UNIVERSITE BRETAGNE LOIRE

université angers

Thèse de Doctorat

Abdel Karim KASSEM

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université d'Angers
sous le sceau de l'Université Bretagne Loire

École doctorale : *Sciences et technologies de l'information et mathématiques*
Discipline : *Informatique et applications*
Spécialité : *Informatique*
Unité de recherche : *Laboratoire angevin de recherche en Ingénierie des Systèmes*
Soutenue le *23 juillet 2021*

Système intelligent basé sur des techniques de l'apprentissage automatique pour l'évaluation de la sécurité et la détection des cyber-intrusions

Jury

Rapporteurs :	**Mohammad HAJJAR, Professeur des Universités, Doyen de la Faculté de Technologie, Liban** **Michaela GEIERHOS, Professeure des Universités, Université de la Bundeswehr, Allemagne**
Examinateurs :	**Abd El Salam AL HAJJAR, Professeur des Universités, Université Libanaise, Liban** **Olivier BARTHEYE, Maitre de Conférence – Ecole de l'Air, France**
Directeur de Thèse :	**Pierre CHAUVET, Directeur délégué IMA, Université Catholique de l'ouest, France**
Co-directeur de Thèse :	**Bassam DAYA, Professeur des Universités, Université Libanaise, Liban**

L'Université Bretagne Loire

L'auteur du présent document vous autorise à le partager, reproduire, distribuer et communiquer selon les conditions suivantes :



- Vous devez le citer en l'attribuant de la manière indiquée par l'auteur (mais pas d'une manière qui suggérerait qu'il approuve votre utilisation de l'œuvre).
- Vous n'avez pas le droit d'utiliser ce document à des fins commerciales.
- Vous n'avez pas le droit de le modifier, de le transformer ou de l'adapter.

Consulter la licence creative commons complète en français :
http://creativecommons.org/licences/by-nc-nd/2.0/fr/



REMERCIEMENTS

First of all, I praise God for providing me the opportunity and granting me the capability to proceed the thesis successfully.

I would like to express my gratitude and appreciation to my supervisors: Professors Bassam DAYA and Pierre CHAUVET for their potential to pursue this study under their direction. They gave me the guidance, inducement and patience as a major support to my study.

Furthermore, I am very thankful to the official referees of my thesis: Pr. Mohammad HAJJAR for presiding the jury; Prof. Michaela GEIERHOS for reviewing my thesis; Prof. Abd El Salam AL HAJJAR and Dr. Olivier BARTHEYE for their valuable comments.

I cannot finish without expressing my gratefulness for all my family, I warmly appreciate my beloved parents for their tenderness and love support since my birth. Finally, I want to express my gratitude and extensive appreciation to my lovely wife "Boshra", thank you for your continuous support and encouragement as well as for bearing with me the life pressures during my study.

My princess daughter "Lea", thanks god for your presence in my life.

Résumé

L'apprentissage automatique est devenu une technologie décisive pour la cyber sécurité dans le but de protéger les réseaux et systèmes informatiques contre les cybercriminels. En conséquence, l'objectif de cette thèse est d'améliorer le mécanisme de sécurité appliquée, et proposer un système intelligent basé sur des techniques d'apprentissage automatique pour la détection des cyber-intrusions. Nous avons donc appliqué la technique de test de pénétration permettant de découvrir les vulnérabilités concernant les attaques les plus courantes. Plus tard, nous avons fourni des suggestions de sécurité et des solutions concernant ces cyber-attaques risquées. De plus, nous avons appliqué les techniques de web mining pour identifier plusieurs approches en termes de comportement des visiteurs et d'évaluation de la cyber sécurité. Par la suite, nous avons parvenu à détecter l'activité des visiteurs, leur comportement, le contrôle des ressources d'accès et les menaces qui peuvent affronter le serveur web. Ensuite, un système intelligent de détection d'intrusion hôte (HIDS : Host-based Intrusion Detection System) a été développé en utilisant les techniques de text mining. Pour cela, nous avons construit un ensemble de données de classification de texte fiables comprenant 6000 enregistrements d'URL malveillantes. Ce type de données nous a amené à proposer le modèle DOC2VEC comme méthode de représentation de caractéristiques dans notre HIDS. De plus, nous avons appliqué plusieurs techniques d'apprentissage automatique. Par conséquent, le perceptron multicouche (multilayer perceptron MLP) s'est avéré être le modèle le plus précis à 90,67% pour détecter les attaques SQLi, XSS ainsi que les attaques par traversée de répertoires. En outre, nous avons développé un nouveau système intelligent de sécurité appelé SIS-ID adopté pour détecter les dernières URL malveillantes et étendu aux attaques par déni de service distribuées (DDoS). De plus, notre système qui est basé sur plusieurs techniques d'apprentissage automatique a été examiné via deux bases de données configurées qui sont les DB-MALCURL et DB-DDOS extraites de l'institut canadien de cybersécurité (CIC). Ensuite, nous avons évolué les performances du système en utilisant nos méthodes d'optimisation d'apprentissage proposées. Ainsi, le SIS-ID a atteint la meilleure précision (98,52%) basé sur le modèle de vote qui détecte l'attaque d'URL malveillantes. D'autre part, le modèle stacking a enregistré la précision maximale (77,04 %) pour détecter l'attaque DDOS. Finalement, nous avons validé notre proposition de SIS-ID à l'aide d'un matériel basé sur la simulation en temps réel au sein de l'université libanaise. Par conséquent, le matériel a été configuré sur la base du modèle facteur de valeur aberrante locale (LOF) qui a atteint l'efficacité d'éviter une attaque par déni de service (DOS) effectuée sur une scène en temps réel.

Mots Clés: cyber-sécurité, vulnérabilités, cybercriminels, cyberattaques, web mining, système de détection d'intrusion, apprentissage automatique, temps réel.

Abstract

Machine learning has become a decisive technology for cybersecurity to protect the computer networks and systems against cybercriminals. Consequently, the aim of our conducted thesis is to enhance the applied security mechanism and to propose an intelligent system using machine learning techniques for cyber intrusion detection. Therefore, we applied the penetration testing technique, it permits the discovering of vulnerabilities for the most popular attacks. Hence, we provided security suggestions and solutions concerning these risk cyber-attacks. In addition, we applied the web mining techniques to identify several approaches in terms of the visitor behavior and the cyber security evaluation. Afterwards, we achieved the detection of the visitor activity, its behavior, the access resources control and the threats that may face the web server. Then, an intelligent host based intrusion detection system (HIDS) has been developed using the text mining techniques. Thus, we constructed a reliable textual dataset which includes 6000 records of malicious URLs. This kind of data derives us to propose the DOC2VEC model as a feature representation method in our HIDS. Additionally, we have applied several machine learning techniques. Hence, the multilayer perceptron found to be the most accurate model by 90.67% in detecting the SQLi, XSS and directory traversal attacks. Furthermore, we developed a new security intelligent system called SIS-ID adopted to detect the latest malicious URLs and expanded to the DDOS attacks. Moreover, our system that is based on several machine learning techniques was examined via two configured data bases which are the DB-MALCURL and DB-DDOS extracted from the Canadian institute for cybersecurity (CIC). Afterwards, we evolved the system performance using our proposed learning optimization methods. Eventually, the SIS-ID achieved the best accuracy (98.52%) based on the voting model that detects the malicious URLs attack. On the other hand, the stacking model recorded the top accuracy (77.04%) for detecting the DDOS attack. Ultimately, we validated our proposed SIS-ID using a hardware based-real-time simulation in the Lebanese university. Hence, the hardware was configured based on the local outlier factor model that achieved the efficiency of avoiding a performed denial of service attack (DOS) on real time stage.

Keywords: cyber security, vulnerabilities, cybercriminals, cyber-attacks, web mining, intrusion detection, machine learning, real time.

Summary

General Introduction 9

Background 9

Objective 12

Contribution 12

Report Structure 13

Part 1: State of The Art 15

Introduction 17

Chapter 1: Cyber Security and Rendered Services 19

1.1 Cyber Security Overview 19

1.2 Cyber Security Domains 20

1.3 Cyber Security Importance 21

1.4 Web Vulnerabilities 22

1.5 Security Technologies 22

1.6 Cyber Crimes 24

1.7 Cyber Attacks 29

Chapter 2: Web Mining Methodology 32

2.1 Web Mining Overview 32

2.2 Web Mining Techniques 32

2.1.1 Web Content Mining 33

2.1.2 Web Structure Mining 34

2.1.3 Web Usage Mining 35

2.3 Web Mining and Security Analytics 37

Chapter 3: Machine Learning Techniques 39

3.1 Machine Learning Overview 39

3.2 Machine Learning Types 39

3.3 Machine Learning Steps 40

3.4 Machine Learning Algorithms 41

3.5 Machine Learning Evaluation Metrics 50
Chapter 4: Intrusion Detection: Concept and Related Works 52
4.1 Intrusion Detection Overview 52
4.2 Intrusion Detection Systems 52
4.2.1 Deployment 53
4.2.2 Detection Methods and Responses 53
4.3 Most Used Open Sources IDSs 54
4.4 Intrusion Detection System based on Machine Learning Techniques 55
4.4.1 Requirements and Materials 57
Conclusion Part I 62
References Part I 63
Part 2: Enhancement of the Defense Level for the Employed Cyber Security Mechanisms in the Lebanese University 71
Introduction 73
Chapter 5: Web Attacks Penetration Testing and Analysis 76
5.1 General Overview 76
5.2 Applying the Penetration Testing 76
5.2.1 Security Testing and Penetration stages 77
5.3 Experimental Results: Security Suggestions and Solutions 81
5.3.1 Fixing the Vulnerabilities 81
5.3.2 Improving the Fundamental Web Server Security 83
5.3.3 Visitor's Behavior Analysis 84
Chapter 6: Detection of Visitor's Behavior based on Web Mining Techniques 85
6.1 General Overview 85
6.2 Designing and Applying the Web Usage Mining Tools 85
6.2.1 Tools Requirements and Implementation 86
6.2.1.1 Data Collection and Selection 86
6.2.1.2 Tool Selection 87
6.2.1.2.1 Deep Log Analyzer Tool 87
6.2.1.2.2 The Security Analysis Tool 88
6.3 Experimental Results and Analysis 92

6.3.1 The Deep Log Analyzer Tool Result 92
6.3.2 The Security Analysis Tool Result 95
Chapter 7: Host based Intrusion Detection System based on Text Mining and Machine Learning 97
7.1 General Overview 97
7.2 The Proposed HIDS Architecture 97
7.2.1 Data Collection 98
7.2.2 Data Preprocessing 99
7.2.2.1 Data Preparation 99
7.2.2.2 Data Cleaning 100
7.3 Feature Representation Method 101
7.3.1 DOC2VEC Model 103
7.4 The Applied Machine Learning Methods and Classification Preliminaries 105
7.5 Experimental Results and Discussion 108
Conclusion Part II 113
References Part II 114
Part 3: Security Intelligent System Based-Intrusion Detection using Machine Learning Techniques (SIS-ID) 116
Introduction 118
Chapter 8: Materials and Development Mechanism 120
8.1 General Overview 120
8.2 The SIS-ID Requirements 120
8.2.1 Data Gathering: Canadian Institute for Cyber-Security Datasets 120
8.2.1.1 DB-MALCURL Dataset Preparation 121
8.2.1.2 DB-DDOS Dataset Preparation 121
8.3 Data and Features Engineering 122
8.3.1 Data Preprocessing 122
8.3.2 Features' Technique 123
8.3.3 Selected Features 124
8.4 The Learning Methodology for SIS-ID System 125
8.4.1 The Applied Machine Learning Methods 126
8.4.2 Learning Implementation 130

8.4.3 Learning Optimization Method 131
Chapter 9: Results: SIS-ID Performance Evaluation 132
9.1 General Overview 132
9.2 Experimental Results and Discussion 132
9.2.1 Applying the SIS-ID System on DB-MALCURL 132
9.2.1.1 Supervised Learning 132
9.2.1.2 Ensemble Techniques 137
9.2.1.3 Evolving the Ensemble Techniques 140
9.2.2 Applying the SIS-ID system on DB-DDOS 144
9.2.2.1 Supervised Learning 144
9.2.2.2 Ensemble Techniques 148
9.2.2.3 Evolving the Ensemble Techniques 152
9.2.2.4 Unsupervised Learning 156
9.3 General Discussion and Evaluation 157
9.4 Hardware-Based Real-Time Simulation 160
Conclusion Part III 163
References Part III 164
General Conclusion and Future Work 165
Publications 169

List of Figures

Figure 1: The distribution of the internet users over the last decades 10
Figure 2: Formjacking attacks per month 11
Figure 3: Web attacks per day 11
Figure 4: Cyber security and its various domains[12] 20
Figure 5: The firewall and the VPN architecture within a network [https://www.cybertroninc.com] 23
Figure 6: The distribution of risks percentage over the companies' types 25
Figure 7: Top 20 international victims' countries [29]. 29
Figure 8: Web mining methodology and its techniques [45] 33
Figure 9: Web content mining structure [45] 34
Figure 10: Web usage mining process 36
Figure 11: Machine learning steps [https://datafloq.com/] 41
Figure 12: Support vector machine technique [https://medium.com/] 44
Figure 13: Random forest technique [74] 45
Figure 14: General architecture of the boosting classifier [https://cppsecrets.com/] 46
Figure 15: General architecture of the voting classifier [77] 47
Figure 16: General architecture of the bagging classifier [Nick Minaie, "The Data Scientist's Guide to Selecting Machine Learning Predictive Models in Python, https" ://towardsdatascience.com/] 48
Figure 17: General architecture of the stacking classifier 49
Figure 18: The local outlier factor formula 49
Figure 19: Network and host based IDSs topologies 53
Figure 20: Intrusion detection based on machine learning techniques 56
Figure 21: General architecture of the applied penetration testing 77
Figure 22: Applying the SQL injection attack on the login module 78
Figure 23: SQL injection attack to destroy the student's accounts 79
Figure 24: Applying the XSS Attack on the ccne web pages 79
Figure 25: The achieved result using the XSS attack on the drop box page 80
Figure 26: Applying the sensitive data exposure 80
Figure 27: Architecture of the proposed web usage mining tools 86
Figure 28: The architecture of web usage mining methodology 88
Figure 29: Picking the input data in the selected tool 89
Figure 30: Example of results after the segmentation process 90
Figure 31: The achieved results using the Deep Log Analyzer tool 92
Figure 32: A summary about the visitor's general activity 93
Figure 33: A summary for the technical summary result 93
Figure 34: Visitor's spending time on the faculty website 93
Figure 35: Some of the top downloaded files 94
Figure 36: A summary about the top accessed directories 94
Figure 37: Results about the occurred web server errors 95
Figure 38: The security analysis tool with our achieved results 95
Figure 39: The results of the security analysis tool with the detected cyber-attacks 96
Figure 40: The Proposed HIDS architecture with the implementation phase 98

Figure: 41 The log file of CLF format 99
Figure 42: An example about the data during the preprocessing stage 100
Figure 43: Part of the generated dataset including the three kinds of attacks 101
Figure 44: A log file with preprocessing steps 101
Figure 45: An example for the proposed Doc2vec model [15] 102
Figure 46: Document vector space plot 105
Figure 47: Artificial neural network: the MLP model 107
Figure 48: MLP ROC curve 109
Figure 49: KNN ROC curve 109
Figure 50: Decision Tree ROC curve 110
Figure 51: SVM ROC curve 110
Figure 52: Confusion matrix for the Decision Tree model 111
Figure 53: Confusion matrix for the MLP model 111
Figure 54: Confusion matrix for the SVM model 112
Figure 55: Confusion matrix for the KNN model 112
Figure 56: Data Preprocessing Workflow 122
Figure 57: General architecture of the SIS-ID learning methodology based on the applied machine learning techniques 126
Figure 58: The proposed pseudocode that was applied in the SIS-ID learning implementation 131
Figure 59: Confusion matrix for OVR model on the DB- MALCURL 134
Figure 60: Confusion matrix for OVO model on the DB- MALCURL 135
Figure 61: Confusion matrix for KNN model on the DB-MALCURL 136
Figure 62: Confusion matrix for Decision Tree model on the DB-MALCURL 136
Figure 63: Confusion matrix for XGBoost model on the DB-MALCURL 138
Figure 64: Confusion matrix for Random Forest model on the DB- MALCURL 139
Figure 65: Confusion matrix for Adaboost model on the DB- MALCURL 140
Figure 66: Confusion matrix for Voting model on the DB-MALCURL 142
Figure 67: Confusion matrix for Stacking model on the DB-MALCURL 143
Figure 68: Confusion matrix for Bagging model on the DB-MALCURL 143
Figure 69: Confusion matrix for OVR model on the DB-DDOS 146
Figure 70: Confusion matrix for OVO model on the DB-DDOS 146
Figure 71: Confusion matrix for Decision Tree model on the DB-DDOS 147
Figure 72 Confusion matrix for KNN model on the DB-DDOS 148
Figure 73: Confusion matrix for XGBoost model on the DB-DDOS 150
Figure 74: Confusion matrix for Random Forest model on the DB-DDOS 151
Figure 75: Confusion matrix for Adaboost model on the DB-DDOS 152
Figure 76: Confusion matrix for Stacking model on the DB-DDOS 154
Figure 77: Confusion matrix for Voting model on the DB-DDOS 155
Figure 78: Confusion matrix for Bagging model on the DB-DDOS 156
Figure 79: Performance measurement of the LOF model that deployed in the proposed hardware 156
Figure 80: The general architecture of our SIS-ID hardware-based real-time simulation 160

