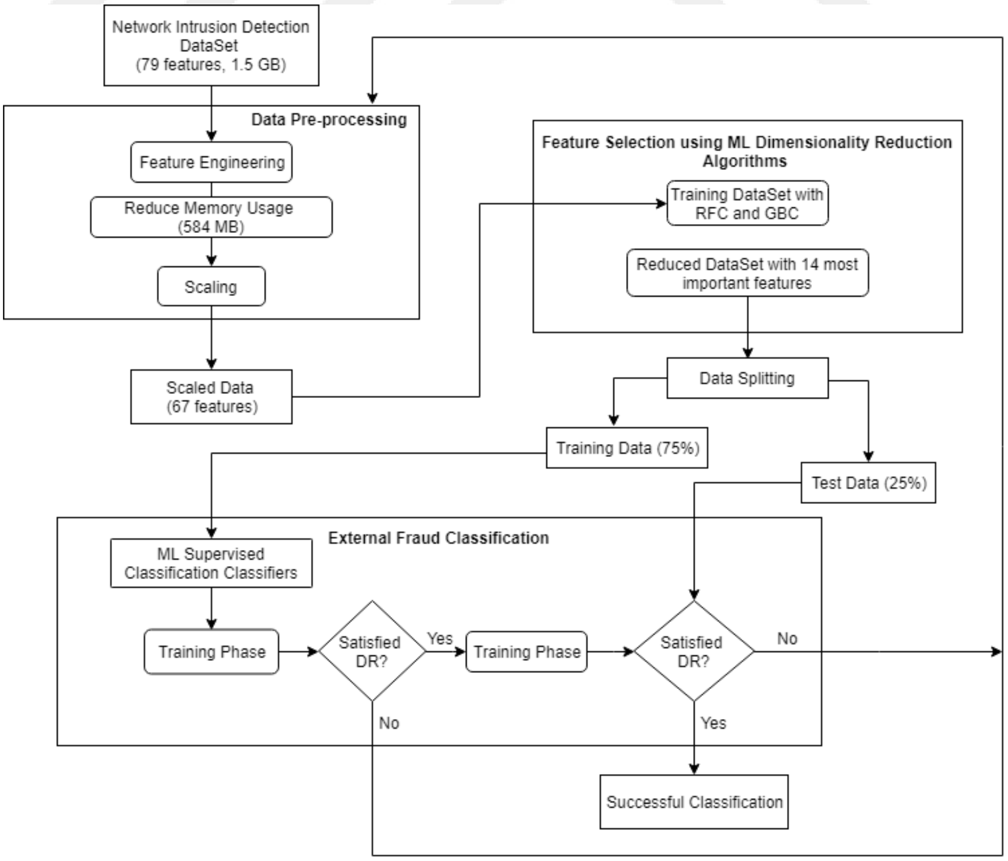
CPU: Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz

RAM: 8GB, OS: Windows 10

Programming Language: Python 3.6.6

Libraries used: numpy: 1.15.2, scikit-learn: 0.20.0, pandas: 0.23.4, matplotlib: 3.0.1, seaborn: 0.8.1, and Tensorflow.

The flow diagram of the machine learning model is presented below, as Figure 4.1.



Network Intrusion Detection DataSet (79 features, 1.5 GB)
Data Pre-processing
Feature Engineering
Reduce Memory Usage (584 MB)
Scaling
Scaled Data (67 features)
Feature Selection using ML Dimensionality Reduction Algorithms
Training DataSet with RFC and GBC
Reduced DataSet with 14 most important features
Data Splitting
Training Data (75%)
Test Data (25%)
External Fraud Classification
ML Supervised Classification Classifiers
Training Phase
Satisfied DR?
Yes
No
Training Phase
Satisfied DR?
No
Yes
Successful Classification

Figure 4.1 : The data flow of our external fraud detection model.

4.1 Data Acquisition

Some existing number of datasets, like DARPA 98 (Lincoln Laboratory 1998-99), KDD’99 (University of California, 1998-99), DEFCON (The Shmoo Group, 2000-2002), PaySim [33], and NSL-KDD [34], have been analyzed for estimating and optimizing the performance of the proposed intrusion detection or intrusion prevention systems by security researchers. Depending on related studies, numerous such kind of datasets are outdated and undependable to do an efficient experimental study. There are several reasons for this strongly demanding of more reliable uptodate datasets, the foremost one can be that proposed libraries or resources are suffering from inadequacy of traffic variaty and capacities.

This research initiates with a trustworthy dataset that includes genuine and most frequent fraudulent or intrusive network traffics approaching real-world benchmarks and criterias. Canadian Institute for Cybersecurity generates a dependable and contemporary dataset (CICIDS2017) that contains benign and generally most frequent network attack traffics [35]. The dataset includes all type of up-to-date network attacks mentioned in chapter 2. The major technical challenge that this dataset poses to predicting and classifying external frauds is the highly imbalanced distribution between genuine and fraudulent classes in over 2 million observations of data [36]. The objective of our experimental work is to solve both these issues of imbalanced distributional skew applying a detailed data exploration and engineering by choosing a suitable machine-learning algorithm..

4.2 Data Preprocessing/ Feature Engineering

In data engineering step, we are trying to estimate the performance of a comprehensive set of network traffic attributes and use some machine learning techniques to figure out the best performanced and efficient of features for detecting the definite external fraud categories by removing redundant features. To get the best result when training a machine-learning algorithm, we want to make the problem as simple as possible for the algorithm to generate the classification model since including irrelevant features can harm the accuracy of the model. Feature engineering also has the crucial ability to convert nonsensical data into meaningful information. Following steps is the process that we apply some feature engineering to the original raw data. All the feature names and corresponding data types are presented in Table 4.1 below.

Table 4.1 : Feature name and their types before functioning.

Feature Name	Type	Feature Name	Type
Destination Port	int64	Max Packet Length	int64
Flow Duration	int64	Packet Length Mean	float64
Total Fwd Packets	int64	Packet Length Std	float64
Total Backward Packets	int64	Packet Length Variance	float64
Total Length of Fwd Packets	int64	FIN Flag Count	int64
Total Length of Bwd Packets	int64	SYN Flag Count	int64
Fwd Packet Length Max	int64	RST Flag Count	int64
Fwd Packet Length Min	int64	PSH Flag Count	int64
Fwd Packet Length Mean	float64	ACK Flag Count	int64
Fwd Packet Length Std	float64	URG Flag Count	int64
Bwd Packet Length Max	int64	CWE Flag Count	int64
Bwd Packet Length Min	int64	ECE Flag Count	int64
Bwd Packet Length Mean	float64	Down/Up Ratio	int64
Bwd Packet Length Std	float64	Average Packet Size	float64
Flow Bytes/s	object	Awg Fwd Segment Size	float64
Flow Packets/s	object	Awg Bwd Segment Size	float64
Flow IAT Mean	float64	Fwd Header Length.1	int64
Flow IAT Std	float64	Fwd Avg Bytes/Bulk	int64
Flow IAT Max	int64	Fwd Avg Packets/Bulk	int64
Flow IAT Min	int64	Fwd Avg Bulk Rate	int64
Fwd IAT Total	int64	Bwd Avg Bytes/Bulk	int64
Fwd IAT Mean	float64	Bwd Avg Packets/Bulk	int64
Fwd IAT Std	float64	Bwd Avg Bulk Rate	int64
Fwd IAT Max	int64	Subflow Fwd Packets	int64
Fwd IAT Min	int64	Subflow Fwd Bytes	int64
Bwd IAT Total	int64	Subflow Bwd Packets	int64
Bwd IAT Mean	float64	Subflow Bwd Bytes	int64
Bwd IAT Std	float64	Init_Win_bytes_forward	int64
Bwd IAT Max	int64	Init_Win_bytes_backward	int64
Bwd IAT Min	int64	act_data_pkt_fwd	int64
Fwd PSH Flags	int64	min_seg_size_forward	int64
Bwd PSH Flags	int64	Active Mean	float64
Fwd URG Flags	int64	Active Std	float64
Bwd URG Flags	int64	Active Max	int64
Fwd Header Length	int64	Active Min	int64
Bwd Header Length	int64	Idle Mean	float64
Fwd Packets/s	float64	Idle Std	float64
Bwd Packets/s	float64	Idle Max	int64
Min Packet Length	int64	Idle Min	int64
Label	object		

The original network traffic dataset is separated into five days from Monday to Friday. In other words, the capturing period of network traffic started at 09:00 on Monday and ended at 17:00 on Friday, subsequently ran for 5 days. Network attacks were continuously executed throughout the total duration, as described in Table 4.2, Monday is the normal day without obtaining any fraudulent activities, consisting of the normal traffic, while the other days always incorporate the most general types of network attacks, described detailedly in chapter 2.