Figure 81: The efficiency in detecting the advent attack on real time stage 161
Figure 82: The result of our hardware for avoiding coming DOS attack..... 162

List of Tables

Table 1: Most important tools according to the CIA triad 22
Table 2: List of the biggest corporate cyber-crimes in the last years 28
Table 3: Confusion Matrix architecture 50
Table 4: Most used open source IDSs 55
Table 5: Related works which used reliable data set in the IDS development 57
Table 6: Most employed data sources in the IDS based on the machine learning techniques 60
Table 7: The detection alarm rates for the IDS measurement based on the machine learning techniques 61
Table 8: Data collection description 86
Table 9: The basic information recorded from log file 87
Table 10: The proposed parameters used to be matched with the log file 89
Table 11: Some of the attack patterns extracted from our configured data base 91
Table 12: Applying the preprocessing process 101
Table 13: Word embedding and document vectors numbers resulted from the doc2vec model . 104
Table 14: Number of document vectors for each class 104
Table 15: The suggested splitting proportion for the training and testing stage 106
Table 16: The highest detection rate achieved by the classifiers for each attack 109
Table 17: Accuracy results of the different ML Techniques 110
Table 18: Classification report for the Decision Tree model 111
Table 19: Classification report for the MLP model 111
Table 20: Classification report for the SVM model 112
Table 21: Classification report for the KNN model 112
Table 22: The distribution of the classes for DB-MALCURL 121
Table 23: The distribution of the classes for DB-DDOS 122
Table 24: The example about the scaling method result 123
Table 25: Results of the applied supervised learning techniques tested via the DB-MALCURL 133
Table 26: A summary about the results for detecting each class with the measurements coefficient using the supervised learning models that tested via the DB-MALCURL 133
Table 27: Results of the applied ensemble techniques tested via the DB-MALCURL 137
Table 28: A summary about the results for detecting each class with the measurements coefficient using the ensemble techniques that tested via the DB-MALCURL 137
Table 29: Results of the proposed ensemble models tested via the DB-MALCURL 140
Table 30: A summary about the results for detecting each class with the coefficient measurements using our proposed ensemble models that tested via the DB-MALCURL 141
Table 31: Results of the applied supervised learning techniques tested via the DB-DDOS 144
Table 32: A summary about the results for detecting each class with the coefficient measurements using the supervised learning models that tested via the DB-DDOS 145

Table 33: Results of the applied ensemble techniques tested via the DB-DDOS 148
Table 34: A summary about the results for detecting each class with the coefficient measurements using ensemble techniques that tested via the DB-DDOS..... 150
Table 35: Results of the proposed ensemble models tested via the DB-DDOS 152
Table 36: A summary about the results for detecting each class with the coefficient measurements using our proposed ensemble models that tested via the DB-DDOS 153
Table 37: The top models results for detecting each attack using the DB-MALCURL..... 157
Table 38: The detection rates achieved by the top models for each attack using the DB-MALCURL 157
Table 39: Results of the different ML Techniques tested via the DB-MALCURL 158
Table 40: Comparative study between the SIS-ID tested via the DB-MALCURL and the CIC Laboratory [3]..... 158
Table 41: The top models results for detecting each attack using the DB-DDOS 159
Table 42: The detection rates achieved by the top models per each attack using the DB-DDOS 159
Table 43: Results of the different ML Techniques tested via the DB-DDOS 160
Table 44: Comparative study between the SIS-ID tested via the DB-DDOS and the CIC Laboratory [2]..... 160

General Introduction

Background

Nowadays, the Internet is a popular topic amid the information technology scientists, software developers and cyber-security researchers. The use of internet has increased; currently people are more likely to be connected and linked to these sociable networks. It is a worldwide interconnection as the internet is rapidly becoming a key gadget in the recent socio-economy situation which provides swift and resilient information that can be shared among internet users and intelligent businesses. It includes reinforcing techniques that offer interesting features: simplicity, framework and language autonomy, availability and synergy with external community, etc. This is predominantly important to enable cooperation among the components of diversified systems that evolve to embrace new technological devices in terms of computing telecommunications such as traditional data, intelligent portable devices, cloud computing deployment and Internet of Things (IoT) devices.

Several activities related to business, money exchanges, interchanges and administration of indispensable activities are currently executed by using the internet. Most organizations are associated constantly with the internet using applications that allow a huge number of users to access and extract vast amounts of information and data services in an enormous scope of areas. Thus, secure media have been carefully used to carry out most of these tasks, several anxieties were identified with spying and data loss. In addition, the internet covers many parts of our daily life concerning significant activities such as Social Networking, education and technology, entertainment and online services.

Cybersecurity ventures predict that internet users will reach 6 billion by 2022 while the world population is near to 8 billion, that means the projected percentage is 75 %, and in 2030, they assume that it may be extended to 90 percent of the whole world population [1].

Regarding to the internet World Stats which is an international statistical organization that covers more than 242 different countries and regions, the users over the internet constitute more than 58% of the world inhabitants [2]. As shown in figure [1], we can illustrate the increasing of the total numbers of internet users by the blue area that reach the limit of 4574 million during the recent years. This growth suggests a rise of the interconnection of huge data and services which becomes a main concern as well as it presents as a real target to the attackers in the cyberspace by means of malicious activities known as cyber-attacks.

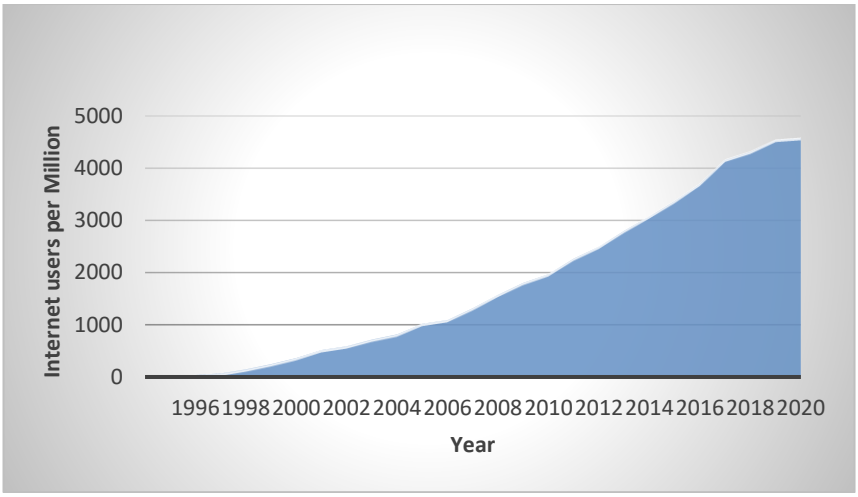


Figure 1: The distribution of the internet users over the last decades

From the modern threats landscape, the networks infrastructure has been a highly significant cyber security interest over the last two decades. The massive growth of the internet users was the main provider of services that offers important benefits such as e-commerce, online banking as well as the Internet of things revolution (IOT). This trend of these services has attracted many cyber-attackers to develop programs that may lead to the existence of malicious activities including intentional ones by stealing sensitive information from computer systems, networks infrastructure and data storage. The cyber security is one of the most important concerns to be observed in which the role of cyber security field is represented in protecting the electronic data against cyber-attacks that has become even more dangerous for the national security and economic stability [3].

In 2017, the Australian Cyber Security Centre (ACSC) critically examined several levels of threats produced by the attackers to penetrate into the network infrastructure [4]. In most cases of electronic piracy and cyber threats, the attackers are usually considered as socially isolated individuals and this tendency will motivate them to perform those illegal activities. They are talented and motivated to carry out malicious actions such as looking for money and other innovative attacks. Moreover, according to the Symantec 2019 Internet Security Threat Report which relied on 123 million sensors to record thousands of attacks that threat 157 countries every second [5], the distributed denial-of-service (DDoS) attacks are increasing in number and ferocity so that most of those attacks are usually for 30 minutes or less. The report reveals the cybercriminals profits using formjacking and web attacks that use malicious actions over the e-commerce websites in order to steal the credit-card payment details and valuable information concerning the client checkout page. In addition, as shown in the figures [2][3], every day in December as an average there are up to 4800 of websites compromised with FormJacking operations during one month and more than 1.3 million unique web attacks on endpoint machines every day.

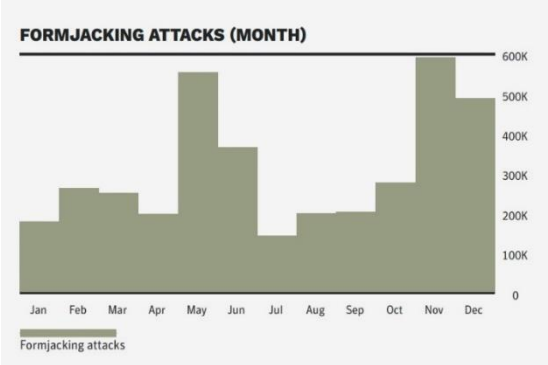


Figure 2: Formjacking attacks per month

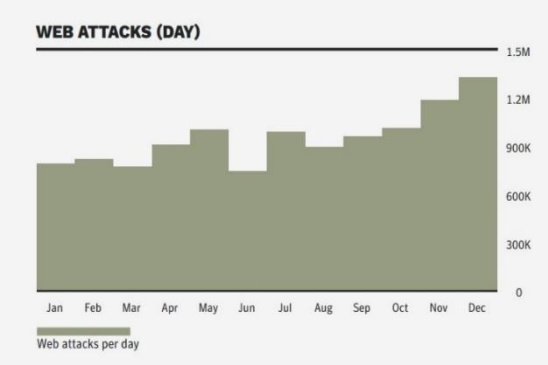


Figure 3: Web attacks per day

Therefore, the cyber-scientists offered several cyber security tools that can be used for detecting, monitoring and preventing those activities such as host and network monitoring, intrusion detection systems and the penetration testing techniques. Indeed, cyber-security mechanism is needed to insure a dynamic protection rule and to hold the cyber space and computer networks to be well secured and configured under to the global safety paradigm. Consequently, the deployment of the security assessment and intelligent systems for intrusion detection based on the machine learning techniques are recommended to achieve this approach.

Objective

The objective of the thesis is to propose a security assessment and a reliable security intelligent system based on the machine learning techniques. Subsequently, our main interest is to find the efficient cyber security mechanisms in order to protect the computer networks and systems. These mechanisms assist the security analysts to evaluate and detect the cyber-attacks and the attackers' behavior. Therefore, a significant part of the thesis is devoted to detect the most important vulnerabilities that may be found in any system, as well as proposing the major solutions to avoid them. Moreover, the thesis emphasizes the web mining techniques to prove the detection of the visitors' behavior and to discover several abnormal actions. Hence, the target is conducted to propose several kinds of intrusion detection systems. We will highlight on the effective methods to detect the most popular attacks that may face the web server and networks such as web attacks, Malicious URLs and Distributed denial of services. Afterwards, we will suggest a learning optimization method to attain the evolving of the systems performance. Finally, the built intelligent model will go through a validation process as an intelligent hardware to prevent the upcoming threats on a real time stage.

Contribution

The contributions of this thesis are stated as follows:

- **Security assessment and enhancement for the cyber security mechanisms:** We employed several penetration testing techniques within the lebanese university to discover the vulnerabilities that may face the web server. Therefore, we respected some of the most important security risks mentioned in the top 10 OWASP. Thus, we deliberated security suggestions to provide the major solutions that assist the IT administrator to protect the systems against cyber-threats.

- **A new approach in detecting the visitor's behavior by applying the cyber security mechanism:** We proposed a new approach by utilizing the web mining techniques. It accomplished the detection of several approaches concerning the visitor's activities, their behaviors and the access resources control. Moreover, we developed a model for the security analysis intent as a rule based detection method validated in the lebanese university. Hence, we constructed a data base that includes more than 100 samples of attack patterns used to detect the behavior related to web server attacks.

- **Demonstration of the efficiency of text mining technique for developing host based intrusion detection system:** We developed a host based intrusion detection system (HIDS) using the text mining technique as an intelligent system. Due to the lack of credible datasets as well as the privacy of data that may be shared among the cyber-researchers, this problematic concern poses a considerable challenge in developing the IDS. Therefore, we provided a reliable data source which contains 6000 records of textual malicious URLs related to the most popular web

server attacks. This kind of data forced us to develop the DOC2VEC model to produce the feature representation method used in learning phase. Afterwards, we validated our HIDS using four machine learning techniques; the multilayer perceptron (MLP) model offered the top accurate classification system in detecting the SQLi, XSS and directory traversal attacks.

- **Validation of the efficiency of an intelligent system based on the machine learning for network based intrusion detection:** We proposed a security intelligent system as an effective network based intrusion detection system called SIS-ID. This system adopted to detect the latest malicious URLs and DDOS attacks. Moreover, we configured our DB-MALCURL and the DB-DDOS which were examined for detecting these kinds of attacks. The system was tested based on several machine learning algorithms. Thus, we deduced that our SIS-ID with the proposed learning optimization technique proved an evolving of the results. Further, the DB-MALCURL based on the ensemble methods attained a better performance than the supervised learning. The voting model recorded the best accuracy (98.52%). On the other hand, our system achieved the top accuracy (77.04%) via the DB-DDOS based on the stacking model. In addition, by comparing the results presented by the CIC laboratory for the detection of both Malicious-URLs and DDOS attacks, our developed SIS-ID proved better performance results.

- **Proving the efficiency of the SIS-ID as an intrusion prevention system:** We validated our proposed SIS-ID system in the Lebanese university as a hardware for intrusion prevention. We tested the system using the local outlier factor model. The model attains the efficiency in avoiding the denial of service attack (DOS) on real time stage.

Report Structure

This report is composed into three major parts.

The first part comprises the state of art of our thesis. It contains the following four chapters:

- The first chapter presents an overview on cyber security domain, the technologies and its importance, the web vulnerabilities concepts, the cyber-crimes and their related activities and the most important cyber-attacks.
- The second chapter introduces a general overview about the web mining methodology and its techniques.
- The third chapter describes the machine learning techniques for building our security intelligent system.
- The fourth chapter outlines the intrusion detection system (IDS), its architecture, the deployment and detection methods. Moreover, a review of the most used open source IDS, a summary of the IDS based on the machine learning technique.

The second part includes the applying of the security assessment mechanisms that will be applied in the Lebanese university, it consists of the following three chapters:

- The first chapter presents our penetration testing technique, the discovered vulnerabilities in several pages concerning the most popular attacks and finally the experimental results.
- The second chapter involves the idea of applying the web mining techniques. Two different tools will be discussed and implemented for detecting the visitors' behavior in terms of their activities and the cyber security approaches. Furthermore, we will present the experimental results for each tool.
- The third chapter comprises the development of a host based intrusion detection (HIDS) system using the text mining techniques. We will construct our dataset then we will propose the architecture of the HIDS. Hence, we develop the features representation technique. Afterwards, we will validate our HIDS based on the machine learning models to offer the best efficient classification system. We will reveal the performance measurements for detecting the web server attacks.

The third part consists of the efficiency validation for an intelligent system based on the machine learning techniques for network based intrusion detection. It includes the following two chapters:

- The first chapter presents the material and the development mechanism to propose our security intelligent system that is called SIS-ID. We offer two configured data bases. The DB-MALCURL and DB-DDOS datasets which will be adopted to detect the latest malicious URLs and the DDOS attacks. Moreover, it indicates the data and features engineering methods as well as the suggested mechanism for applying the ML algorithms and the performance optimization method.
- The second chapter led to prove our SIS-ID using the experimental results. Moreover, we will discuss the contribution of our system in terms of the enhanced performance and results comparing with other approaches. Ultimately, we will validate the efficiency of the SIS-ID as a hardware that is tested in the Lebanese university as an intrusion prevention system on real time stage.

Part 1: State of The Art

Introduction 17

Chapter 1: Cyber Security and Rendered Services 19

1.1 Cyber Security Overview 19

1.2 Cyber Security Domains 20

1.3 Cyber Security Importance 21

1.4 Web Vulnerabilities 22

1.5 Security Technologies 22

1.6 Cyber Crimes 24

1.7 Cyber Attacks 29

Chapter 2: Web Mining Methodology 32

2.1 Web Mining Overview 32

2.2 Web Mining Techniques 32

2.1.1 Web Content Mining 33

2.1.2 Web Structure Mining 34

2.1.3 Web Usage Mining 35

2.3 Web Mining and Security Analytics 37

Chapter 3: Machine Learning Techniques 39

3.1 Machine Learning Overview 39

3.2 Machine Learning Types 39

3.3 Machine Learning Steps 40

3.4 Machine Learning Algorithms 41

3.5 Machine Learning Evaluation Metrics 50

Chapter 4: Intrusion Detection: Concept and Related Works 52

4.1 Intrusion Detection Overview 52

4.2 Intrusion Detection Systems 52

4.2.1 Deployment 53

4.2.2 Detection Methods and Responses 53

4.3 Most Used Open Sources IDSs 54

4.4 Intrusion Detection System based on Machine Learning Techniques 55

4.4.1 Requirements and Materials.....57

Conclusion Part I.....62

References Part I 63

Introduction

Résumé

De nos jours, le cyberespace est considéré comme l'un des domaines de recherche les plus concernés auxquels sont confrontés les chercheurs de la cybersécurité. Dans cette partie, intitulée état de l'art, nous avons introduit les éléments requis à l'avancement de notre thèse. Cette partie est formée de quatre chapitres ; Le premier chapitre consiste à présenter le domaine de la cybersécurité. Il définit les cyber-domaines et leur importance ainsi que les technologies de sécurité qui peuvent être utilisées comme des outils contre les cyber-menaces. De plus, ce chapitre présente les cyberattaques les plus importantes ainsi qu'une revue des actes cybercriminels les plus populaires au cours des dernières années. Le deuxième chapitre présente une revue générale sur le web mining et leurs techniques. Dans ce chapitre, Nous avons montré l'approche de la cybersécurité basée sur le web mining pour détecter le comportement des visiteurs. Dans le troisième chapitre, nous avons introduit la technique d'apprentissage automatique en indiquant les étapes fondamentales de ces techniques intelligentes. Par conséquent, nous avons défini plusieurs modèles de classification et expliqué comment ils peuvent être utilisés dans le développement des modèles intelligents. Le quatrième chapitre comprend le déploiement des systèmes de détection d'intrusion (IDS). Dans ce chapitre, Nous avons discuté l'exigence essentielle des phases de développement, leurs types, les techniques appliquées, l'IDS basé sur les techniques d'apprentissage automatique, et les sources de données les plus utilisées pour développer un IDS intelligent. Enfin, une revue de la littérature sur des travaux déjà réalisés sur l'IDS a été présentée.

Overview

In this part, we will define and introduce all the necessary elements for the progress of our thesis in terms of the state of art. This part contains four chapters: (i) The first chapter consists of overviews on the cyber security field to introduce the cyber-domains and their importance with the fundamental objectives. Moreover, we mentioned the most important security technologies applied in this research domain as well as introducing the cyber-crimes and the related acts in relation to the cyber-attacks and their different types. (ii) The second chapter presents the methods used in the web mining, in particular for the phases concerning the development of a model using the web mining techniques in relation with cyber security approach. (iii) The third chapter consists of introducing the machine learning technique and its types. Thus, we explain the basic steps for the learning techniques with the algorithms which are used in our thesis and the examined evaluation metrics in each experiment. (iv) The fourth chapter includes an overview for the deployment of intrusion detection systems (IDS). Furthermore, we discuss the essential requirement of the IDS construction phases and their latest types and techniques as well as listing the most applied data sources in developing an IDS based on the machine learning techniques with introducing some related works.

Chapter 1: Cyber Security and Rendered Services

1.1 Cyber Security Overview

In the recent years, the use of networks systems and interconnected programs over the internet has witnessed a tremendous expansion and has turned into an essential part in our life and invades for all levels of society. With the rise of cyber security, it has become a basic requirement as well as a necessity to provide a deep threat analysis. Its name comes from two interconnected words; cyber that is related to the corresponding technology which covers the whole networks equipment such as systems and programs, and security that is related to the security methodology which comprises networks security, system and information security [6].

With the growing of cybercrimes and the difficulties of handling large amounts of incoming data to the networks, the state of affairs requires further countermeasures methods to preserve the fundamental protection against cyber-attacks [7]. Thus, the security enhancement becomes more significant due to the ability of cyber-attackers to use several potential ways to penetrate the infrastructure privacy including hardware and software to apply illegal activities and penetration concerning valuable information [8]. Cyber Security enhancement plays an essential role to achieve a continual effort in order to protect the online information and the computing resources from unauthorized use or threat. Furthermore, it is specialized in protecting hosting and networking infrastructure from corruption due to the hacking and attacking activities and malicious software such as viruses, worms and Trojan. One of the first known malicious attacks was the "Creeper worm", which referred to the 1970s by Robert (Bob) Thomas in the BBN Technologies in Cambridge, Massachusetts. Creeper was developed to penetrate DEC PDP-10 computers running over the "TENEX" operating system based on the ARPANET [9]. According to the Australian Cyber Security Centre (ACSC), the center examined various sophistication levels of studies according to the piracy activities in 2017 [10]. Since the threats can be exploited by attackers and lead to cyber-attack, the threats that face the network form a significant problem that is becoming worse and increasing with every passing day. Such threats come from hardware failures, extant vulnerabilities, bugs in programs, malicious code and network threats related to local or remote intrusion.

Cyber security employs a set of technical methods and processes that are designed to protect networks and computer software and data from attack operations. It prevents illegal purposes with unauthorized access for information used to change or destroy the cyber-systems [11]. Cyber security systems consist of several security components, the most common one can be related to network security or local security systems. These devices often contain proposed level of security systems such as firewall, some harmful anti-virus software and intrusion detection systems (IDS). In recent times, these technologies have been powered with the artificial intelligence techniques and machine learning to be reliable to detect infiltration as well as fundamental procedures in networks analysis.

1.2 Cyber Security Domains

Cyber security refers to protection measures that facilitate and strengthen the process of ensuring the latest technologies for cyber security. Thus, the essential requirement was proposed by coordinating it with other barrier defense domains [12][13]. The relationship between the different cyber-domains are stated below and illustrated in figure [4].

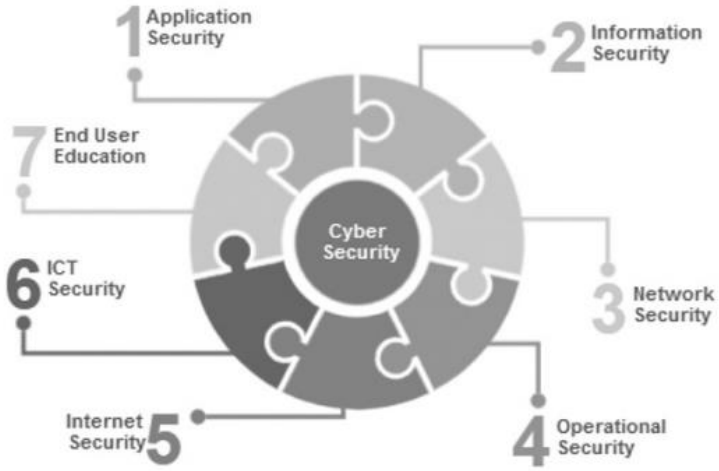


Figure 4: Cyber security and its various domains[12]

- Information Security (IS): it is a strategy with a set of practices that preserve the confidentiality, availability and integrity of the registered data or the information in several domains.
- Network Security: it is considered as a designed process to maintain the usability and integrity of the networks. It ensures the data privacy in order to secure the access of hardware or intelligent detection tools.
- Internet Security: a cyber-field related to the internet environment that involves the browsing security as well as ensuring the World Wide Web (WWW) protocol. Moreover, the essential purpose of providing this domain is to establish and find protective rules and measurement approaches to be used against any attack activities occurring over the systems.
- Application Security: it is used to implement several mensuration approaches to enhance the cyber-security phenomena for desired applications. It can be improved by several methods such as deep system monitoring, attacks detection or prevention and application vulnerabilities solutions.
- End User Knowledge: Currently, the end user knowledge is one of the most cyber concerns. The internet user's weaknesses are the desired gap processes for the attackers that form the threat. The feebleness occurs due to the lack of the user's awareness regarding the cyber-crime risks which can be caused by the misuses of the internet visitor's behavior.
- Operations Security: it is the process of protecting and identifying valuable information which isn't classified and often attacks the attackers to attain vulnerable data point.

1.3 Cyber Security Importance

Cyber security has several approaches with common objectives to protect and maintain information from being robbed or attacked. The three fundamental objectives are: the confidentiality in order to save sensitive information, integrity to provide the data reliability and availability to ensure the information delivery. The confidentiality, integrity, availability (CIA) triad are significant and are considered as the base rules for all security systems to guide the overall policy and data to be completely secured inside the organizations [14]. Therefore, several levels of data protection must be defined in terms of ensuring the CIA triad. The essential tools can be categorized as listed in the table [1] below:

CIA Triad	Tool	Objective
Confidentiality [15][16][17]	Encryption and decryption	Protecting sensitive and valuable data like credit banking numbers and e-commerce transactions.
	Access control	Defining the policies to access systems , physical components and virtual networks resources by giving the users access privileges and permissions.
	Authentication	Giving the permission to confirm the user's identification for any authentication access process.
	Authorization	Allowing the user to have permission in terms of his/her behavior related to restricted security mechanism to Authorized to access system resources using predefined policies.
Integrity[18][19]	Backups	Storing data periodically or automatically in a database management system.
	Checksums	Providing the integrity of transferred data among the networks using mathematical function in order to map the file's content into numerical value.
	Data correcting	Detecting unexpected changes by storing the authentic data and identifying the unoriginal ones among each other.
Availability[20][21]	Physical Protections	Protecting the infrastructure components to transmit their resources to be stored in a secure zone in order to hold the available data.

	Computational Redundancy	Storing Protection devices in case of protection failure against attacks.

Table 1: Most important tools according to the CIA triad

1.4 Web Vulnerabilities

Generally, the term of vulnerability comes from the discovering the weaknesses that attract the cybercriminals to earn unauthorized access or to execute illegal actions on systems. It can be adopted by attackers to inject malicious code, access memory, install worms, send malware, identity theft and access critical data. Web vulnerabilities include software weakness or failure into web based application.

Furthermore, the vulnerabilities that have been discovered in the previous decades in many companies are detected due to the lack of validation or controlling the fields inputs. The misconfiguration of a web server and its systems and components are the main target that may be exploited to attack any system's security. The diversity of those vulnerabilities is present in many web applications because they arise in several cases and in different kinds. Furthermore, the web security vulnerabilities should be detected using predefined plan and can be adopted by respecting the suggested cyber rules listed each year by the OWASP Top 10 which is a cyber-standard awareness document for developers that covers the most critical security risks related to web application security such as input and access validation error; Exceptional Condition Error Handling, Configuration Error, Design Error, Injection, Broken Authentication, Sensitive Data Exposure, Broken Access control, Security misconfigurations, Cross Site Scripting (XSS), Using Components with known vulnerabilities, Insufficient logging and monitoring [22].

1.5 Security Technologies

Cybersecurity has become an interested field for the organizations and companies in the cyber-world. In fact, the data based technological tools take plentiful awareness due to the infiltration of their security privacy. The organizations networks are becoming mightily obtainable and has increased in terms of security concern. Today, the main awareness in the cyber-security field is to exclude the attempts of piracy happening over protective components to beat its privacy. Each organization based on internet services needs a fundamental security strategy such as detecting, analyzing and preventing to figure out the occurred cyber-crimes. Indeed, the most important cyber-security techniques such as the firewall and VPN, intrusion detection system (IDS), IDS based machine learning and intrusion prevention system (IPS) are listed as follow:

i- Firewall and VPN

Firewalls are discovered in 1990s. They offer a solid barrier between the networks components to assure the protection services. Normally, a firewall is constructed on all sides of a network or sub-network to protect it against any threat from outside. Hence, Steven et al. [23] state that the network firewall is a

collection of elements located between inner and outer network. It achieves a cyber-security enhancement by controlling all the passing traffics to precise who's the authorized one to be specified for allowing it to pass among the networks. The firewall checks each coming packet that may blocks the kinds that is'nt recognized via the particular cyber-security criterions. On the other hand, a VPN is a virtual encrypted tunnel located between network and distant server which is employed by the VPNs services. The foreign incoming traffic usually should be routed via secured tunnel using an authentication key to gain access through its server for accessing the desired company resources. Thus, the data will be safe and protected from the attackers. Furthermore, the real user's IP address becomes the VPN server that permits hiding current identity [24]. The homogenous between the firewall and the VPN can be considered as a cyber-security enhancement requirement. Firewalls are the gateways to affirm the safety into the internal network while the VPNs are rules to access that internal network. The VPN is recommended to be implemented where the place of the network firewall assures to secure the network traffic. The lack of the firewall makes the VPN encryption methods worthless. Figure [5] illustrates the presence of the firewall and the VPN while the VPN server is located on the internet ahead of the firewall.

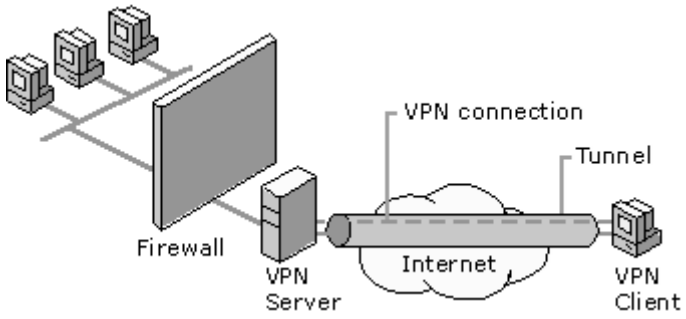


Figure 5: The firewall and the VPN architecture within a network [https://www.cybertroninc.com]

ii- Intrusion Detection System (IDS)

Intrusion is an unauthorized activity that damages the data and information system. It is considered as an illegal activity that poses a direct threat to the CIA of information and therefore has been considered as a harmful interference to networks and systems. Thus, the employment of Intrusion Detection System (IDS) which is a tool that identifies malicious and harmful actions on computer systems and networks contribute in maintaining systems security [25]. It identifies the types of malicious network traffic that couldn't be identified by the traditional firewall over the network. IDS plays an important role in any cyber-security mechanism. It provides a dynamic monitoring through the host and network traffic, and it is necessary to achieve a cyber-protection against attacks. The Intrusion detection system alerts the IT administrator when someone attempts to breach inside the network privacy and the installed firewall gains the access to their components using trusted aspects.

iii- IDS based on Machine Learning Techniques

Intelligent intrusion detection system is one of most concerns in the cyber-researches centers. The need of developing an effective IDS using machine learning has become extremely important to detect the novel and advanced attacks that isn't discovered by the traditional IDS and firewall. In addition, the IDS scrutinizes the desired network to avoid the penetration of valuable information that the attackers focus on during that sneak. For instance, it derives a real time monitoring process as well as incident management to provide a security barrier related to events collected from networks, security devices, systems and applications. Nevertheless, it provides a workflow that tracks and escalates the incident. Moreover, the IDS could be used in different manners such as network or host based intrusion detection systems, log management as well as for generating analysis reports for investigation purpose.

iv- Intrusion Prevention System (IPS)

The growing of interconnected network has become an increasingly critical and dangerous issue. Previously, the conventional type of intrusion systems, firewall and other types of protection tools have mostly been enough to treat the traditional cyber-technology. However, the need of developing a novel approach of protection is deemed as one of the most essential concerns of security researchers by putting their efforts to develop an intrusion prevention system. It is considered to be an integration between an IDS, firewalls and another kind of barrier protection [26]. Generally, The IPS functionality is the same of IDS but it takes a preventative conduct. Moreover, comparing to IDS, IPS can be categorized into two main types; the Host-Based Intrusion Prevention System (HIPS) and the Network-Based Intrusion Prevention System (NIPS). Moreover, Guan Xin et al. in [27] prove that the IPS works as an attack-prevention model and the attacks can be avoided through the studied data traffic. Thus, IPS helps in protecting the network against coming attacks on a real time stage. Furthermore, Salah et al. [28] stated that the network intrusion prevention system can also provide the detection for coming traffic by providing a barrier against the abnormal activities and the HIPS installation over selected machine which achieves a protection method that opposes any suspicious attack.

1.6 Cyber Crimes

Cyber denotes to anything that can be made on the internet network. Crime is the illegal effort that occurs without authorization and implements immoral actions to gain the access of protected cyber-data. Cybercrime is a type of criminal activity that takes place in the bastion of cyberspace. Piracy is a type of cyber-crime and is outdated and known since the 1960s. It steals important information to exploit security system and security vulnerabilities such as finance and identities as well as violating privacy and fraud. The Identity Theft Resource Center announced that more than 170 million personal records were discovered, causing 780 cases of data security system penetration in 2015 [29].