Table 4.2 : Label of original separated dataset.

Days	Labels
Monday	Benign
Tuesday	FTP-Patator, SSH-Patator
Wednesday	DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris, Heartbleed
Thursday	Web Attack Brute Force, Web Attack SQL Injection, Web Attack XSS, infiltration
Friday	Bot, DDoS, PortScan

In order to generate holonomic and general classification model, we need to merge all that separated datasets. After concatenating process, our dataset consists of more than two and half million observations and 79 attributes or features to help us predict the type of network intrusions which is our target or output variable. In total, there are 16 categorical, 60 numerical and 3 objective variables, which make a total of 79 variables (/features/attributes), and there are no any missing values in our data frame.

Table 4.3 : The number of instances for each type of attack.

Attack Type	Number of Instances
FTP-Patator	7938
SSH-Patator	5897
DoS GoldenEye	10293
DoS Hulk	231073
DoS Slowhttptest	5499
DoS Slowloris	5796
Heartbleed	11
Web Attack Brute Force	1507
Web Attack SQL Injection	21
Web Attack XSS	652
Bot	1966
DDoS	128027
PortScan	158930
Benign/ Normal	1984531

The target variable named ‘Label’ has 12 different values, making this a multiclass classification problem, Table 4.3 above shows the number of observations each attack type includes. For classifying the network traffics clearly, we tried to group the target variable *Label* into two integer categorization with which applied our proposed classification algorithms separately. Firstly, we make changes to the target variable *Label* ranges between integer value zero to seven and each number is a key representing names of different attack type, as shown in Table 4.4.

Table 4.4 : Specific categorization of all traffic types.

Label	Category	Traffic Type
0	Normal	BENIGN
1	BruteForce FTP/SSH	FTP-Patator, SSH-Patator
2	DoS	Hulk, GoldenEye, Slowloris, Slowhttptest
3	Heartbleed	Heartbleed
4	Web Attack	Brute Force, SQL Injection, XSS, Infiltration
5	Botnet	Bot
6	DDoS	DDoS
7	PortScan	PortScan

Later, we categorized the string target variable in only two groups of labeling by normal traffics as zero and all fraudulent traffics as one decently, listed below in Tabel 4.5.

Table 4.5 : Decent categorization of all attack types.

Label	Category	Traffic Type
0	Normal	BENIGN
1	Fraud	BruteForce FTP/SSH, DoS, Heartbleed, Web Attack, Botnet, DDoS, PortScan

Figure 4.2 and Figure 4.3 illustrates the distribution of all type of intrusions after categorized in different types as mentioned above, 557610 anomaly transactions totally, it is an uneven and skewed data, since anomaly traffics only account for 0.219% out of all transactions.

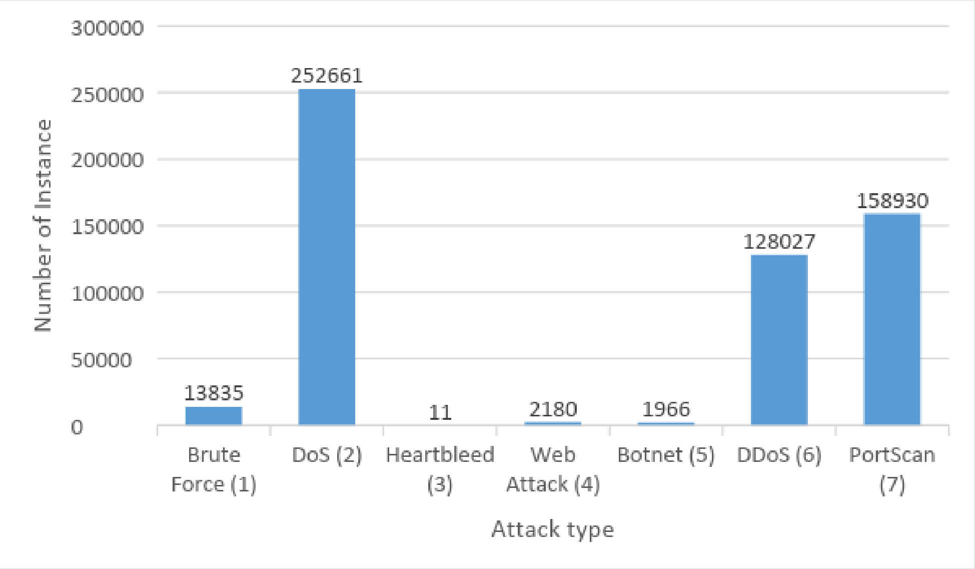


Figure 4.2 : Distribution of existed all network attacks.

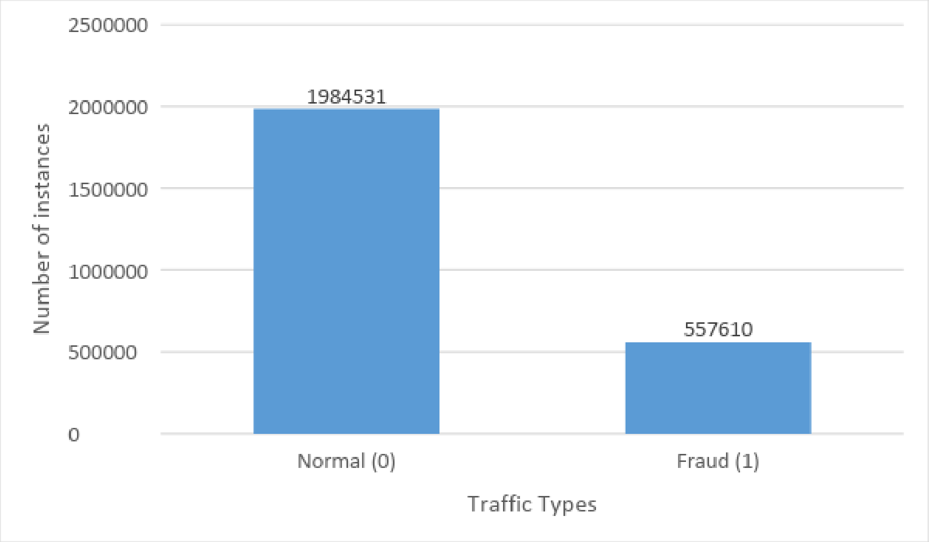


Figure 4.3 : Distribution of all traffics according to their categorization.

Training such a big amount of data with two and a half million (2,542,141) of observations would take a lot of time and would over-fit the data. Shrinking the data may give us a preferable performance and avoid overfitting. What we would like to do is remove all redundant features manually that contain the same value all the time, let us display a short list of the redundant attributes having no any special means, shown in Table 4.6. Super add, there are two more attributes we also need to drop manually from our merged dataset, type of NAN, meaning not a number or infinite number, that hard to deal with in training data.

Table 4.6 : Redundant attributes that have to be removed.

Feature	Type	Value
Bwd PSH Flags	int64	0
Fwd URG Flags	int64	0
Bwd URG Flags	int64	0
CWE Flag Count	int64	0
Fwd Avg Bytes/Bulk	int64	0
Fwd Avg Packets/Bulk	int64	0
Fwd Avg Bulk Rate	int64	0
Bwd Avg Bytes/Bulk	int64	0
Bwd Avg Packets/Bulk	int64	0
Bwd Avg Bulk Rate	int64	0
Flow Bytes/s	object	NAN
Flow Packets/s	object	NAN

After deleting 12 irrelevant features out of total dataset, it remains 66 features adding to one object feature. According to the weight of feature importance, we can also drop other redundant features automatically later by using dimensionality reduction of machine learning algorithms.