According to the Vulnerability Statistics Report in 2020 [30], there are more than 60% of researchers and technicians working in the field of security systems in their institutions. They exert efforts in checking false positives' rate by discovering the reported vulnerabilities that can take more than 3 hours per day. Further, according to the 2019 online survey extracted from Info security Europe, it turns out that it is not surprising that the current lack of cybersecurity skills is facing security officials who have been responding to this study. It has been shown that only 32% are working with their enormous potential, and 68% remain, as the factors requiring a greater number of specialists that have emerged to manage the security of their cyber data so that they work more strongly with vulnerabilities cases. The report proved the percentage of the total gaps as they analyzed and linked the intensity of risks applied for institutions to cover both small and medium companies and reach them. Moreover, it turns out that the small companies with 11 to 100 employees present the average of risks percentage of all gaps which is 4.1% because such organizations do not have a large digital space to be considered as lightly under attack. In addition, for large organizations the risks are similar in proportions to large extent. For example, organizations with more than 100 employees can have a similar risk density. Moreover, figure [6] presents that there are 14% of vulnerabilities classified as high or critical risk for companies that cover from 101 to 1000 employees. However, 11.5 % of vulnerabilities are classified as high or critical risks for companies that include from 1001 to 10000 employees. On the other hand, 11 % of vulnerabilities are classified as high or critical risk for the companies of 10000+ employees.

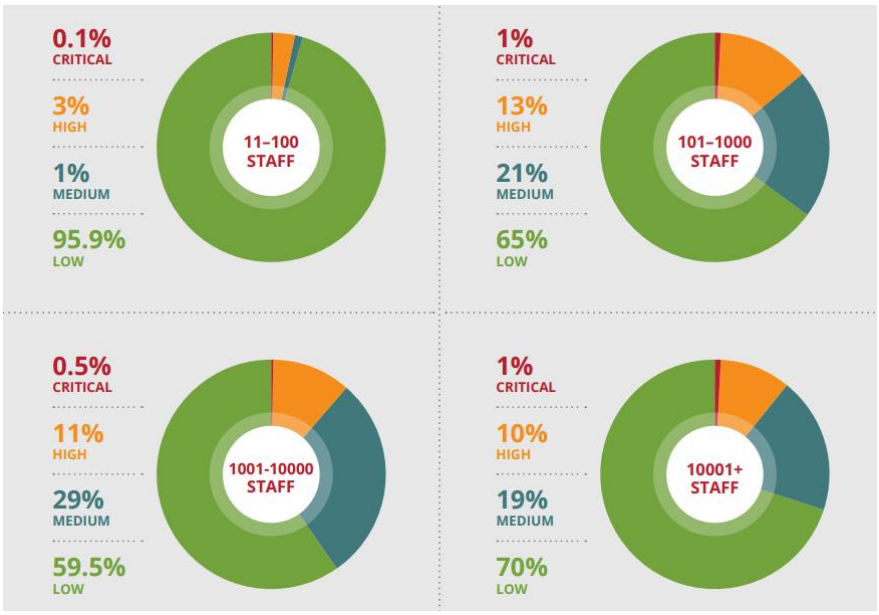


Figure 6: The distribution of risks percentage over the companies' types

1.6.1 Most Popular Acts of Cybercrimes

As a matter of fact, each country has its various rules to combat cyber-crime that prohibits stealing the data center, hacking and fraud. Cyber criminals are motivated in several approaches including financial profits, emotional insecurity, societal criterion, lack of enactment and deficiency of cyber-punishments

[31]. Moreover, cyber-crimes in companies have risen in terms of frequency and severity over the past decades. The cyber-crimes affected billions and more users in many sectors and they caused the companies to pay more than hundreds of millions of dollars. In the last decades, prominent types of cyber-crimes have been detected such as the malicious threat, identity theft and fraud, hacking, social engineering and ransomware.

In August 2003, a malicious threat which is the Sobig.F worm infected millions of computers during the worst aspect of the infection, the worm has appeared in each seventeen emails that were sent [32]. Moreover, according to the Statista Research Department, a study presented in 2016 revealed that in France almost 40% of respondents were under identity theft attack via their social media profile using the social engineering cybercriminals [33]. Furthermore, according to the Crime Complaint Center (IC3) study in 2019 the report shows that the IC3 received 2,047 complaints which identified as ransomware and 8.9 million $ were lost due to the ransomware attacks [34]. Table [2] shows a list of the biggest corporate cyber-crimes in the last years, however, figure [7] shows the top 20 international victims with their case numbers for each country in 2019.

Company	**Year**	**Attack Description**	**Reference**
The Logic Bomb	1982	One the most destructive cyber-attacks in the history of cybercrimes. It occurred by embedding a part of malicious code during the Cold War in 1982. After activating this code in order to control the gas pipeline in Siberia, this malicious code caused a powerful explosion as it was clearly seen in the space.	IFFLAB[35]
Citibank	1995	The Citibank was exposed to a lot of cyber-threats especially in 1995 where the hacker stole the bank with illicit activity to transfer about 3.7 million of dollars into his accounts.	IFFLAB[35]
Hannaford Bros	2008	In 2008, Hannaford which is a supermarket chain with many stores fell victim under a cyber-attack that attained more than 4 million of credit card .It leads to more than 1800 fraud cases which cost the chain about 250 million dollars.	IFFLAB[35]
Linkedin	2012	In 2012, Linkedin suffered from a cyber-crime that led to 6.5 million of hacking accounts.	IEEE[36]
Yahoo	2013	It is a historic data violation that occurred in 2013 which affected more than 3 billion people .Yahoo proved that it paid $35 million in the recent years to solve charges concerning this hacking activity.	CNN[37]

Target	2013	110 million customers were exposed during 2013. Individual and financial information were breached.	IEEE[36]
JPMorgan	2014	In JPMorgan important data about millions of bank accounts were stolen from JPMorgan Chase servers.	IEEE[36]
Home Depot	2014	A payment of 179 million dollars as settlement was paid after stealing valuable data related to emails and credit cards concerning 50 million customers.	IEEE[36]
Sonny	2014	It is well-known as one of the latest breaching activities which caused a havoc as well privacy violation at the Pictures Entertainment while releasing emails and confidential data from their studios.	IEEE[36]
Hilton Hotels	2015	It was the theft of critical customer's credit card data that occurred at Hilton branches across the country when the hackers accessed the Shay's payment system.	IEEE[36]
Law Firms	2015	This cyber-attack destroyed several email accounts at both firms, Cravath Swaine & Moore and Weil Gotshal & Manges. Thus, they achieved about 4 million dollars when the information trading was leaked about the upcoming corporate mergers.	IEEE[36]
Adult FriendFinder	2016	Swinger website states in 2016 that there was more than 412 million of Adult FriendFinder users who lost their private individual information.	CNN[37]
Linkedin	2016	In 2016, Linkedin suffered from a cyber-crime as 117 million accounts were hacked.	IEEE[36]
Swift	2016	The Vulnerabilities in the SWIFT payment system have been exploited by attackers while 81 million dollars have been stolen from Bangladesh Central Bank account.	IEEE[36]
Equifax	2017	Equifax (EFX) reported in 2017 that the company was compromised for more than 143 million people. The company paid over 700 million dollars in order to solve the investigation regarding that incident. Moreover, the breach discovered important personal information related to credit histories about the Americans	CNN[37]

		considered as critical information to financial sectors such as banks and credit card companies.	
Chipotle	2017	The phishing attack employed to attain the information concerning credit cards was related to millions of the restaurant chain customers.	IEEE[36]
Ransomware WannaCry	2017	The United Kingdom was under one of most elusive cyber-attacks called the Ransomware WannaCry threat that faced 300,000 computers through 150 countries. It was sent as a malicious attachment via emails. The attack closed all the files in a manner to demand a ransom in order to unlock them.	IFFLAB[35]
Marriott	2018	This attack occurred when an attacker obtained an unauthorized access to the Marriott reservations system. About 500 million customer's information related to their names, bank cards and passports were stolen. This breach produced a fine of 124 million dollars due to the protection fail for the personal customers data.	CNN[37]
Facebook	2019	According to the report made from UpGuardIn, security-researchers discovered large data gathering among Facebook users. They were publicly appearing on the cloud of amazon servers to be downloaded by the public without any permission. This attack produced a fine of 5 billion dollars due to the lost control over large troves of private users data.	CNN[37]
Capital One	2019	In 2019 the Capital One (COF) server has been hacked by an attacker. The breach delivered 140,000 Social Security numbers as well one million Canadian Insurance numbers, 80000 accounts related to banking sector and unknown number of personal information such as names, addresses, credit scores and several information.	CNN[37]
Mitsubishi	2020	Mitsubishi states that an illegal group attacked the Mitsubishi company via huge cyber-attack by extracting valuable information of 8,000 individuals as well as worthy data for partnering businesses. Moreover, the breach compromised sensitive government agencies which contained secret projects about protection equipment .	CSIS[38]

Table 2: List of the biggest corporate cyber-crimes in the last years

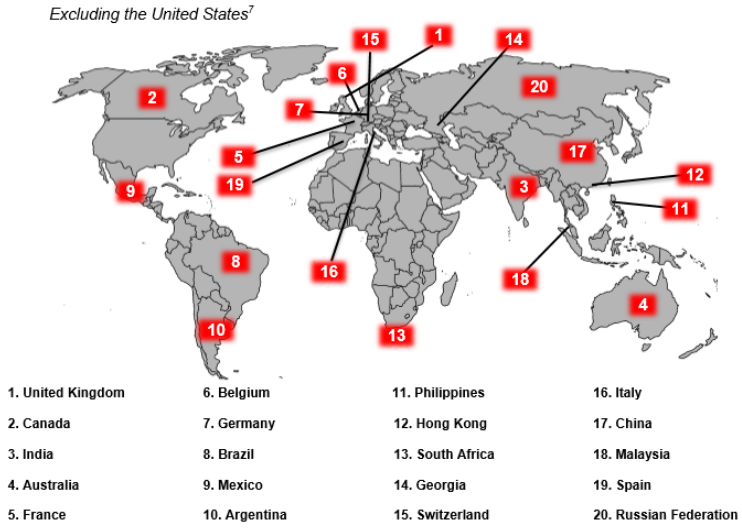


Figure 7: Top 20 international victims' countries [29].

1.7 Cyber Attacks

The cyberattack had a popular target in destroying the availability of a network infrastructure. It denotes to attempt the penetrating of their privacy. It has the potential to harm or destroy the system confidentiality using predefined plan opposed the network privacy. Currently, the newfangled cyber-attacks are presented as blended attacks that uses abundant attacks to infiltrate desired system. In addition, the internet security threat report assumes that the small companies are more probably to be hacked using common attack techniques such as receiving email threats, including spam, malware and phishing email which can likely be more powerful violating smaller organizations than larger ones. Moreover, the report estimates that the spam attack of continued to increase along with 55 percent in 2018 compared with 2015. There is an average of 10,573 malicious mobile apps presented within one day that have been blocked. Therefore, the act of blocking was dividing into 39 % for tools, 15% to lifestyle and 7 % for Entertainment as the most common categories that remained to be malicious applications [39]. Commonly and according to [40], cyber-attack is classified into 2 fundamental categories which are the passive attack which consists of listening without altering the data in a system or network payload. They are generally undetectable while the prevention is possible. The active attack is while the intruder tries to bypass secured system to infiltrate it using malicious activities such as hacking, malwares and denial of services. The most common cyber-attacks are stated and introduced [41]:

i. Phishing

Phishing is a type of prohibited activity that makes the attackers be fraudsters for legitimate action. For instance, they use spoofing e-mail to entrap by extracting the secret data such as passwords and credit card information. A huge number of e-mails are being spoofed by attackers after they try to carry

vulnerable links related to fake websites. The phishing attacks are employed to attain stolen information while the users think that this information is real after they enter their personal information.

ii. Malware

Malware can be referred to malicious tools. It includes many kinds of cyber-crime such as ransomware, viruses, spyware and worms. Malware has a main objective; it breaches the network infrastructure after discovering the system vulnerabilities. It can be presented when the victim clicks on an unknown risky link via received untrusted email attachment that can install harmful and malicious software. By inserting this malicious software into the system, malware may come inside the computer networks and can secretly extract the data by storing it in a hard drive (spyware). In addition, it damages particular important components and makes it impracticable.

iii. Spamming

Email spamming is group of unasked magnitude sending messages from uncommon companies and associations. It can be carried out by sending to huge mass of users to deal with attractive promos and advertisements that permit to fool up the visitors.

iv. Distributed Denial-of-Service

A denial-of-service can flood the networks systems or servers with traffic to drain out the bandwidth and affect the resources. Thus, the Distributed Denial of Service (DDOS) temporarily disconnects the networks or servers that are successfully running. The systems through those networks will be completely offline and will take a long time to respond. The attacker launches a flood of payload to the victim to compromise particular features to interrupt the network availability to be unavailable for users.

v. SQL Injection

Structured Query Language Injection (SQL injection) is an attack in which malicious code should be inserted into a web server. It normally uses malicious SQL statement passed in a vulnerable entry box within a web application. The injection will be executed in order to force the victim to expose extracted data while normally it would not. Scanty of input data validation and inappropriate form of unsafety SQL statement in a web system application can expose them to those kinds of attack.

vi. DNS Tunneling

DNS Tunneling is an attack within a network inside the DNS queries and the responses. A protocol data will be encoded to send the HTTP and other protocols over the DNS. Predominantly, the DNS tunneling comprises the payloads that can be transmitted to a DNS server in a way that the attacker can access the distant server components. Practically, this attack requires external network and should be connected with the attacker spot to access the inner DNS server within the victim's network. It helps the cybercriminals to

add malicious attacks inside the DNS queries in order to create a secret illegal connection tunnel that penetrates into the installed security tools.

vii. Cross-Site Scripting (XSS) Attack

Cross-site scripting (XSS) is classified as a client-side attack. It injects a payload with malicious JavaScript code into the website which will be executed and run on the victim's browser. Specifically, the victim will request the vulnerable page that contains the injected code. The attacker will transmit the pernicious page with the injected payload to the target to execute that illegal action for altering the web page body [42].

viii. Path Traversal

A string path denotes the address location of a system file directory. Therefore, the path traversal attack is considered as a threat process in order to access the stored files or directories which are stored out of main web root. The variables can manipulate the reference to selected files with the dot-dot-slash (../) series and its diversity through ultimate file paths. It can access the random component and the directories files registered among the system. Therefore, it achieves the main source code or the application configuration as well as critical system files. This attack is known with several names like the "dot-dot-slash", "directory climbing" and "backtracking [43].

Chapter 2: Web Mining Methodology

2.1 Web Mining Overview

Day after day, data is massively growing with the daily internet usage. Therefore, the importance of treating this enormous data is considered as a real challenge to determine the needed information for every user to meet their satisfaction. Therefore, analyzing as well as extracting appropriate data from huge databases require a serious necessity with applying robotic programed techniques. This is needed since it is impossible to apply manual ways to perform the accurate fetching of user's data queries from endless pages over the internet to determine important and appropriate information. Moreover, Internet users used to search using world wild web (www) via diverse search engines such as Google, Bing, DuckDuckGo, and Yahoo to attain their concerns of desired data. Statista organization, reported that the internet active people until October 2020 have increased to 4.66 billion users and they include 59 % of the universal population [44].

Data mining known as data analysis and discovery algorithm emerges from the knowledge discovery aimed at searching for a pattern from huge volume of data. It employs diverse patterns, intelligent approaches and tools to transform the raw data into useful information. Using these methods, the enterprises can gain business advantages to explore customer activities and forecasting upcoming expectations as well as developing the marketing plans to rise their revenues and advertising policies [45]. Data mining includes a subset branch called web mining which is the action of mining useful data presented on internet pages. Every Internet user selects several search engines to find their desired information over the internet. This beneficial data that the user is looking-for is accomplished and recognized using mining technique named Web Mining. Many diverse tools can be employed to extract data from internet web pages containing documents, links, information etc… [46]. It is obvious that web mining is quickly seen to be truly significant since the documents that include texts are existing more and more through the internet. However, in case we want to handle relevant patterns, knowledge and desired data manually, it will be a difficult and complicated process that requires much time to gain beneficial information to have significant data since the stored data isn't simply understood like a plain text; data can be in an unstructured form such as multimedia, tags and blogs. Therefore, Structure (page-hyperlinks), Usage (stored data, visited resources,), Content (text, pages content) are desired resources which can be collected via Web mining for mining purposes [45][47].

2.2 Web Mining Techniques

Web Mining is comprised of dynamic and enormous data and mostly with unstructured form that gives the ability for working with huge volume of data to be analyzed. The interconnection of networks prompts the interest of studying discovering of relevant data as well as the issues that face the network concerning the

user behavior. In order to tackle this kind of issues, hard work is needed for offering pertinent data in existence form that is straightforward and valuable for associations to foresee client's requirements [48].

Web mining can be categorized into three groups named as web content mining, web structure mining and web usage mining based on the knowledge extracting [49], listed and clarified in the following figure [8] [45].