Since we consider the properties, the size of the dataset is big for a lower/mid-range laptop so we made a script to make the dataset smaller without losing information. In that way, we can reduce the accounting memory of the dataset by selecting smaller data types and applying them by fitting the range of corresponding values. It can be easily discovered that after reducing dataset memory size by approximately 62.3 %, memory usage was reduced from 1.5+ GB to 584+ MB, not changing the total number 67 of features, as illustrated obviously in Table 4.7 below. All the feature names and corresponding data types after applying the reduction of memory usage algorithm to a reduced dataset are presented in Table 4.8.

Table 4.7 : Before and after applying the reduction of memory usage algorithm to a reduced dataset.

Before	After
1.5+ GB	584.3+ MB
float64(22), int64(44), object(1)	float32(22), int32(8), int64(2), uint16(6), uint32(19), uint8(9), object(1)

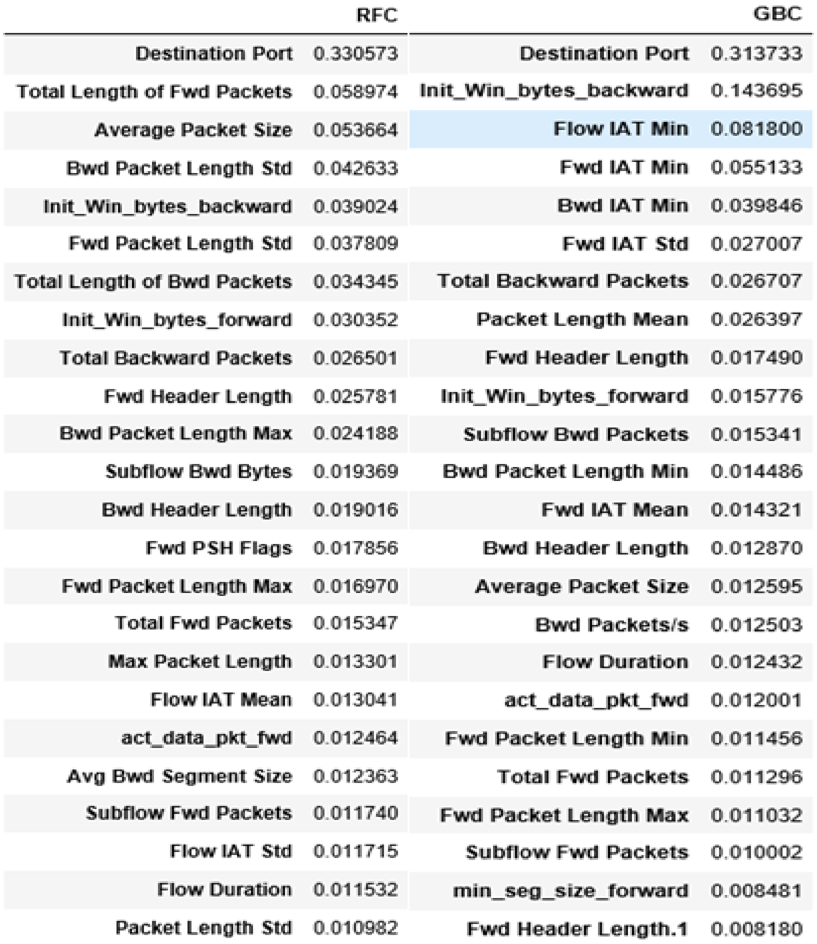
Table 4.8 : Feature name and their types after functioning.

Feature Name	Type	Feature Name	Type
Destination Port	unit32	Max Packet Length	unit16
Flow Duration	int32	Packet Length Mean	float32
Total Fwd Packets	unit32	Packet Length Std	float32
Total Backward Packets	unit32	Packet Length Variance	float32
Total Length of Fwd Packets	unit32	FIN Flag Count	unit8
Total Length of Bwd Packets	unit32	SYN Flag Count	unit8
Fwd Packet Length Max	unit16	RST Flag Count	unit8
Fwd Packet Length Min	unit16	PSH Flag Count	unit8
Fwd Packet Length Mean	float32	ACK Flag Count	unit8
Fwd Packet Length Std	float32	URG Flag Count	unit8
Bwd Packet Length Max	unit16	CWE Flag Count	unit8
Bwd Packet Length Min	unit16	ECE Flag Count	unit8
Bwd Packet Length Mean	float32	Down/Up Ratio	unit8
Bwd Packet Length Std	float32	Average Packet Size	float32
Flow Bytes/s	object	Awg Fwd Segment Size	float32
Flow Packets/s	object	Awg Bwd Segment Size	float32
Flow IAT Mean	float32	Fwd Header Length.1	int64
Flow IAT Std	float32	Fwd Avg Bytes/Bulk	unit8
Flow IAT Max	unit32	Fwd Avg Packets/Bulk	unit8
Flow IAT Min	unit32	Fwd Avg Bulk Rate	unit8
Fwd IAT Total	unit32	Bwd Avg Bytes/Bulk	unit8
Fwd IAT Mean	float32	Bwd Avg Packets/Bulk	unit8
Fwd IAT Std	float32	Bwd Avg Bulk Rate	unit8
Fwd IAT Max	unit32	Subflow Fwd Packets	unit32
Fwd IAT Min	int32	Subflow Fwd Bytes	unit32
Bwd IAT Total	unit32	Subflow Bwd Packets	unit32
Bwd IAT Mean	float32	Subflow Bwd Bytes	unit32
Bwd IAT Std	float32	Init_Win_bytes_forward	int32
Bwd IAT Max	unit32	Init_Win_bytes_backward	int32
Bwd IAT Min	unit32	act_data_pkt_fwd	unit32
Fwd PSH Flags	unit8	min_seg_size_forward	int32
Bwd PSH Flags	unit8	Active Mean	float32
Fwd URG Flags	unit8	Active Std	float32
Bwd URG Flags	unit8	Active Max	unit32
Fwd Header Length	int64	Active Min	unit32
Bwd Header Length	int32	Idle Mean	float32
Fwd Packets/s	float32	Idle Std	float32
Bwd Packets/s	float32	Idle Max	unit32
Min Packet Length	unit16	Idle Min	unit32
Label	object		

4.3 Dimensionality Reduction/ Feature Selection

After applying feature engineering to the original dataset described in the previous section, we still have copies of features. Redundant features will make the classification algorithm run very slowly later, have difficulty in learning and also tend to overfit in training set while doing worse in testing. Furthermore, having fewer features would decrease the training time. To reduce dimensions of our dataset automatically, we need to learn how each feature has an impact on predicting classes, and the most appropriate way to solve this is by using some feature selection algorithms related to dimensionality reduction methods of machine learning. Classifiers like Random Forest and Gradient Boosting provide a variable called *feature_importance_*, with which we can learn that which feature has more importance compared to others and by how much.

Table 4.9 : The comparison result of RFC and GBC getting top important 25 features.