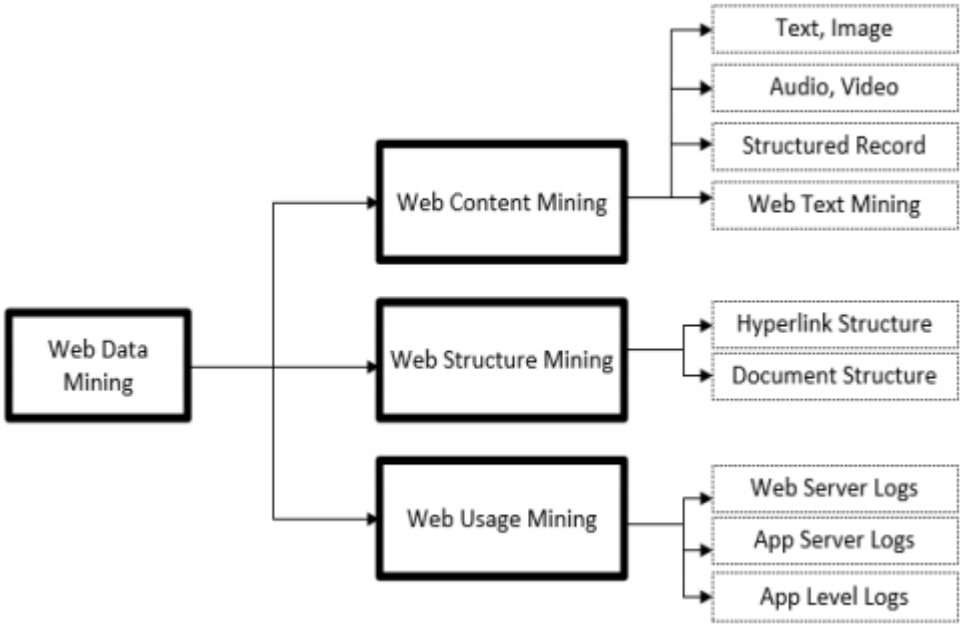


Figure 8: Web mining methodology and its techniques [45]

2.1.1 Web Content Mining

Web Content Mining is a practice of web mining technique that requires useful data mined from the World Wide Web (WWW). The content consists of audial and visual data, text, and hyperlinks. Web contents intend to convey data to internet users in several forms. Over most recent couple of decades the quantity of web pages has risen to a huge amount to reach billions and still keeps on increasing [50].

Searching query in billions of web pages is seen to be problematic and of time tedious errand. Thus, the content mining retrieves queried data by executing diverse mining methods to access valuable data which comes to be simple in discovering essential handler data using four techniques presented in figure [9].

Generally, the data appears in the web contents in several forms. The unstructured form needs a specialized mining technique such as text mining. For instance, the existing text into documents are correlated to the mining technique in order to extract prior information from its content source. The structured techniques are methods employed in order to mine the structured data form. Structured data mining is considered as critical method since it signifies the host page to attain inaccessible information. Therefore, unlike the unstructured pattern, the structured type can be determined to extricate the desired

data [51]. In addition, the semi-Structured cannot be entirely considered as a kind of structured data and mostly it can be as a related shape. The desired text in semi-structured data can be grammatical as well as its structure like Hierarchical, and not predefined. The representation would be in frame of tags such as ODBC data source, Extensible Markup Language (XML), Hypertext Markup Language (HTML) [52]. Conventionally, computing data is usually presented as text and numbers, but presently there are distinctive computing related to multimedia data such as images, videos and audios, etc…[53].

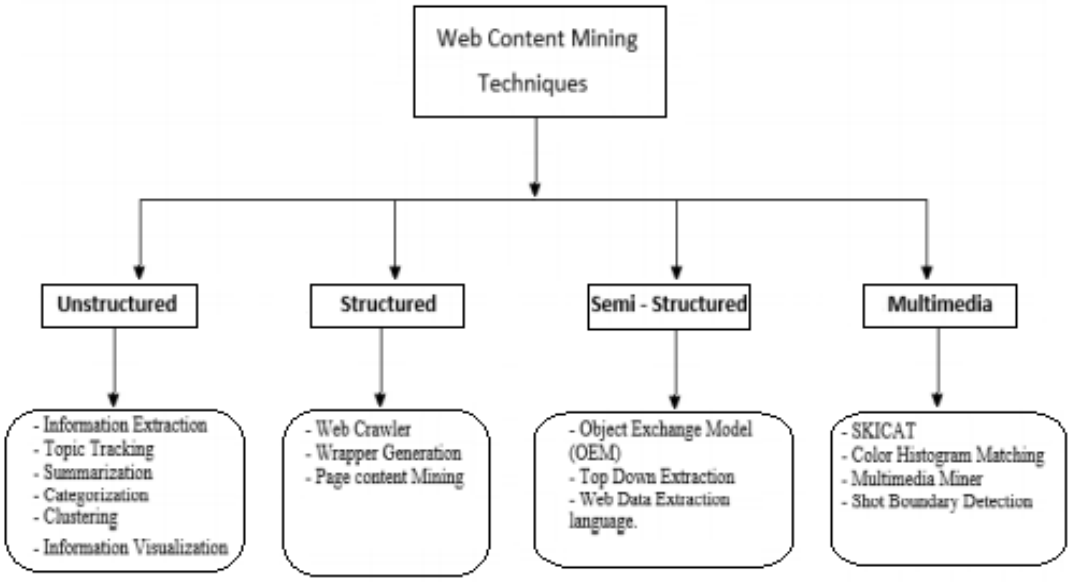


Figure 9: Web content mining structure [45]

2.1.2 Web Structure Mining

Recently a huge volume of data is growing over the internet resources. World Wide Web is the favorite beneficial resource deployed for information retrieval. Structure mining is presented as main primary method of web mining which is convenient with hyperlinks form. Web Structure Mining mainly reveals the structured instance of the website. It recognizes the hyperlinks topology of the websites. The structure of the WWW can be visualized as schema where the Web pages can be considered as nodes and the hyperlinks are boundaries related to the connected pages. It is a useful process in discovering information structure over the cyber-web [54].

Moreover, Web structure mining is divided into two essential parts coveted to be used which are the Hyperlinks and Document structure.

A Hyperlink is a structural component that links a predefined location in Web page to either diverse position on the same or different Web page. Furthermore, The Hyperlink presented in Intra -Document should be

pointed to a defined place inside the current Web page. On the other hand, the Inter-Document connects the links within two different Web pages. There are numbers of suggested techniques for the extracted links via analysis methods.

Brin and Page, in 1988, proposed one of the most important algorithms which is PageRank method as well as the Weighted PageRank (WPR) offered by Xing and Ghorbani in 2004, and, finally, the Hypertext Induced Topic Search (HITS) indicated by Kleinberg in 1999 [55][56].

The Document structure is an organized web page displayed like a tree-structured format based on the varied HTML and XML tags which appears on the web pages. Thus, Moh, Lim, and Ng employed the DTD-Miner algorithm to automatically extract the document object model (DOM) structures within the documents.

2.1.3 Web Usage Mining

Web Usage mining is the strategy of applying web mining techniques to discover and analyze in real time clickstreams usage patterns and related data generated as a result of user interactions with one or multiple web sites. Specifically, web usage mining is the process of grabbing and extracting valuable information in order to find patterns related to a user's behavior of a specific web based system that can determine: who they are, and what they tend to do. Web usage mining techniques consists of the following sections: pre-processing, pattern discovery, and pattern analysis [57].

When a user requests specific and particular resources of web server, each request will be recorded and stored in a web log. This record is referring to the browsing behavior for a user. In Web Usage Mining, data can be collected from multiple resources such as: files (image, sound, video and web files), operational databases and server log files that can include web server access logs and application server logs. On the other hand, the collected data in the web log file will be in an unstructured format and it can't be used directly for mining purposes as many techniques should be applied on it. The Pre-processing technique plays the role of converting the data into a suitable and an organized form that can help to precise the pattern discovery and to provide accurate, appropriate and summarized information for data mining intent. Data pre-processing, includes data cleaning, user identification, user sessions identification, path completion and data integration. Pattern discovery benefits from the preprocessing results in order to offer some techniques such as statistical analysis, sequential pattern analysis, association rules, clustering and other techniques. The pattern analysis should be executed and performed by the following techniques: visualization techniques, OLAP techniques and usability analysis [57].

Aside from detecting the visitors' activities and their behavior, web usage mining can be effectively used to detect existing weaknesses on the web server components and analyzing audit results for anomalous patterns detection. In addition, one of the main functionalities of the web usage mining is to analyze collected data from the web server access logs, browser logs, proxy server logs, registration data, user

profiles and sessions, user queries, cookies, mouse clicks and scrolls, bookmark data and other detailed data as interaction results.

The web usage mining technique can be declared by a three-step process: data pre-processing, pattern discovery and pattern analysis listed as shown in figure [10].

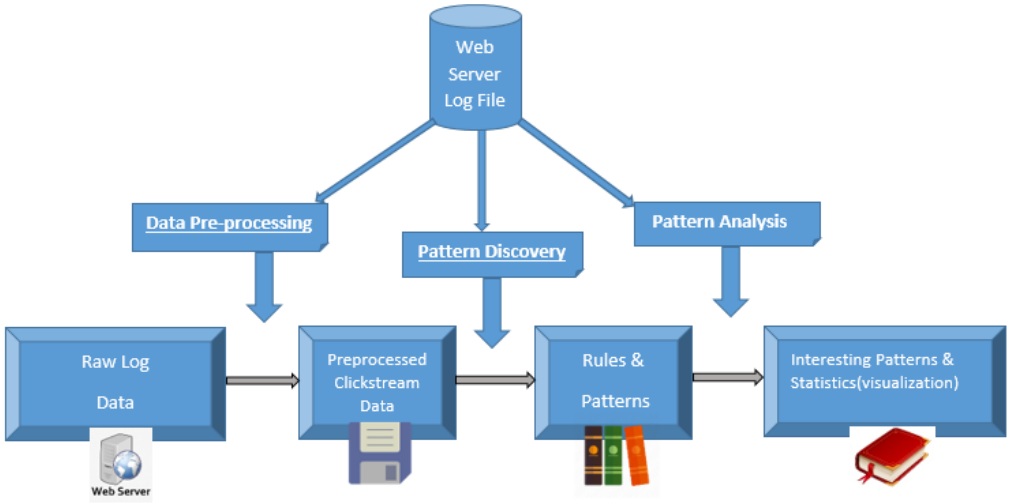


Figure 10: Web usage mining process

i. Data Preprocessing

By accessing any website, the user's behaviors will be stored in the web server log file in an unclear and unorganized form. As a definition, data preprocessing is the process for converting the raw data presented in log files into suitable forms such as data base and different logging methods that contribute effectively when applying the data mining algorithm [58]. The main log file cannot be directly used in the web usage mining process due to the large amount of irrelevant entries in the log file and difficulties and many reasons. Hence, web log file preprocessing becomes essential and significant. Nowadays, many research centers are interested in data preprocessing of Web Usage Mining methodology.

Thus, data preprocessing plays an essential role in increasing the mining accuracy to improve the data quality for further usage.

ii. Pattern Discovery

Pattern discovery employs the preprocessing results to offer some techniques such as statistical analysis, association rules, sequential pattern analysis, dependency modeling, classification and clustering to capture useful and beneficial information [59]. The results that are grabbed can be represented and employed in several ways such as graphs, charts and tables, etc. As an example the visitor's location can be specified using his own IP address. Therefore, by analyzing the discovered patterns related to the web visitors, the web server administrator can discover and detect the most active countries who are visiting a certain website or any web page that can provide the useful information relevant to the specific country [57].

iii. Pattern Analysis

Pattern analysis can be classified as the final step in the Web Usage Mining process. The main purpose of applying the pattern analysis is to filter out the unusable and the non-beneficial rules and patterns from the set that has been found in the pattern discovery phase. Most Pattern analysis techniques are used to attain the above mentioned purpose [60].

One of the above techniques is the knowledge query mechanism like SQL which is a standard language for storing, retrieving and manipulating data in databases [61]. Another method is called (OLAP) which is an operation to load usage data into a data cube in order to perform Online Analytical Processing. Moreover, Visualization techniques is the process of conveying information in a way that the information can be quickly and easily digested by the viewer or the analyzer such as graphing patterns by assigning colors to a specific value to highlight overall patterns in the data. Content and structure information is used to extract patterns that contain several pages of a certain usage type that can match a certain hyperlink structure.

2.3 Web Mining and Security Analytics

The internet provides actions with useful platforms to build consciousness brand and links with actual customers. Thus, a web over internet also makes E-businesses prone to illegal cyber activities and steals valuable personal information. Cyber field is predominantly difficult to be secured for several reasons which include the ability of malicious actions made by attackers to unknown control within the cyber space, the interconnection of physical as well the digital systems for companies, and also the ever-changing kinds of networks [62]. Moreover, Web traffic can be a worthy data center and usually examined in the E-commerce platform to track trademark sensibility and advertisement efficiency. Therefore, the term of web traffic is a sensitive target and plentiful source in the cyber-security field. To catch the threat of malicious cyber vigor, a monitoring and evaluating stage must be applied in the server components. Matomo, Deep Log Analyzer, Woopr and Google Analytics are frameworks employed to analyze the retrieval data. They collect the occurred web traffic to detect the user's behavioral analysis [63].

According to Verizon's 2020 Data Breach Investigations Report [64], the cybercriminal can take hours, minutes or even seconds to steal sensitive information. In addition, the report shows the recording of 157,525 cyber-incidents and 3,2002 of them met its standard quality, and 3,950 for data privacy violations and breaches were confirmed. Therefore, the integration of several tools and algorithms and analytic operations is particularly applied to detect prospect threats. Thus, the cyber-analytics requirement currently is growing due to the swift progresses in several cyber-crimes and exploits. It can be deployed as an approach to produce analytics studies related to proactive cyber security measurements. For instance, Traffic monitoring can be used to detect indicators of threat before it occurs.

2.3.1 Benefits of security analytics and use cases

Security Diagnostic analytics are mostly linked to data discovery process using several mining techniques and detection processes. Often, data is efficiently examined for detecting visitor's behavior. It assists in identifying the indicators which can affect directly or indirectly the desired matter [65]. Moreover, Security analytics could be performed for vast assortment of case studies. The security administrator uses several methods for analyzing network traffic. It permits to detect vulnerable patterns that assist to reveal potential threat. Furthermore, the user's behavior monitoring can predominantly detect the suspicious, illegal behavior and insider threats [66]. The security analytics benefit is related to the growing volume of information to be analyzed. The most important useful data can be but never limited to:

- Visitor's behavior data
- Contextual data
- Endpoint data
- Identity and access management data
- Business software data
- External data threat
- Cloud computing data
- Network traffic data (packet, flow)

Chapter 3: Machine Learning Techniques

3.1 Machine Learning Overview

Machine learning (ML) is an application subset from Artificial Intelligence (AI) that offers the systems to automatically learn and develop from the past knowledge without being manually programmed. The history of AI began when McCulloch and Walter Pitts presented the first neural network application in 1943. Subsequent, in1950 Alan Turing offered the next noteworthy act in terms of developing another AI model where the famous question was published about: can machines think? [67]. Machine learning in data mining is usually used for implementing the process of finding and searching for useful and hidden information from large amount of data. Moreover, it can be effective in the development of computer programs that use information having inputs and outputs in order to create an automatic learning function that infers the outputs to extract valuable knowledge based on the inputs data. Mainly, ML techniques are used frequently in several branches such as marketing, health, internet of things, intrusion detection and scientific discovery [68][69].

3.2 Machine Learning Types

Generally, machine learning employs intelligent algorithms during the modeling stage and it is commonly categorized into four main types of classes which are the supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning stated as follow:

i. Supervised Learning

Supervised learning is divided into two main classes; the Classification and Regression. Classification maps the data to some predefined categories. When we examine new data, this process provides a level of accuracy to which categories the data instances belong, and it predicts a discrete label such as the boat and car. On the other hand, regression is mainly used to predict continuous values. Most applications take advantages of regression for purpose of predicating and forecasting, it helps in identifying the behavior of variables under study. Regression forecasts a continuous quantity or value, like the time or weight.

Here are sets of related algorithms for supervised learning such as; Decision Trees Random forests, K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), Naive Bayes and Linear or logistic regression.

ii. Unsupervised Learning

In the unsupervised learning, the input data is not labeled with the output ones, so there is no previously noticed patterns as well given with minimal oversight for humans. As a result, it's a target here to discover the patterns for the available data.

The different methods for the unsupervised learning are listed as follow:

A. Association

This technique mainly searches for the most occurring item set from the large data volume. It is used to identify the relationship between the feature variables and thus indicate the most effective variables of this relationship as an outcome of future values. In enormous databases and through association rules we can generate associations between data objects in large databases to notice the valuable relationships among variables. An example of association rules in the market analysis when the customers frequently buy the items together.