	RFC		GBC
Destination Port	0.330573	Destination Port	0.313733
Total Length of Fwd Packets	0.058974	Init_Win_bytes_backward	0.143695
Average Packet Size	0.053664	Flow IAT Min	0.081800
Bwd Packet Length Std	0.042633	Fwd IAT Min	0.055133
Init_Win_bytes_backward	0.039024	Bwd IAT Min	0.039846
Fwd Packet Length Std	0.037809	Fwd IAT Std	0.027007
Total Length of Bwd Packets	0.034345	Total Backward Packets	0.026707
Init_Win_bytes_forward	0.030352	Packet Length Mean	0.026397
Total Backward Packets	0.026501	Fwd Header Length	0.017490
Fwd Header Length	0.025781	Init_Win_bytes_forward	0.015776
Bwd Packet Length Max	0.024188	Subflow Bwd Packets	0.015341
Subflow Bwd Bytes	0.019369	Bwd Packet Length Min	0.014486
Bwd Header Length	0.019016	Fwd IAT Mean	0.014321
Fwd PSH Flags	0.017856	Bwd Header Length	0.012870
Fwd Packet Length Max	0.016970	Average Packet Size	0.012595
Total Fwd Packets	0.015347	Bwd Packets/s	0.012503
Max Packet Length	0.013301	Flow Duration	0.012432
Flow IAT Mean	0.013041	act_data_pkt_fwd	0.012001
act_data_pkt_fwd	0.012464	Fwd Packet Length Min	0.011456
Avg Bwd Segment Size	0.012363	Total Fwd Packets	0.011296
Subflow Fwd Packets	0.011740	Fwd Packet Length Max	0.011032
Flow IAT Std	0.011715	Subflow Fwd Packets	0.010002
Flow Duration	0.011532	min_seg_size_forward	0.008481
Packet Length Std	0.010982	Fwd Header Length.1	0.008180

After running mentioned classifiers on our entire model, it gave us the ranking of feature importance according to their weight values. Table 4.9 above illustrates the comparison result of random forest and gradient boosting classifier top 25 important features with their weights in a descending direction.

After selecting the most important 14 features with the highest weight values generated by random forest and gradient boosting classifiers, as listed in Table 4.10, the memory usage was reduced from 584+ MB to 145.5 MB. If our score on the training data with selected efficient features is more than the original training data score, this means those features that we deleted, according to the weight value of features, do not quite give any additional information to predict and perform well.

Table 4.10 : The most important features after selecting.

Feature	Type
Destination Port	unit32
Init_Win_bytes_forward	int32
Init_Win_bytes_backward	int32
Flow IAT Min	int32
Fwd IAT Min	int32
Bwd IAT Min	unit32
Average Packet Size	float32
Bwd Packet Length Std	float32
Fwd Packet Length Std	float32
Packet Length Std	float32
Total Backward Packets	unit32
Total Length of Bwd Packets	unit32
Min_seg_size_forward	int32
Label	int64

We would apply random forest algorithm again for the selected features to learn their weight of importance rate, then obtain the graphical result as Figure 4.4.

Since all our selected features are numerical, we can find the correlation matrix of all data to learn how much each features are correlated with each other. As manifested clearly in Figure 4.5 below, features with high correlation are colored orange and purple, while features that have less or no correlation are colored black. *Total Backward Packets* depicts the highest positive correlation with *Total Length of Bwd Packets*, while *Average Packet Size* and *Bwd Packet Length Std* show the second highest positive correlation with *Packet Length Std*. Other features which have relatively high correlations are *Fwd IAT Min* and *Bwd IAT Min*, *Average Packet Size* and *Bwd Packet Length Std.*

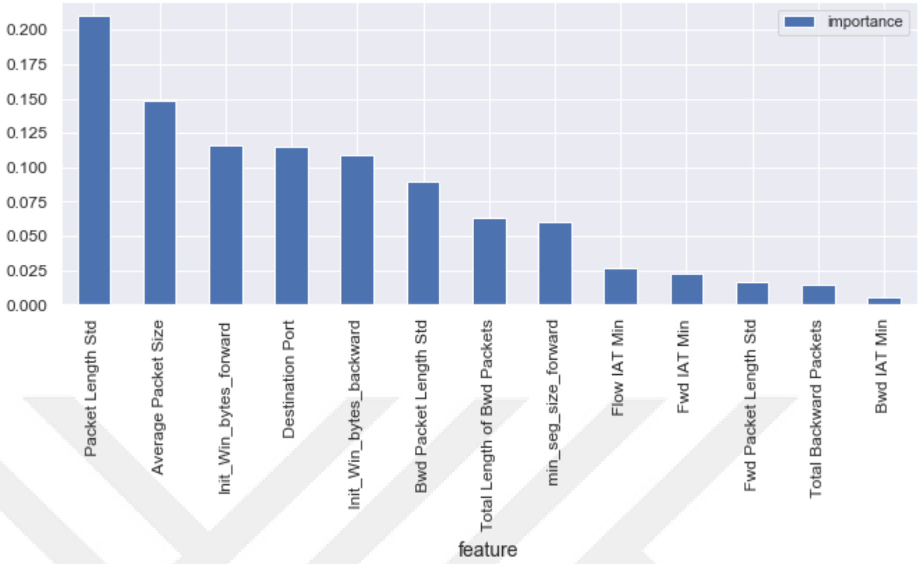


Figure 4.4 : The weight value of each selected important features.

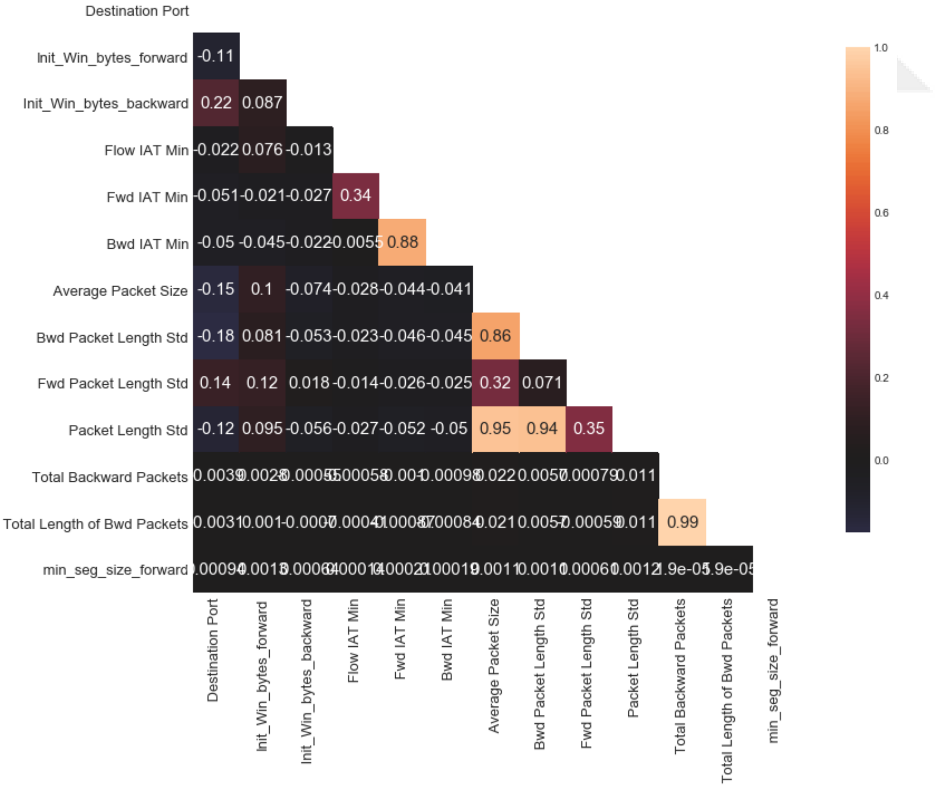


Figure 4.5 : Correlation matrix between selected features.

If we noticed the detailed description of the first four features in a generated dataset, as in Table 4.11, the difference between the max and min value of each feature are extremely big. So we will scale all features values to a specific range of 0 to 1, except for target variables.

Table 4.11 : Description of first four feature values.

	Destination Port	Init_Win_bytes_forward	Init_Win_bytes_backward	Flow IAT Min
Count	2.542141e+06	2.542141e+06	2.542141e+06	2.542141e+06
Mean	8.057795e+03	7.148799e+03	2.034795e+03	1.723543e+05
Std	1.836861e+04	1.447264e+04	8.520327e+03	3.039023e+06
Min	0.000000e+00	-1.00000e+00	-1.00000e+00	-1.400000e+01
25%	5.300000e+01	-1.00000e+00	-1.00000e+00	3.000000e+00
50%	8.000000e+01	2.510000e+02	-1.00000e+00	4.000000e+00
75%	4.430000e+02	8.192000e+03	2.350000e+02	6.600000e+01
Max	6.553500e+04	6.553500e+04	6.553500e+04	1.200000e+08

4.4 Classification Models

After having finished feature preprocessing and engineering to the original dataset, generated dataframe now includes attributes that make fraudulent transactions efficiently detectable. We split our processed data to train set with 75% and test set with 25% respectively and used 10 K-Fold cross validation to test the performance of our final model. Inspired by this, we apply a variety of supervised learning and anomaly detection approaches. It is time to make the last effort for training a robust external fraud detection model using preeminent machine learning algorithms.

4.4.1 K-Nearest Neighbor classifier model

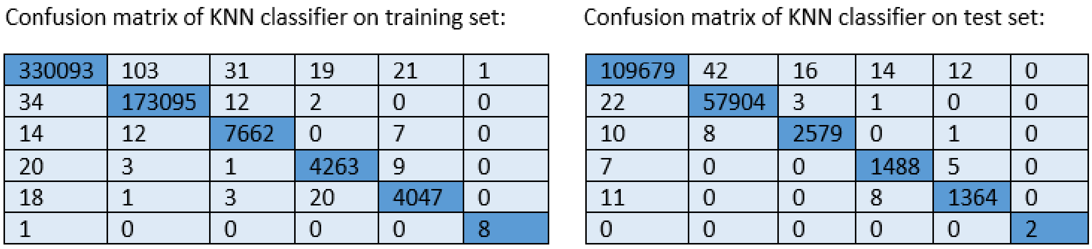
Cross-validation, mean score: 0.9991

Model accuracies of KNN classifier on train and test data are 0.9993 and 0.9990 respectively, as Table 4.12 below.

Table 4.12 : The model accuracy of KNN classifier on train and test data.

N-neighbors Classifier	
Model Accuracy	
Evaluating the model (training data)	Validating the model (test data)
0.9993	0.9990

Confusion matrix of KNN classifier on train and test data is displayed as Figure 4.6.



Confusion matrix of KNN classifier on training set:

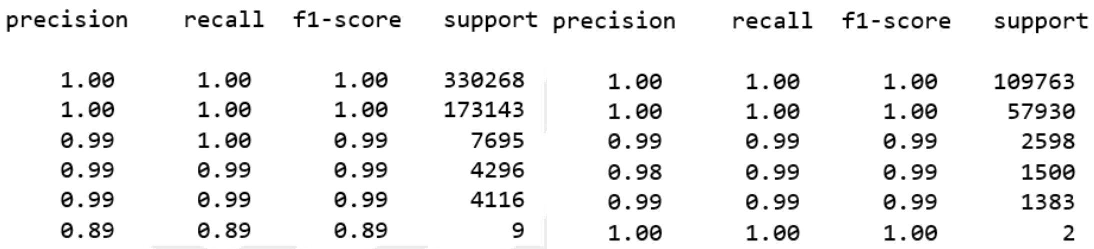
330093	103	31	19	21	1
34	173095	12	2	0	0
14	12	7662	0	7	0
20	3	1	4263	9	0
18	1	3	20	4047	0
1	0	0	0	0	8

Confusion matrix of KNN classifier on test set:

109679	42	16	14	12	0
22	57904	3	1	0	0
10	8	2579	0	1	0
7	0	0	1488	5	0
11	0	0	8	1364	0
0	0	0	0	0	2

Figure 4.6 : Confusion matrix of KNN classifier on train and test data.

Figure 4.7 describes the classification report of KNN classifier on train and test data.



```
precision     recall   f1-score    support  precision     recall   f1-score    support

     1.00       1.00       1.00     330268       1.00       1.00       1.00     109763
     1.00       1.00       1.00     173143       1.00       1.00       1.00      57930
     0.99       1.00       0.99       7695       0.99       0.99       0.99       2598
     0.99       0.99       0.99       4296       0.98       0.99       0.99       1500
     0.99       0.99       0.99       4116       0.99       0.99       0.99       1383
     0.89       0.89       0.89          9       1.00       1.00       1.00          2
```

Figure 4.7 : Classification report of KNN classifier on train and test data.

4.4.2 Naive Bayes classifier model

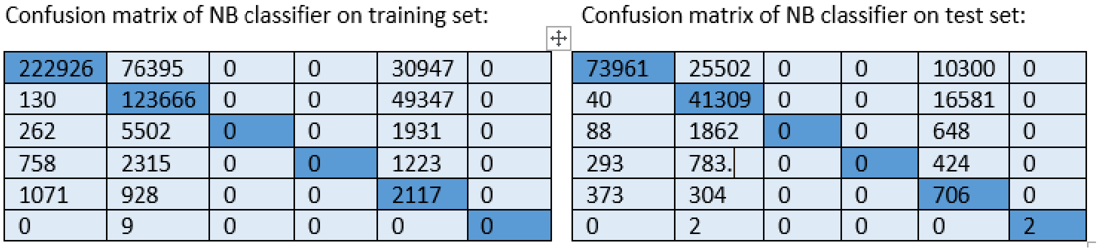
Cross-validation, mean score: 0.6684

Model accuracies of NB classifier on train and test data are 0.6712 and 0.6697 separately as in Table 4.13.

Table 4.13 : The model accuracy of NB classifier on train and test data.

Naïve Bayes Classifier	
Model Accuracy	
Evaluating the model (training data)	Validating the model (test data)
0. 6712	0.6697

Figure 4.8 depicts the confusion matrix of NB classifier on train and test data.



Confusion matrix of NB classifier on training set:

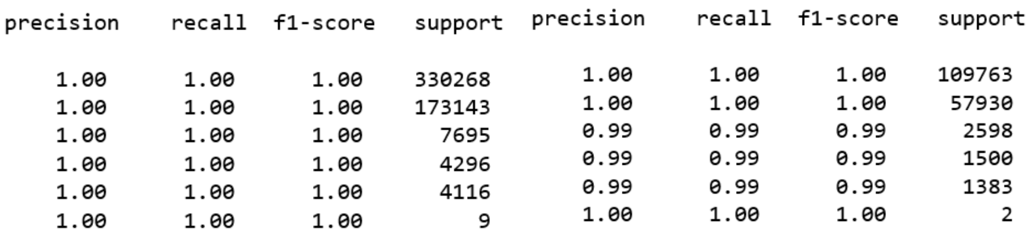
222926	76395	0	0	30947	0
130	123666	0	0	49347	0
262	5502	0	0	1931	0
758	2315	0	0	1223	0
1071	928	0	0	2117	0
0	9	0	0	0	0

Confusion matrix of NB classifier on test set:

73961	25502	0	0	10300	0
40	41309	0	0	16581	0
88	1862	0	0	648	0
293	783.	0	0	424	0
373	304	0	0	706	0
0	2	0	0	0	2

Figure 4.8 : Confusion matrix of NB classifier on train and test data.

Classification report of NB classifier on train and test data is displayed in Figure 4.9 below.



```
precision    recall  f1-score   support  precision    recall  f1-score   support

    1.00      1.00      1.00    330268      1.00      1.00      1.00    109763
    1.00      1.00      1.00    173143      1.00      1.00      1.00     57930
    1.00      1.00      1.00      7695      0.99      0.99      0.99      2598
    1.00      1.00      1.00      4296      0.99      0.99      0.99      1500
    1.00      1.00      1.00      4116      0.99      0.99      0.99      1383
    1.00      1.00      1.00         9      1.00      1.00      1.00         2
```

Figure 4.9 : Classification report of NB classifier on train and test data.