B. Clustering

By using this technique, the data is divided into groups that are not defined with unknown properties which are recognized as cluster. A cluster is a group of instances that describes the data it contains that reveal the natural structure of the studied data. We can say that data instances in the same cluster are more similar to each other than the data contained in any other cluster. Some elements do not belong to any cluster which are known as outlier. We can observe the structure of the outlier when comparing them to data found in other clusters and this reveals the nonstandard system behavior [69]. Also, it encompasses unsupervised learning. The main target is presented in discovering the suitable pattern with a group of unlabeled data. The processing of data takes place in the algorithms for clustering then fetch whether the clusters are present within the data. The cluster's number to be defined can be changed in the algorithms to manipulate the precision of these clusters. The Clustering methods can be summarized by these areas; market segmentation, search result grouping, social network analysis, medical imaging and image segmentation.

iii. Semi-Supervised Learning

Both of labeled and unlabeled data inputs are trained in semi-supervised learning, but the amount of unlabeled data is much more than the labeled one through the training. This combination enhances the learning and accuracy. Therefore, semi-supervised learning is a mixture of supervised and unsupervised learning together.

3.3 Machine Learning Steps

The machine learning mechanism is separated into seven stages described in figure [11] below. The whole first part of the process of Machine Learning is to collect data. It requires three tactics: data discovery, data augmentation, and data generation. Whenever we need to explore new datasets, data discovery is important with the more accessible public datasets over the internet network or research facilities. After the data discovery, data augmentation occurs, as current databases are strengthened by adding further external data. If the existing dataset isn't really adequate and the external data is also not accessible, data

generation may be beneficial, even though there is capable to produce datasets alternatively [70]. The dataset has to be preprocessed after getting it to be capable to feed all to the model. The method of converting raw data to data types consistent with the input of the models is data preparation or data preprocessing. In all machine learning techniques, there is always a need for input data. But in most of ML techniques these input data should be reformatted first before employing it. Certain datasets contain data that is incomplete, unreliable or complicated for the operating of algorithms. Then, in the third step, the selection of the proper model has to be performed to fix the real problem in an accurate and scalable way. A further complex model does not guarantee that the model is better. The purpose of picking the model is for dataset training. In dataset training, we are training the model too to progressively enhance the predictions of the model. The biases and weights continue to change in every iteration. The model is trained with labeled data in supervised machine learning; however, unsupervised machine learning needs to learn through the data that is unlabeled. The model evaluation phase begins after training; in this phase the testing of the ML algorithms takes place using the portion of data that isn't employed before in the training previous step. It reflects the model's success in real world situation. Then the model can be improved by testing the parameters. By raising the training iterations number, more reliable results can be obtained. Further tuning is required till you ensure the most effective model performance efficiency [71]. We can execute the model once all the above steps are completed to find solutions for real life problematic scenarios.



Figure 11: Machine learning steps [https://datafloq.com/]

3.4 Machine Learning Algorithms

In this section, we will list our commonly applied machine learning algorithms that can be used to almost any classification problem.

3.4.1 Decision Tree

The Decision Tree is categorized as one of the supervised learning. It can be applied via two kinds of machine learning types which are the classification and regression that permits to make decisions in scientific and related problems. In addition, the Decision Tree employs an algorithm to predict the concerning outputted label based on trained data source. It must be examined by a feature vector to model it in decision rules that should be learned from their own dataset. This algorithm begins as a tree starting from the top via a root node. The values of the root attribute must be compared with the record attribute. The internal nodes symbolize the features of a subset and the branches representing the decided rules

while each leaf node will represent the predicted results. The Decision Tree uses several algorithms such as: ID3, C4.5, CART, CHAID and MARS [72].

- Decision Tree components:

Root Node: It includes the entire sample divided into several homogeneous sets.

- Splitting: It splits the selected node into two or many sub nodes.
- Decision Node: it occurs when the sub-node will be divided into another sub node.
- Leaf / Terminal Node: it is called like a Leaf or Terminal node when the nodes do not split.
- Pruning: this phase is employed in order to remove the sub-nodes of a tree
- Branch / Sub-Tree: It represents a subsection of the whole tree.
- Parent and Child Node: parent node occurs when the node is severed into sub nodes while the sub nodes are considered the child of a parent node.

There are several impurity measures, we will state the most important of two measurements; the Entropy and Gini index/ Gini impurity.

Entropy is a set of needed information to accurately delineate some instances. For example, if there is a homogeneous instance and the element is identical than Entropy is 0. Otherwise if it is equally split, the entropy should be maximum 1. Mathematically the Entropy can be calculated using the following formula:

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Gini index refers to the mensuration of inequality in instance with value between 0 and 1. While if Gini index is equal to 0, it means that the instance is homogeneous, whereas, if Gini index is of 1, it refers to inequality between elements. mathematically, it is the summation of the square of probabilities for each class and can be calculated using the following formula:

$$Gini\,index = 1 - \sum_{i=1}^{n} p_i^2$$

Where i is number of classes

3.4.2 K-Nearest Neighbors

The method of K-nearest neighbors saves the accessible data inputs and according to the likeness measurement, it classifies the newly fresh records employed for regression and classification complications.

Another name for K-nearest neighbors' method is lazy learner since the learning phase doesn't happen directly from the training set, however it saves the data trained and classifies novel input data according to the matching criteria [73].

Here are the K-NN algorithm stages:

- Step one: Detect the k-neighbors' number
- Step two: Compute Euclidean space among the k-neighbors.
- Step three: Choose K nearby neighbors.
- Step four: Calculate the total data points' number for every category between the k-neighbors.
- Step five: A class is resulted as a prediction outcome with the highest total of the neighbors.

3.4.3 Support Vector Machine (SVM)

Support Vector Machine (known as SVM) is a supervised learning method designed for binary classification "either positive or negative class". It is applied for the purpose of finding patterns from the collection of data. Generally, the pattern classification applies an activity to be involved in two main steps; the first is mapping the input to higher dimension feature space, this is done due to the fact of the SVM that usually depends on geometrics characteristics of inputted data, and, the second is finding the most suite hyper plane that classifies the mapped features in the higher dimensional space [69]. It's beneficial for classification and regression approaches. The margin is the space between the hyperplane and the nearby data point. The core target is discovering the hyperplane with the greatest database split; into 2 classes for gaining novel vectors with proper classification.

The important expressions in Support Vector Machine are:

- Support Vectors: are the Data points that are the nearest to the hyperplane. The Data points determine the Separating line.
- Hyperplane: is the dividing line of the features' sets into various classes.
- Margin: is the hole among 2 lines with the nearby data points for diverse classes. The Margin is the space that dissociates the line from the support vectors. The higher the margins are, the more convenient it is.

In addition, these hyperplane separators (decision boundaries) are defined by the formula:

$$h(x) = x^T\beta + \beta_0 = 0".$$

Where:

x: a vector of features.

β: a vector of coefficients.

β_0: a constant value.

For any point in space, the distance from a point to the hyperplane is given by the formula:

$$d(x) = \frac{|x^T\beta + \beta_0|}{||\beta||}$$

knowing that $||\beta|| = \sqrt{\beta_1^2 + \cdots + \beta_p^2}$ and maximizing the margin amounts to minimizing the norm of the vector of parameters β.

The selected hyperplane is the one that maximizes the margin represented by the distance between the learning points closest to the separator hyperplane: support vectors shown in figure [12].

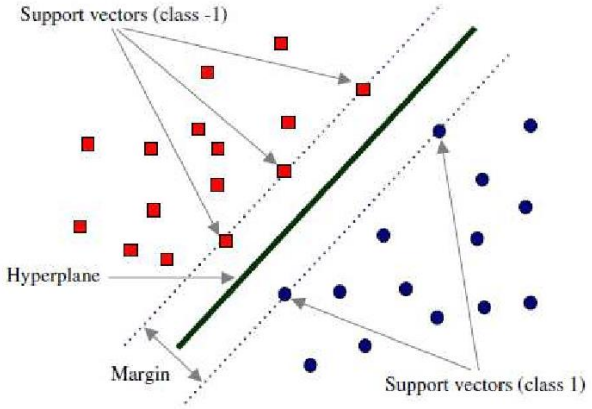


Figure 12: Support vector machine technique [https://medium.com/]

There are several kernel functions (called Kernel) that can be applied to non-linearly separable data:

Linear kernels: $k(x_i, x_j) = x_i . x_j$

Polynomial kernels: $k(x_i, x_j) = (1 + x_i . x_j)^p$

Gaussian kernels: $k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$

Sigmoid kernels: $k(x_i, x_j) = \tanh(\alpha x_i . x_j + \beta)$

3.4.4 Ensemble Techniques

The ensemble methods employ multiple learning algorithms by combining them to produce single and powerful prediction output. Indeed, the applying of an intelligent model using the ensemble methods achieves and produces accurate results than a single model as resulted previously. The usage of ensemble methods can be employed for several classification purposes which perform the model by decreasing the variance as well as decreasing the bias in order to improve prediction of the desired outcome.

i. Random Forest

Random Forest are supervised ensemble learning models used for classification and regression. The Random Forest consists of several decisional trees to classify a new object from an input vector while the random forest's input vector is the input of every tree. The scientist proves that this method achieves an

accurate and stable prediction with high performance. Moreover, contrary to the decision tree the RF, which is shown in figure [13], uses the processes of finding the root node and splitting the feature nodes will run randomly. In the prediction stage, the RF employs the test of several features in order to use the essentials of every randomly generated tree to predict and store the output by selecting the classification rules that gain the maximum votes [74].

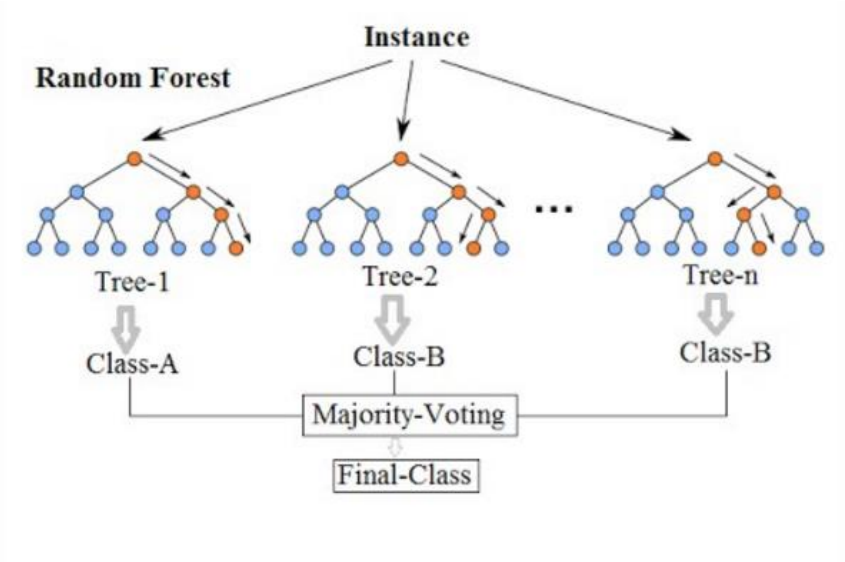


Figure 13: Random forest technique [74]

ii. Boosting

It is considered as an ensemble approach used for machine learning method to make the forecasting of weakened model more efficient by consecutively training every weak learner, so every learner adjusts its ancestor. This method performs several steps listed below and shown in figure [14][75].

- From the main dataset, the subset is generated.
- Similar weights are designated to the whole data input points.
- A base model is formed with this subset.
- This model is utilized to generate forecasts upon the dataset
- Larger weights are provided to the findings which are inappropriately forecasted.
- A further model is developed, and inside the dataset forecasts are produced.
- Likewise, several models are formed where everyone fixes the prior model's mistakes.
- The last model is the weighted mean for the whole models.

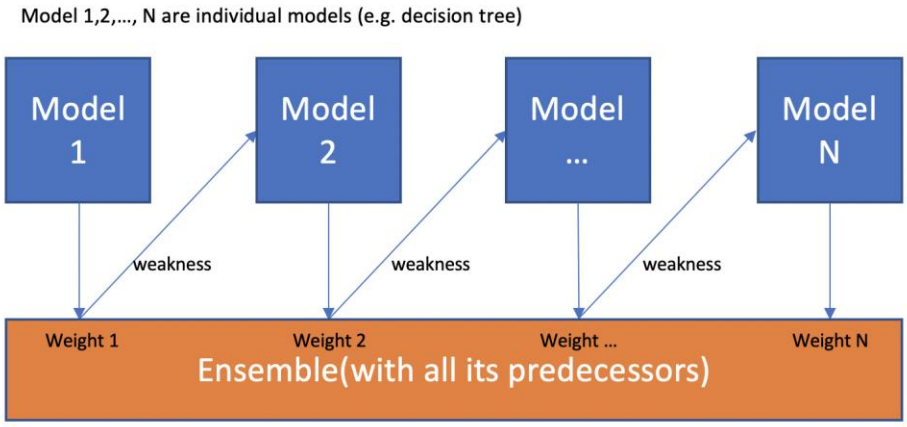


Figure 14: General architecture of the boosting classifier [https://cppsecrets.com/]

There are many boosting algorithms such the AdaBoost, GBM, XGBM, Light GBM, CatBoost and Extreme Gradient Boosting. we will list some of them as mentioned below

A. ADABoost:

Around1996, Yoav Freund and Robert Schapire suggested an Adaptive Boosting that is defined as the ensemble-boosting classifier. It is an iterative ensemble technique for constructing an effective classifier by merging several not strong enough classifiers such as the produced powerful model that can have effective results [75]. Throughout the steps outlined, it performs the following

- Training begins with a random subset of a dataset.
- By choosing the best reliable training data set of the preceding training forecast, it trains in an iterative manner.
- Larger weights are allocated to the experiments with false forecasts, yet these experiments will have a strong chance with classification in the following iteration.
- The allocation of the weight is done per every classifier's precision. When the classifier will be more precise, it will gain greater weight.
- This approach finishes the iterations once the entire training data matches with no mistake or till the defined largest number of estimators is achieved.
- In order to classify, the voting among all algorithms is proceeded.

B. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) also is an ensemble approach formed with decision tree methods which enhances the boost of the gradient, it's useful for regression or classification. Gradient boosting presented in the formula shown below integrates predictors in sequential manner and fixes the past unreliable estimators. From the past forecast outcomes, this approach matches the novel model and reduces the loss by inserting the recent prediction. To optimize model generalization performance, XGBoost utilizes automated regularization. It produces better efficiency comparing to Gradient Boosting, moreover it

is much regularized form of Gradient Boosting. The training is very rapid and might be distributed/paralleled through clusters [76]. There are 2 components of the objective function: training loss and regularization so our requirement is to minify in t iteration:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Real value (label) known from the training data-set

Can be seen as f(x + Δx) where x = $\hat{y_i}^{(t-1)}$

iii. Voting

This technique can be used to solve classification issues. The voting approach employed as an ensemble technique is presented as an example in figure [15]. In order to forecast every input record, several models are often utilized. The forecast for every model is treated as a vote. The last predictions seem to be the output label which gains the minority vote [77].

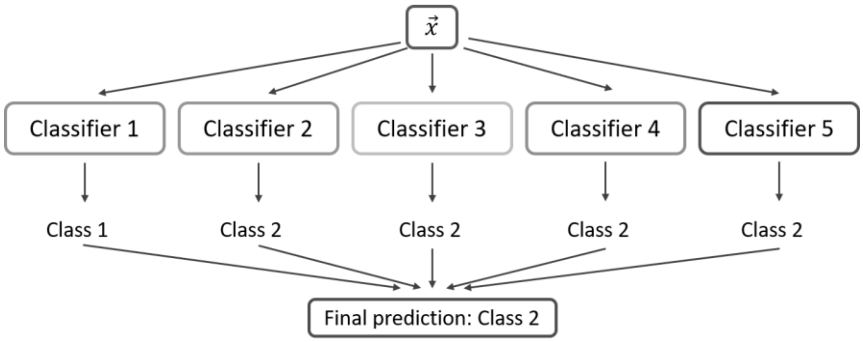


Figure 15: General architecture of the voting classifier [77]

iv. Bagging (Bootstrap Aggregating)

In order to produce a standardized result, bagging incorporates the outputs of many models. Based on the subsets included in the main dataset, several base models are generated and trained. As you can see in figure [16], bagging takes with N samples in recurrence from the training instances with N magnitude in order to train the main classifier and it keeps in repeating until the coveted size of the ensemble is attained. Mainly, the bagging method must be employed via unstable classifiers which are critical to variations within the training phase like the decision trees and others. In addition, the Bagging algorithm can be used via tow estimators, the bagging meta-estimator, or random forest. In order to pick the best forecasting stage, models are executed together and independently then merging the outputs from all the models [77].

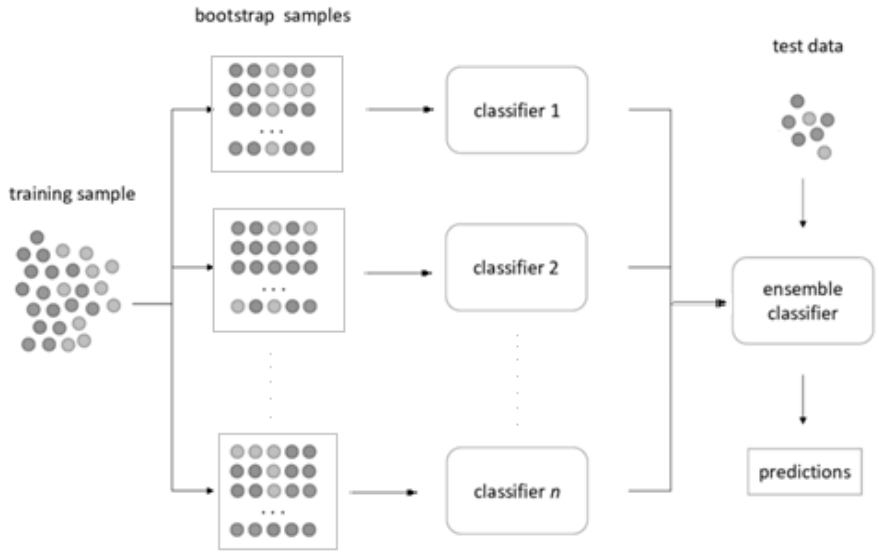


Figure 16: General architecture of the bagging classifier [Nick Minaie, "The Data Scientist's Guide to Selecting Machine Learning Predictive Models in Python, https" ://towardsdatascience.com/]

v. Stacking

Stacking considered as an ensemble technique which creates a novel model using forecasts from various models as shown in figure [17][77].

- The training set is divided into ten components.
- There are nine sections of a base model as well as the forecasts are generated from the tenth section. This is achieved for every subset of the training set.
- On the entire training dataset, the base learner is then adjusted.
- Forecasts are generated on the testing set utilizing this model.
- Again, for the next base model, the second to fourth stages are reapplied, producing in other set of forecasts for the training and testing set.
- By creating a novel model, the forecasts from the training set are employed as features.
- Finally, the model is utilized to finalize the forecasts on the test forecasting set.

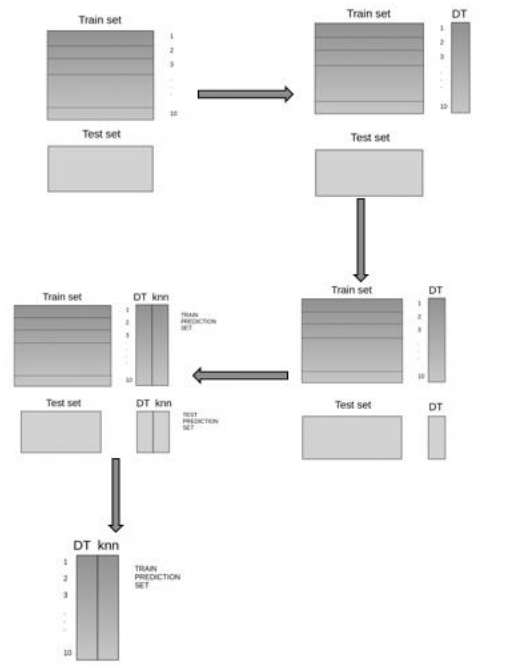


Figure 17: General architecture of the stacking classifier

3.4.5 Local Outlier Factor (LOF)

In the year of 2000, Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander proposed the local outlier factor (LOF) algorithm. In this model, the local density deviation should be measured according to data point with respect to its neighbors. Moreover, it used to detect anomalous outliers in any datasets. Thus, the scalar value is the critical factor to decide the LOF. It should be assigned to each data points with their scores which must be compared to find outliers. Hence, the high value for that LOF have the ability to be an outlier [78].

Mathematically, for a given dataset, the factor can be calculated according to the formula below:

Dn = { (xi , yi)|xi ϵ R2, yi ϵ {X,Y,Z) }

Local outlier factor for each data point is given by the formula shown in figure [18]:

$$LOF(x_i) = \frac{\sum_{x_j \in N(x_i)} lrd(x_j)}{|N(x_i)|} \times \frac{1}{lrd(x_i)}$$

$|N(x_i)|$: Number of elements in the neighborhood of x_i

$lrd(x_i)$: Local Reachability Density of x_i

Figure 18: The local outlier factor formula

$LOF(x_i) < 1$, it means that there is higher density than neighbors which represents the Inlier.

$LOF(x_i) > 1$, it means that there is lower density than neighbors which represents the outlier.

3.5 Machine Learning Evaluation Metrics

In this part, we will discuss the diverse method to examine the performance of the classification techniques. We will debate the evaluation metrics in terms of a binary classification such as confusion matrix, area under curve (AUC - ROC) and the classification measurement coefficient explained as follow:

i. Confusion Matrix

The confusion Matrix represents the full efficiency of the model. It can be used to evaluate the performance of any classification problem by plotting it for the training and testing stages as shown in the table [3] using the rates below:

- True Positive (TP): True Positive corresponds to the percentage of positive instances which are predicted as positive.
- True Negative (TN): True negative corresponds to the proportion of samples that are predicted negative and they are actual negatives.
- False Positive (FP): It corresponds to the probability that an actual positive will be predicted negative.
- False Negative (FN): It corresponds to the probability that an actual negative will be predicted positive.
- Sensitivity = TP/(TP+FN): Probability of correctly labeling members of the targeted class.
- Specificity = TN/(TN+FP) = 1−FalseAlarm: a statistical measure of how well a binary classification test correctly identifies the negative cases.
- False Alarm = FP/(TN+FP)

		Actual Class		
		Positive	Negative	
Predicted Class	Positive	TP	FP	Precision
	Negative	FN	TN	
		Sensitivity	Specificity	Accuracy

Table 3: Confusion Matrix architecture

ii. Area Under Curve (AUC - ROC)

Area Under Curve determines if the models can predict the best classes. The AUC plot displays the rates of FPR against the TPR at different points in [0,1] in which 1 presents as the highest performance [79].

iii. Classification Measurement Coefficient:

Generally, the classification measurement coefficient can be calculated using the classification report in order to measure the quality of prediction from a classification model. It displays how many predictions are true or false to prove the essential classification measurement stated as below:

A. Classification Accuracy

The accuracy rate should be resulted as (TP+TN)/total samples; the ratio of number of correct predicted instances to the total number of input instances. This rate is considering as an effective indicator for the classification error in terms of predicting two or more classes.

B. Precision

Precision of a system is calculated using TP/(TP+FP); Probability that a positive prediction is correct.

C. Recall

The recall of a system calculated using TP/(TP + FN); all the actual positive results divided by the number of the all relevant instances.

D. F1 Score

F1 Score is a harmonic mean between precision and recall with a formula of "(2 * Precision * Recall) / (Precision + Recall)". By examining the F1 Score, we can find the balance between the precision and recall of the model. High rate of precision with low rate of recall produces an extreme accuracy but unfortunately it can miss several predicted instances that are intractable to classify.

Chapter 4: Intrusion Detection: Concept and Related Works

4.1 Intrusion Detection Overview

With the enormous amount of transferred data from different networks, the fast development of network technologies, the ease of accessing various services and information materials become essential part in our daily regular operations. Cybercrimes committed using computer networks lead to billions of dollars' losses, the illegal access into computer system, stealing data and destroying networks infrastructure which in turn affect the cyber resources.

The evolution of the internet, networks and its availability for numerous people makes their resources targeted to attacks for downgrading the performance of the networks and its privacy using several kinds of threats. Therefore, networks security is in real concerns related to sensitive and private resources to provide the stability, availability and integrity of data. Network security includes hardware such as firewall, proxy servers, malicious code detection, antivirus and intrusion detection system (IDS) [69]. Therefore, the Intrusion detection system acts as cyber-security mechanism for networks and related components to protect it from promiscuous activities. It decreases the harm of systems and networks and it can be useful to track, detect and classify attacks. It is considered as one of the most used cyber security mechanism by monitoring the traffic coming in, the vulnerabilities and system configuration. IDS can guarantee that the system is able to provide secured services and identify the attacks access patterns presented in the host or network based systems [80].

The currently used IDS requires human intervention by either creating signature data base or developing the system which makes it far from intelligent. Thus, more sophisticated IDS that support intelligent algorithms is highly desired to provide a prospect alternative to human intervention. It can be capable of detecting both of known and unknown hacking activities using known signature or machine learning techniques. The fundamental task of these intelligent techniques is to discover usable patterns from reliable data sources in order to forecast the attack behavior. Thus, the developing of intrusion detection systems using the machine learning techniques have been deployed from the cyber-scientists to be proposed as solution for the IDS drawbacks stated in the next parts [69] [81].

4.2 Intrusion Detection Systems

It is substantial for intrusion detection system developers to realize and investigate the IDS taxonomy in order to select the preferable characteristics in the building phase. In fact, IDS classification depends on several stages concerning the deployment purpose, detection methods and responses in order to achieve the fundamental security level.

4.2.1 Deployment

As shown in figure [19], the intrusion detection system can be classified according to the detection place as where the traffic is being captured into three topologies stated as follow:

i. Network-based intrusion detection system (NIDS): The NIDS is prominent to monitor and analyze network traffic to detecting the coming network attack. It is considered as a passive role and plays an essential process to flag any suspicious traffic, attempting the scanning of routed packets, analyzing, gathering, alerting and logging the activities into specific storage.

ii. Host-based intrusion detection system (HIDS): The HIDS controls and monitors a machine resource such as log file, system file and action in order to scan the desired activities to involve the installed agent on every system. Moreover, it has the efficiency to check the integrity for local system, Rootkit, Trojans, malwares and virus detections etc.

iii. Hybrid based intrusion detection system that combines both of host and network based IDS by examining the coming packets over the network.

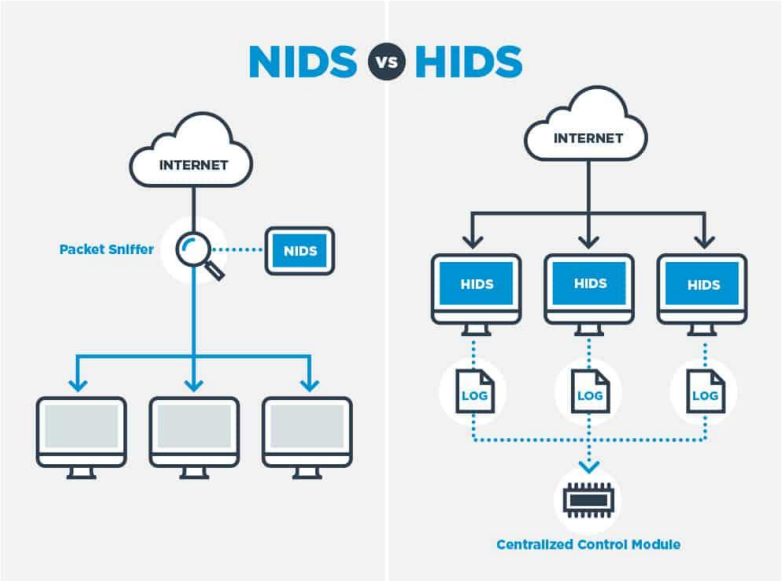


Figure 19: Network and host based IDSs topologies

4.2.2 Detection Methods and Responses

Depending on what is being detected by the intrusion detection system, the detection method can be divided into three categories; the misused, anomaly and hybrid detection systems [82]:

i. Misused detection system depending is employed as a model that depends on the signature of attacks like antivirus and system defender while the coming traffic to the network are compared with the signature database of the familiar intrusions that have been controlled by the security administrator to judge whether the data is normal or not. This kind of detection gives high detection rate of known attack without generating an overwhelming number of false positive

alarms. The drawback of this type of IDS is that it has daily regular update in terms of the signature rules.

ii. Anomaly detection system works as a model that describes the normal behavior and usage patterns of the examined system where the coming traffic to the network is compared with the reference model and can flag any deviation from the normal activities. Thus, it has the ability to be developed using intelligent algorithms to detect coming novel attacks and dynamically can be updated to novel one. The main disadvantage of this IDS is the fact that it produces a high false positive rates due to the coming unknown traffic.

iii. Hybrid detection system is a combination of both misuse and anomaly detection techniques. They are employed to increase the detection rate of known attacks and decrease the false positive rate of unknown attacks.

In addition, the IDS architecture is surrounded of many reliable sensors, detection model and investigation results. It has two possible detection responses: Alerting Response (Passive) and defensive Response (Active). According to Lokman et al. [83], the IDS depends on the proposed configuration, the passive type can find out the attack as well as detect and log it and enable to raise alarm and notifications to specific device. On the other hand, the active IDS has the same target of passive type (detecting, logging, notifying) but it can be upgraded to the prevention mode that permits to block and prevent for coming threat in real time stage.

4.3 Most Used Open Sources IDSs

Open source intrusion detection tools are applied from small and medium businesses to protect users, internal servers and public cloud environments. In this part, some of the most used open source IDSs are stated according to their types and features listed in the table below [4].

Name	Developer	Type	Important Features
SNORT [84]	Martin Roesch, Cisco Systems	NIDS	• The multi-threading can be employed for packet treatment • Shared configuration and features • Plain configuration with scriptable method • Framework with plugin to produce pluggable components • Dynamic detection of services and documentation • Upgraded memory
Suricata [85]	Open Information Security Foundation	NIDS	• High performance of detection • Multi-threading • Multi-purpose of detection (network detection and prevention system with offline analysis). • Multi- OS support (Linux, Windows, mac, etc.)

			• TCP/IP support • HTTP engine with parser and logging. • Extraction of logs
Bro (renamed Zeek) [86]	Vern Paxson	NIDS	• Networks Traffic logging • Powerful scripting language (Bro scripts) • Deploys on UNIX and Mac systems • Multiple protocols support (DNS, HTTP,SMTP, SSL, other) • Real-time analysis • Extraction of logs
OSSEC [87]	Daniel B. Cid	HIDS	• Log monitoring • Log analysis • Rootkit attack identification • Dynamic response • Client Server platform for Linux, Windows, MacOS, etc.) • Real-time notice • Host monitoring
OSSIM [88]	AT&T Cybersecurity	HIDS	• Security information management • Networks and Logs management • Vulnerabilities assessment • Threats and Behavioral detection • Reporting event response • Powerful web interface • Easy installation with virtual system

Table 4: Most used open source IDSs

4.4 Intrusion Detection System based on Machine Learning Techniques

The year of 1980 was the announcement of intrusion detection systems [89]. In fact, several IDSs are suffering of weaknesses due to the large number of false alarms that lead to rise of challenges facing the cyber analysts with harmful effects. This causes destruction in case of unnoticed threat. Developing of IDSs gains major attention to the cyber researchers to overcome the problem concerning the increase of the detection rate to minimize the false one and in terms of the incapability to distinguish unfamiliar attacks since the patterns of the network variate rapidly and several kinds of attacks appear continuously. The prior intrusion detection is performed by human analysts (security analysts) who work with manual decisional efforts and it's considered an intricate job to process huge numbers of attacks. In addition, the

standard intrusion detection system requires human intervention by creating signature data which makes it far from intelligent. Therefore, the advanced IDS supported by machine learning is extremely required to provide an alternative process related to human effort intervention. Thus, to treat this approach, the scientists concentrated the IDSs development based on the machine learning techniques to attain a dynamic intelligent form with accurate detection performance. In addition, the usage of IDS using machine learning as shown in figure [20] achieves the forecasting of attacks built from reliable and preprocessed data sources trained on a model to achieve the decision results. It can be effective compared with the traditional one in observing malicious activities that happen in the networks and systems on real time with worthy detection rate [90].

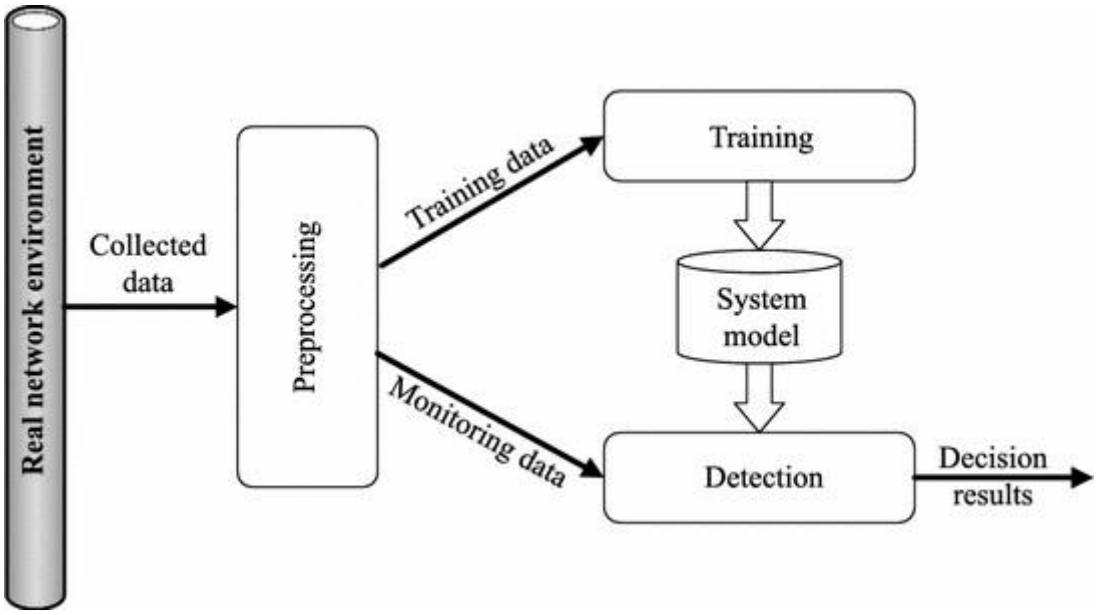


Figure 20: Intrusion detection based on machine learning techniques

In addition, we will present an overview to cover several ML techniques that were implemented from researchers in their intrusion detection models. Further, we will show various approaches that used reliable datasets as shown in table [5].