4.4.3 Random Forest classifier model

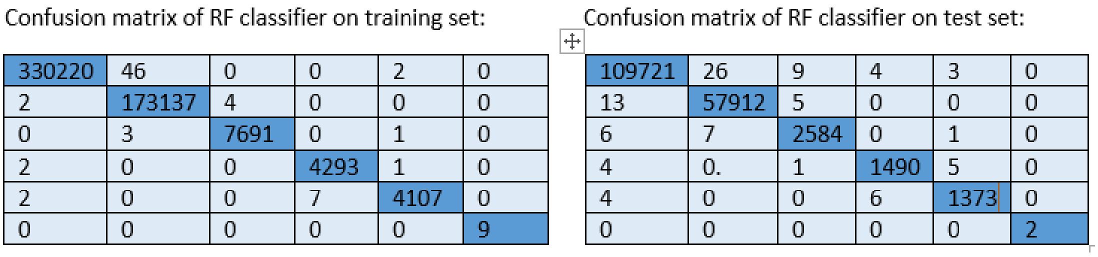
Cross-validation, mean score: 0.9994

Model accuracies of RF classifier on train and test data achieved 0.9998 and 0.9994 respectively, described like Table 4.14.

Table 4.14 : The model accuracy of RF classifier on train and test data.

Random Forest Classifier	
Model Accuracy	
Evaluating the model (train data)	Validating the model (test data)
0.9998	0.9994

Confusion matrix of RF classifier on train and test data is illustrated as Figure 4.10.



Confusion matrix of RF classifier on training set:

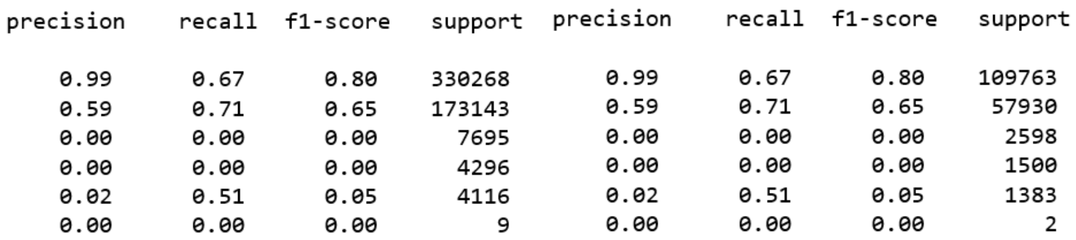
330220	46	0	0	2	0
2	173137	4	0	0	0
0	3	7691	0	1	0
2	0	0	4293	1	0
2	0	0	7	4107	0
0	0	0	0	0	9

Confusion matrix of RF classifier on test set:

109721	26	9	4	3	0
13	57912	5	0	0	0
6	7	2584	0	1	0
4	0.	1	1490	5	0
4	0	0	6	1373	0
0	0	0	0	0	2

Figure 4.10 : Confusion matrix of RF classifier on train and test data.

Classification Report of RF classifier on train and test set are described in detail in Figure 4.11.



```
precision    recall  f1-score   support precision    recall  f1-score   support

     0.99      0.67      0.80    330268      0.99      0.67      0.80    109763
     0.59      0.71      0.65    173143      0.59      0.71      0.65     57930
     0.00      0.00      0.00      7695      0.00      0.00      0.00      2598
     0.00      0.00      0.00      4296      0.00      0.00      0.00      1500
     0.02      0.51      0.05      4116      0.02      0.51      0.05      1383
     0.00      0.00      0.00         9      0.00      0.00      0.00         2
```

Figure 4.11 : Classification report of RF classifier on train and test data.

4.4.4 Logistic Regression classifier model

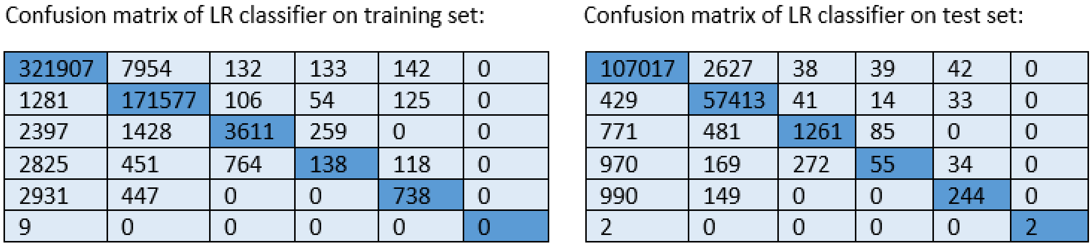
Cross-validation, mean score: 0.9582

Model accuracies of LR classifier on train and test data are both 0.9585, as described obviously in Table 4.15.

Table 4.15 : The model accuracy of LR classifier on train and test data.

Logistic Regression Classifier	
Model Accuracy	
Evaluating the model (Train set)	Validating the model (Test set)
0.9585	0.9585

Figure 4.12 depicts the confusion matrix of LR classifier on train and test data.



Confusion matrix of LR classifier on training set:

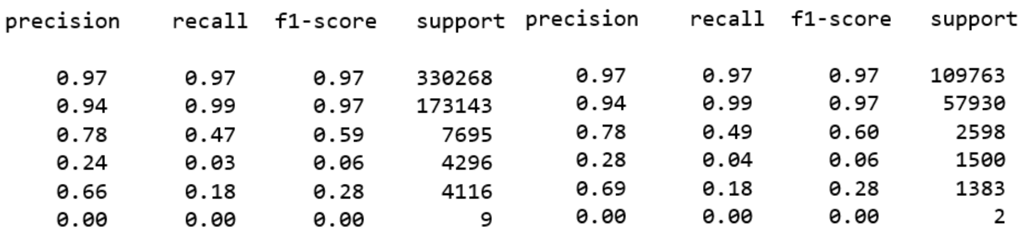
321907	7954	132	133	142	0
1281	171577	106	54	125	0
2397	1428	3611	259	0	0
2825	451	764	138	118	0
2931	447	0	0	738	0
9	0	0	0	0	0

Confusion matrix of LR classifier on test set:

107017	2627	38	39	42	0
429	57413	41	14	33	0
771	481	1261	85	0	0
970	169	272	55	34	0
990	149	0	0	244	0
2	0	0	0	0	2

Figure 4.12 : Confusion matrix of LR classifier on train and test data.

Last but not least, Figure 4.13 illustrates the Classification report of LR classifier on train and test data.



```
precision    recall  f1-score   support precision    recall  f1-score   support

     0.97      0.97      0.97    330268      0.97      0.97      0.97    109763
     0.94      0.99      0.97    173143      0.94      0.99      0.97     57930
     0.78      0.47      0.59      7695      0.78      0.49      0.60      2598
     0.24      0.03      0.06      4296      0.28      0.04      0.06      1500
     0.66      0.18      0.28      4116      0.69      0.18      0.28      1383
     0.00      0.00      0.00         9      0.00      0.00      0.00         2
```

Figure 4.13 : Classification report of LR classifier on train and test data.

5. RESULTS AND CONCLUSIONS

No challenge is more dangerous now than the one that the number of false positives during the process of generating classification and detecting models, while the significant number of them is turning out to be normal transactions anyway. Therefore, improving the accuracy of external fraud detection is a key to success in these kind of cases. To prevent mentioned typical problems, we will determine to use network-based and anomaly-based IDSs for the methodology of our research. Defense in depth, one of the key essence of information and network security, generally applies in the case of network security. The principle of defense in depth declares that organizations should demonstrate multifarious and overlapping security controls to achieve the identical control objective. This kind of layered approach detects opposed to the failure of any isolated security control. If one single control fails, there is still another isolated control established to achieve the identical security objective standing in it's place. In order to understand what the protocols do, a direct comprehension of the OSI model and the encapsulation procedure is essential to do efficacious packet investigations. The seven layers OSI model systematizes the functions of data transformation by seperating it into isolated layers. We narrated the main function of each layer by describing about some common protocols in seven layers in ISO model and mentioned the protocol data unit, which defines the shape of the data positioning in that layer, and we also summarized about any addresses that are needed.

We trained our data on the training set and test the performance of the benchmark machine learning model mentioned in the previous section. We had chosen Random Forest classifier as our benchmark model and we also used other kinds of classification algorithms since we have a classification problem to solve.

5.1 Comparison Results Between Proposed Models

In this thesis, we came to a conclusion that, firewall, IDS and IPS are not the same creatures or engines. As narrated, defense in depth is the efficacious measure we are looking to allocate. A Firewall is a control mechanism applied to obstruct and restrict the protocols traversing between two networks at layers 3 named network layer, and layer 4 named transport layer. In some cases, the firewall can perform restricted inspection of layers 5, 6 and 7, named session layer, presentation layer, and application layer. But in those instances, it's straightforwardly attempting to do complementary commission, which it may not perform well because firewalls don't have the processing muscle to do protocol analysis. Eventually, we can think of a firewall as a tool to control protocols. An IDS is not a control mechanism. It has much more processing power than a typical firewall, but generally less throughput since it is performing much more work by taking much more time. An IDS can detect intrusions but it cannot control them. It cannot function as a firewall and it cannot function as an IPS. An IDS can perform detection in layers 2 through 7 in proposed OSI model. An IPS is a control mechanism, while an IDS is being with the ability to control frames and packets in the same layers 2 through 7. Moreover, they have much more processing power than a firewall, but generally less throughput due to the inspection it must perform. An IPS may have the ability to perform many firewall-like functions, while the IPS is generally more difficult to administer when deployed for that function since it's designed to detect exploits and prevent attacks, not act like a firewall.

The IDS and IPS sensors were placed in front of the firewall and behind the firewall in the network of our organization. Routers on a DMZ network shared segment would filter traffic before it even reaches the firewall. Similarly, an intrusion prevention and detection systems were sit in front of and behind the firewall, filtering out potentially malicious traffic that manages to pass through the firewall before it reaches the eternal network. However, they could not be placed in the firewall, since the monitoring process of firewall is faster than the other detection systems, while IDS and IPS engines taking a lot of time by generating and testing the network intrusion detection classifiers. Defense in depth is a time-tested security principle and it certainly applies to network security. It is actually the process not a product, and is a proactive approach to thinking about security from the inside out. The major supremacy of anomaly-based and network-based IDS/IPS is the capability to distinguish undefined malicious traffics.

Various kind of preeminent classifiers generate various results. As is depicted clearly in Figure 5.1, the comparison results of different classification models on total training set which is the 75% of the total set, supervised learning algorithm with random forest function has the best performance compared to other classification algorithms.

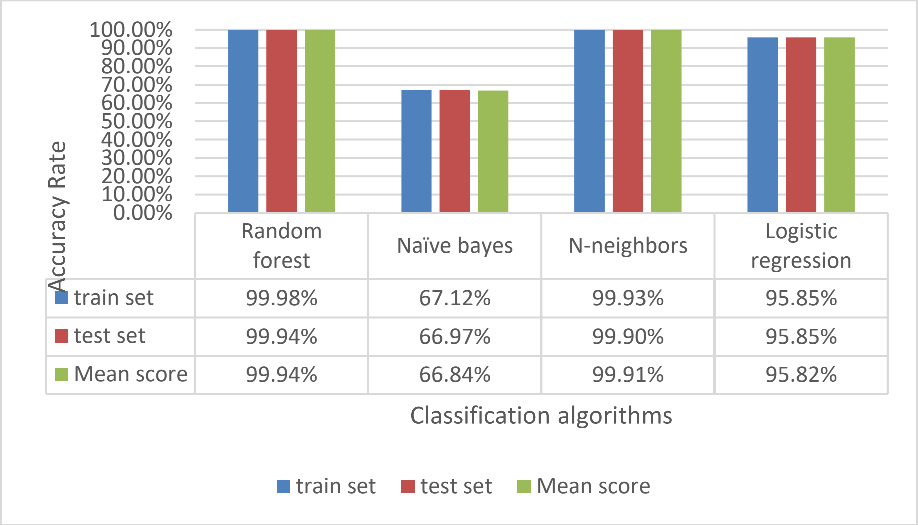


Figure 5.1 : The comparison result of different classification models on the total training set.

As is illustrated obviously in Figure 5.2 below, the comparison results of different classification models on total test set which accounts the 25% of the total set, supervised learning algorithm with random forest function has the best performance compared to other classification algorithms, while Adaboost and KNN classifiers perform as well as RF does.

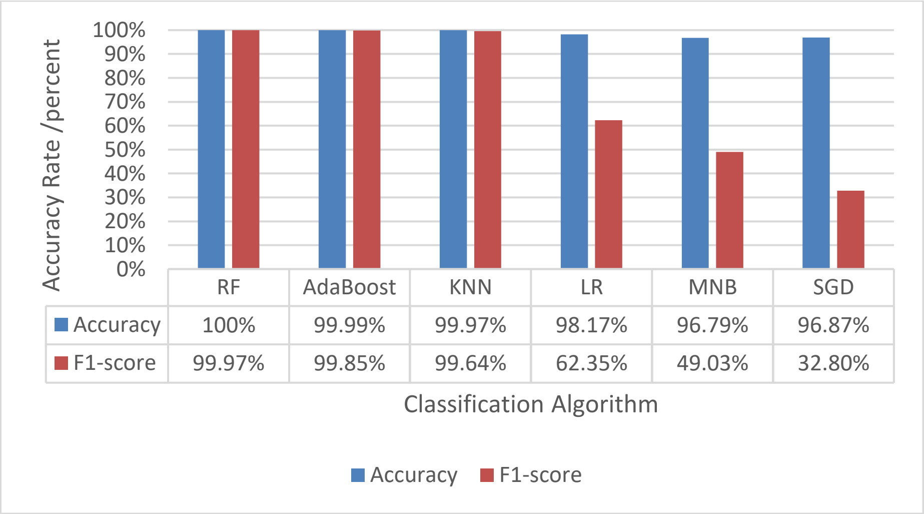


Figure 5.2 : Comparison result of different classification models on total test set.

5.2 Comparison of the Proposed External Fraud Classifier Performance with Existing Work

In general, we should compare our results with existing research performed by the same machine learning and deep learning algorithms that we used, using the same dataset. Unfortunately, since the dataset we used in this research has been generated at the begin of this year 2018, by Canadian Institute. There are not enough researches on this field to compare but the one [35]. Authors fist extract all existed traffic features from the original dataset and distinguish the best short feature set, then detect each attack family using different kind of classification and anomaly detection algorithms. Table 5.1 depicts the comparison of our IDS classifiers with previous work, we can obviously discover the fact that the performance and accuracy of our result with the most important selected features are better than the previous research. But we have still performed additional analysis and compared the results with related existing research, despite different dataset resources.

Table 5.1 : Comparison of the IDS classifiers performance with existing work.

Previous research				Our research			
	Pr	Rc	F1		Pr	Rc	F1
KNN	0.960	0.960	0.960	KNN	0.999	0.990	0.996
RF	0.980	0.970	0.970	RF	0.990	0.994	0.999
Adaboost	0.770	0.840	0.770	Adaboost	0.999	0.970	0.998
NB	0.88	0.04	0.04	LR	0.958	0.970	0.623
MLP	0.77	0.83	0.76	MNB	0.669	0.668	0.490
ID3	0.98	0.98	0.98	SGD	0.968	0.890	0.328

5.3 Conclusions

We meticulously examine the data to gain a reliable understanding in which attributes could be removed and which could be efficiently engineered. The dimensionality reduction algorithms and the result of correlation matrix between features can help to generate the classification model with high accuracy and a short time. Then we mainly trained four types of machine learning classifiers and tried to make our mind to which one could be more reliable and effective in detecting network external fraud transactions. Out these models, Random forest classifier gave us the better results than any other did, but it would tough to deduce the best model against others since both all other models can perform very well handling high dimensional data. All models have its pros and cons, with all that said our final solution model would be Random forest classifier. Because of expeditiously increased processing of data, discussed traditional methods of network security are rapidly failing to detect and prevent efficaciously. Redundant features or observations of such enormous volume of data make training time long and ever, also decrease the efficiency of external fraud IDS. Proposed RFC and GBC shortened the network training and test time by transforming or updating the original raw data into low dimensional one. Furthermore, generated external fraud detection system combined with random forest algorithm achieved prosporous result in an actual public network environment. Since real and complex environment that our dataset possesses, training and testing time will take a lot of time, so we decided to apply GPU acceleration technology, to build a high-performance external fraud IDS trained in a short time.

5.4 Future Work

The projects of this research mainly discussed various types of machine learning approaches to detect and classify external frauds like real-time network transactions. Generated experimental results concluded that machine learning techniques contribute to ameliorate the false positive issues in network intrusion detection systems. The use of deep learning has in last few decades been outstanding due to its efficiencies in detecting fraudulent network traffics. In the near future, we will try to use more prominent deep learning techniques to the real-time dataset to put all those theories into practice for practical using.

REFERENCES

[1] **Tioton, H. F. and Krause, M.** (2004). Information Security Management Handbook. Fifth edition. CRC Press LLC. Washington, USA.

[2] **Seymour, B. K. and Eric, W.** (2014). Computer Security Handbook. Sixth edition. *Willey*, USA.

[3] **Canavan, J. E. (**2001). Fundamentals of Network Security. *Artech House*. Boston, London.

[4] Reports to the Nations on Occupational Fraud and Abuse. *Association of Certified Fraud Examine.* 2016 Global Fraud Study, USA.

[5] **Peaston, S.** (2017). External and Internal Fraud Threats – Essential Reading for Fraud and Financial Crime Strategies. *Fraudscape, UK.*

[6] **Dieter, G.** (2011). *Computer Security*, Third edition, Hamburg University of Technology, Willey.

[7] **Kim J., Huong L. T. T., and Kim H.** (2015). Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. *International Conference on PlatCon.* South Korea: Jeju, 15-17 February.

[8] **Tuan A. T., Lotfi M., Des M., Syed A. R. Z., and Mounir G. (**2016). Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. *International Conference on WINCOM.* Morocco: Fez, 26-29 October.

[9] **Nataraj, L. and Karthikeyan** (2011). Malware Image: Visualization and Automatic Classification. 8th International Symposium on Visualization for Cyber Security, 20 July, Pennsylvania, USA.

[10] **Bo, D. and Xue, W.** (2016). Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection. *IEEE 8th International Conference on ICCSN*, 581-585. China: Beijing. 4-6 June.

[11] **Mohd R. Z.A., Megat F. Z., Shadil A. Z., and Hassan D. (**2016). Anomaly-Based NIDS: A Review of Machine Learning Methods on Malware Detection. *International Conference on ICICTM*, 266-270. Malaysia: Kuala Lumpur, 16-17 May.

[12] **Guangzhen Z., Cuixiao Z., and Lijuan Z.** (2017). Intrusion Detection using Deep Belief Network and Probabilistic Neural Network. *IEEE International Conference on CSE and EUC.* 639-642. China: Guangzhou, 21-24 July.

[13] **Zahangir, M. A. and Tarek, M. T.** (2017). Network Intrusion Detection for Cyber Security on Neuromorphic Computing System. *International Joint Conference on IJCNN*. 3830-3837. USA: Anchorage, 14-19 May.

[14] **Nguyen T. V., Tran N. T., and Thanh L. S.** (2017). An Anomaly-based Network Intrusion Detection System Using Deep Learning. *International Conference on ICSSE.* 210-214. Vietnam: Ho Chi Minh, 21-23 July.

[15] **Vinayakumar R., Soman K., and Prabaharan P.** (2017). Applying Convolutional Neural Network for Network Intrusion Detection. *International Conference on ICACCI.* 1222-1228. India: Udupi, 13-16 September.

[16] **Norbert A., Branislav M., Anton B., and Tomáš P. (**2017). Artificial Neural Network based IDS. *IEEE 10th International Symposium on SAMI.* 159-164. Slovakia: Herl'any, 26-28 January.

[17] **Timenko V., Gajin S.**, 2017. Ensemble Classifiers for Supervised Anomaly based Network Intrusion Detection. *IEEE 13th International Conference on ICCP.* 13-19. Romania: Cluj-Napoca, 7-9 September.

[18] **Yin C., Zhu Y., Fei J., and He X. (**2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access.* 21954-21961. China: Zhengzhou.

[19] **Taro I., Ryoichiro O., Tetsuya O., Leonard B.** (2017). Application of Deep Recurrent Neural Networks for Prediction of User Behavior in Tor Networks. *31th International Conference on WAINA.* Taiwan: Taipei, 27-29 March.

[20] **Georgi A. A., Nareg A., Imad H. E., Ayman K, and Ali C.** (2017). Flow-Based Intrusion Detection System for SDN. *IEEE Symposium on ISCC.* Greece: Heraklion, 3-6 July.

[21] **Phai V. D., Tran N., Nathan S., Áine M., and Qi S. (**2017). Deep Learning Combined with De-noising Data for Network Intrusion Detection. *21st Pacific Symposium on IES.* 55-60. Vietnam: Hanoi, 15-17 November.

[22] **Letou, K. and Devi, D.** (2013). Host-based Intrusion Detection and Prevention System. International Journal of Computer.

[23] **Nathan S., Tran N., Phai V. D., and Qi S.** (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on ETCI,* 41-50.

[24] **Nicholas, F. R.** (2016). Understanding Payment Card Fraud through Knowledge Extraction from Neural Networks using Large-scale Datasets. PhD Thesis. Department of Computer Science. University of Surrey.

[25] **Cole E., Krutz R., and Conley J. W.,** (2005). Network Security Bible. *Wiley*. Canada.

[26] AWS Best Practices for DDoS Resiliency. (2016). Amazon Web Services, Inc. or its affiliates. All rights reserved, June.

[27] **Dey D., Dinda A., and Kundapur P. P.** (2017). Warezmaster and Warezclient: an Implementation of Ftp based R2l Attacks. 8th international conference on ICCCNT, India: Delhi.

[28] **James E. G., Wolfgang K. H., and Yuichi M.** (2012). Handbooks of Computational Statistics. Springer. London.

[29] **Ghorbani, A. A. and Lu, W.** (2010). Network Intrusion Detection and Prevention. Advances in information security. *Springer*. Canada.

[30] **Emmanuel, S. P. and Joshi, R.C**, (2016). Fundamentals of Network Forensics. Computer Communications and Network, Springer, Swindon, UK.

[31] **Li W., Yi P., and Pan Li,** (2014). A New Intrusion Detection System based on KNN Classification Algorithm in Wireless Sensor Network. Journal of Electronical and Computer Engineering, China.

[32] **Alpaydin, E.** (2010). Introduction to Machine Learning. Second edition, Library of Congress Cataloging-in-Publication Information. London, England.

[33]**Url-1**<https://www.kaggle.com/ntnu-testimon/paysim1/data>, data retrieved 06.10.2016.

[34]**Url-2**<https://www.kaggle.com/ntnu-testimon/paysim1/data>, NSL-KDD Dataset Description, Improvement to the KDD' Dataset. data retrieved 27.06.2000.

[35] **Sharafaldin I., Lashkari A., and Ghorbani A.** (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th ICISSP, 108-116, Purtogal, January.

[36]**Url-3**<https://www.unb.ca/cic/datasets/index.html>, CICIDS 2017, IPS/ IDS dataset, date retrieved 07.07.2017.

CURRICULUM VITAE

Name Surname **:** Aji MUBALAIKE

Place and Date of Birth **:** Xinjiang/ China

E-Mail **:** mubalaike15@itu.edu.tr

EDUCATION **:**

- **B.Sc.** **:** 2014, Xinjiang University, Faculty of Computer Science and Technology, Computer Engineering
- **M.Sc.** **:** 2018, Istanbul Technical University, Faculty of Informatics, Information Security Engineering and Cryptography

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

• **Aji, M.** ADALI E. (2017). Multilayer Perceptron Neural Network Technique for Fraud Detection. *International Conference - UBMK*, October 05-08, Antalya, Turkey.

• **Aji, M.** ADALI, E. (2018). Deep Learning Approach for Intelligent Financial Fraud Detection System. *International Conference - UBMK*, September 20-23, Sarajevo, Bosnia.