Bouzida et al. [91] employed the KDD data that were extracted from the DARPA dataset. Due to the lack of new attacks in this data set, they proposed a methodology to add real traffics from their laboratory. Therefore, DT and neural network were used as supervised learning techniques to detect the intrusion based on anomaly detection method. The neural network was conducted to enhance the accuracy rate, but it attained low performance in the detection process for the novel attack. The decision tree presented better results concerning both the accuracy and the novel attack detection. Thus, the researches reached interested results when they compared their performance of the applied models with the old approaches. Finally, the contribution of the researchers proved the applicability of anomaly detection system for both normal and known attack after adding their novel attack records in the training phase [3].

McElwee et al. [92] suggested the alert filtering technique using DNN. Firstly, they gather the log produced by McAfee. After that the training is accomplished for the DNN model to discover the essential security patterns in the logs. Then, the mined significant actions examined by security specialists and the examination results are employed for training data to improve the DNN model to establish an interaction and promotion cycle and facilitating the detection process.

Najafabadi et al. [93] developed forecast models to distinguish regular network traffic from abnormal network traffic through machine learning methods. The suggested architecture integrates the data reduction approach (like feature selection) to eliminate unnecessary and duplicated features to result a reduction in the processing consumed time. They used 4 distinct feature selection approaches for examining 3 classification models (Nave Bayes ,5-nearest neighbor and C4.5 DT). The models are trained and tested using the Kyoto2006+ dataset. The results demonstrated that when feature selection decreases the features, it preserves the similar or does not greatly reduced outcomes suggestively. They determined that in the domain of IDS application, the selecting step for the feature is a critical pre-processing stage that must not be overlooked.

Researcher(s)	**IDS Type**	**Preprocessing Phase**	**Nb. Classes**	**Used Dataset**	**Best Model Performance**
Sharafaldin I et al [94]	IDS for DDOS	Feature Extraction	13	CICDDoS 2019	Random Forest - Precision: 0.77 - Recall 0.65 - F1-Score 0.69
Thabtah et al. [95]	Anti-phishing Model	Feature Selection	2	Phishing Websites	Neural Network epoch (500) - Acc = 93.06% - F1-score = 92.30% - Recall = 91.12% - Precision = 93.71%
Elsayed MS et al. [96]	Detecting Network Attacks	- Features Reduction - Data cleaning - Data Normalization	2	CICDDoS 2019	Deep Learning - Accuracy:0.99
Mamun MSI et al. [97]	Detecting Malicious URLs	Feature Selection	5	ISCX-URL-2016	Random Forest - Precision: 0.97 - Recall: 0.97
Kapil Det al. [98]	Detecting Malicious URLs	Feature Selection and Reduction	5	ISCX-URL-2016	Weka: Random Forest - TPR:0.961;FPR:0.032 - Precision: 0.961 - Recall 0.961 - F1-Score 0.961

Table 5: Related works which used reliable data set in the IDS development

4.4.1 Requirements and Materials

Intrusion detection system based on machine learning can be illustrated in the following important steps; data acquisition through different tools like (sniffing, logging, sensors of hardware tools), data and features

engineering (such as regularization, data cleaning and normalization), modelling phase (e.g. classification, clustering) and performance metrics.

i. Data Acquisition

Providing an informative data set (Input Data) is usually employed for the purpose of analysis requirement. The input data is a collection of instances like vectors of samples, appropriate patterns and raw data which can demonstrate whether the selected pattern is normal or not. This phase in the intrusion detection system consists of collecting data from systems under attacks. The latest interest in research centers focuses on the idea of knowing the corresponding features to select data source to be taken as reference of intelligent algorithm. There is a limitation of research centers in providing a reliable data source having an extensive quantity and quality of data. The restriction comes due to the privacy adopted from the companies which is considered one of the main priorities to maintain the confidentiality of data.

In this Section, we will describe the most used data set that is employed to develop an intelligent IDS. There are two sources to acquire the acquisition of data which are the web server and networks based data. Each one has its own specification and has its primary advantages and disadvantages.

A. Web Server Data

It consists of records collected from logging data which contain user activities that are maintained and created dynamically (Web Server Logs, Data Base, Firewall Logs) [99]. These Log files can be considered as data sources, hence all users' activities get recorded by the log utility. This kind of data highlights on several indicators for serious problems occurring on a system. However, log files contain thousands and millions of activities represented in form of binary formats, plain texts or a combination of both [100]. The web server log is evaluated in many studies as an end device of HTTP request such as IIS for the Microsoft platform and apache for Linux. It registers the visitor's activities that occur on the webserver based on Common Log Format specifications (CLF). Furthermore, the log file contains valuable information including the visitor's behavior concerning the user ID, requested URL, Referring URL, IP Address, the status code and many parameters including the hits that recorded as GET parameter in that file. It is considered as an input data set which represents an essential part in detecting the malicious activities passing through the web server. Thus, many cyber security scientists employ this useful data source in detecting hacking and abnormal activities [69] [80] [101]:

B. Networks Data

The networks based data consists of sources which are already developed by cyber security centers collected from network traffics at which potential attacks get recorded. Therefore, the network traffic will be selected to be exported as dataset. Below we will state the most used data sources in the IDS development as shown in the table [6] [69].

Data set	Source	Year	Description
DARPA [102] [69]	Linco ,Laboratory	1998	It developed for security analysis intent and consist of seven weeks of data acquisition. Each day was formed of BRM audit and TCP dump data which labels each record as output of attack or not. DARPA consists of various network activities such as FTP, email, browser, SNMP, telnet, IRC actions to produce attacks classes of DOS, buffer overflow, guess password, remote FTP, syn flood, Nmap and rookit
KDD [103] [69]	University of California Irvine	1998	It was formed from the tcpdump portion of DARPA 98 dataset. The network traffic recorded by DARPA 98 had been converted to network connection with number of 23 features per connection. KDD99 contains more than 20 classes wich covers several attacks types like the neptune-dos, smurf dos, rootkit, satan, teardrop, pod-dos, few buffer-overflow ect. The records of the traffic concerning the normal and attack classes are combined in a simulated framework. The resulted dataset has a massive number of redundant records resulting with data depravity which leads to a deviation of testing output. Thus, the NSL-KDD was performed based on KDD dataset to solve the presented shortcomings of the KDD.
UMASS [104] [69]	University of Massachusetts	2011	It covers the network packets and traces file about wireless applications components. This data set is produced based on single request attack via the TCP download [86]. Due to the insufficiency variety of the hacking activities which occur on the network traffics, this data set is classified based on a particular requested attacks which makes neither useful in training nor in testing
ISCX [105][69]	University of New Brunswick	2012	It contains two profiles which are "alpha and beta", where alpha is the development of attacks process at which multistage attacks scenario, while the beta profile represents the creation of normal traffic that capture all packet's payload carried by different protocols such as HTTP, SNMP, POP3, FTP. However, the ISCX 2012 suffers from novel network protocols as the latest attacks type are lacked to novel trances
ADFA [80]	University of New SouthWales	2013	It includes two kinds of data the training and testing which has Ten attacks represented in one vector. The covered attacks are the brute force includes the SSH authentication password and FTP, C100-Webshel hacking activities, super user, Linux and Meterpreter Java payloads.
URL dataset (ISCX-URL2016 [106]	Canadian Institute for Cyber-Security	2016	It includes the malicious URLS dataset that contains five classes of attacks: defacement, malware, phishing, benign, and spam. Moreover, the CICIDS2017 it was developed in 2017 which contains benign and incoming network attacks, it contains normal and most up to date network attack, which represent real world traffic. This dataset contains the network traffic exported via the CIC Flow Meter. It will occur as labeled flows based on specific networks traffics features such as the time stamp, source and destination IPs, source and destination ports, protocols and

			attack (CSV files)
CSE-CIC-IDS2018 [106][69][80]	Canadian institute for cyber-security	2018	It contains two profiles, the B-profile to produce benign traffic and M-Profile for making traffic under attack by implementing seven attacks scenario: Brute-force attack, Heartbleed attack, Botnet, Denial-of-Service, Distributed Denial-of-Service, Web Attacks and Infiltration. The CSE-CIC-IDS2018 includes the traffic and their logs where 80 features have been extracted from the captured data using the CICFlowMeter-V3.

Table 6: Most employed data sources in the IDS based on the machine learning techniques

ii. Data and Features Engineering

According to available literatures related to the construction of an intelligent IDS [69] [107], the data and feature engineering and its representation play an important role in the detection process and become the most imperative factors that has influenced the adequacy of an IDS. This process depends on cleaning and segmenting the data as well as reducing the features that does not have impact before the IDS classification process in order to improve the IDS with an increase in both pf computation speed and detection accuracy. There are several features techniques which help in extracting and selecting useful predictors from the gathered data such as latent semantic indexing (LSI) used to capture knowledge from unstructured data and the principal component analysis (PCA) as a feature reduction technique to decrease the number of the selected feature by developing a new preprocessed data that holds the final data with small number of effective predictors. Moreover, the feature selection method can be employed to improve the outcome performance by proposing a feature ranking based recursive feature elimination, using the cross validation method "feature_selection.RFECV" that permits to attain the best accrued features to be used in the learning step.

iii. Modeling Phase

The modeling process is categorizing in several techniques to propose the anomaly classifier that should be used in the learning and detection processes. It can use the machine learning techniques such as the supervised, unsupervised and semi supervised learning. In addition, A Proximity Measurement should be validated which returns numerical rates for its evaluation concerning the final decision of the detection stated in the following part.

iv. IDS Performance Metrics

According to A.K. Saxena et al., 2017, the intrusion detection system (IDS) based on machine learning is employed to automate in high level of accuracy to enhance the fundamental network systems security [107]. For instance, it derives a real time monitoring process as well as incident management in order to provide a behavior related to events collected from networks, security devices, system and applications.

The IDS based on machine learning is evaluated by employing performance metrics to assess the detection performance. Thus, the cyber-researchers apply different measurements to estimate the model effectiveness. Some of those mensuration are listed as follow:

- Attack Detection Rate (ADR): the proportion between the detected total number of attacks by the system to the total number of attacks present in a dataset.
- Attack Detection Rate = Total detected attacks / Total attacks $*$ 100
- False Alarm Rate (FAR): the proportion between the misclassified total number instances to the total number of the normal instances.
- False Alarm Rate = Total misclassified instances / Total normal instances $*$ 100

Table [7] clarifies each phase of the alarm rate; the lower right cell indicates the number of connections classified as attack were they are really an attack (TN) and the upper left cell that indicates the number of connections classified as normal and they are were really Normal (TP). The other cells denoted by the number of misclassified connections. The upper right cell indicates the number of connections classified as attack but they are being really normal (FP), whereas the lower left indicates the number of connections classified as normal but they were really attack (FN) [69].

	Classified as Normal	Classified as Attack
Normal	TP	FP
Attack	FN	TN

Table 7: The detection alarm rates for the IDS measurement based on the machine learning techniques

Conclusion Part I

Nowadays, the cyberspace is considered as one of the most concerned research field that faces the cyber security scientists. In fact, the objective of the thesis is to develop an intelligent model avoiding hacking activities using the machine learning techniques and to provide a security assessment by implementing several cyber experiments that can be taken as a guideline in any educational organization or other organizations.

In this part, we cover the state of the art section that permits to provide a comprehensive overview of our thesis materials. We have mentioned four chapters; the first chapter consists of presenting an overview of the cyber security field. We introduce the cyber-domains and their importance with significant security technologies that can be employed against the cyber-threats. Moreover, the most important of cyber-attacks have been defined to be employed in our works in the following parts of our thesis. The second chapter presents a general review about the web mining techniques to clarify the web usage mining techniques which are used in our applied experiment. We present a related approach which is the cyber security field in terms of detecting the visitors' behavior. In chapter three, we have introduced the machine learning technique by clarifying the fundamental steps for these intelligent techniques. Therefore, we expose several classification models and explain how they can be used in the development of any intelligent model. Furthermore, in the chapter four we have reviewed relevant systems related to proposing the host and network intrusion detection systems. We state the most important approach concerning the requirements phase to be applied on the machine learning.

References Part I

[1] Herjavec Group. (2019). Official Annual Cybercrime Report. Retrieved from https://www.herjavecgroup.com/wp-content/uploads/2018/12/CV-HG-2019-Official-Annual-Cybercrime-Report.pdf.

[2] Internet World Stats. (2021). Internet Growth Statistics. Retrieved from https://www.internetworldstats.com/emarketing.htm.

[3] H. Alyasiri. (2018). "Developing efficient and effective intrusion detection system using evolutionary computation".

[4] Australian Cyber Security Centre. (2017). ACSC Threat Report. Retrieved from https://www.cyber.gov.au/acsc/view-all-content/reports-and-statistics/acsc-threat-report-2017.

[5] Symantec. (2019). Internet Security Threat Report. Retrieved from https://docs.broadcom.com/doc/istr-24-2019-en.

[6] D. Craigen, N. Diakun-Thibault, R. Purse. (2014). "Defining cybersecurity", Technology Innovation Management Review.

[7] R. Toshniwal, K. G. Dastidar, A. Nath . (2015). "Big Data Security Issues and Challenges", International Journal of Innovative Research in Advanced Engineering (IJIRAE), ISSN: 2349-2163 Issue 2, Volume 2.

[8] J. Jang-Jaccard, S .Nepal. (2014). "A Survey of Emerging Threats in Cybersecurity", Journal of Computer and System Sciences, pp. 973-993, Issue 5, Volume 80.

[9] A. B. Magoun. (2014). "Connecting Computers with Robert E. Kahn", in Proceedings of the IEEE, no. 12, pp. 1952-1957, Volume 102.

[10] F. Touchette. (2016). "Network Security", pp. 11-14, doi: 10.1016/S1353-4858(16)30008-3, Issue 1.

[11] E. Ozkaya, M. Aslaner. (2019). "Hands-On Cybersecurity for Finance: Identify vulnerabilities and secure your financial services from security breaches", Packt Publishing Ltd, 178883173X, 9781788831734.

[12] A. SARAVANAN and S. S. BAMA. (2019). "A Review on Cyber Security and the Fifth Generation Cyberattacks", Oriental Journal of Computer Science and Technology, ISSN: 0974-6471, No. (2), pp. 50-56, Volume 12.

[13] Kaspersky Lab. (2020). "What is Cyber Security". Retrieved from https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security.

[14] S. Qadir, SMK. Quadri . (2016). "An Insight into the Most Important Attribute of Information Security", Journal of Information Security 07(03):185-194.

[15] M. M. Alhassana, A. Adjei-Quayeb. (2017). "Information Security in an Organization", International Journal of Computer (IJC), ISSN 2307-4523.

[16] S. Tumin, S. Encheva. (2012). "A Closer Look at Authentication and Authorization Mechanisms for Web-based Applications", 5th World Congress: Applied Computing Conference.

[17] A. Wadhwa, V. K. Gupta. (2014). "Framework for User Authenticity and Access Control Security over a Cloud", International Journal of Advanced Trends in Computer Science and Engineering 6(4):138 DOI: 10.1111/IJCSE213456.

[18] R. S. Limaye. (2013). "The importance of Information Integrity, Security, Networking and Data Protection", International Journal of Innovations in Engineering and Technology (IJIET), ISSN: 2319-1058, Issue 3, Volume 2.

[19] Javatpoint. (2020). "Cyber Security Goals". Retrieved from https://www.javatpoint.com/cyber-security-tutorial.

[20] S. Qadir, S. M. K. Quadri. (2016). "Information Availability: An Insight into the Most Important Attribute of Information Security", Journal of Information Security, 7, 185-194.

[21] P. white. (2020) "Risk Management and Security in The Cloud", La Trobe University, ITC 561, cloud computing.

[22] The OWASP Foundation. (2020). "OWASP Top Ten, Top 10 Web Application Security Risks". Retrieved from https://owasp.org/www-project-top-ten/.

[23] W. R. Cheswick, S. M. Bellovin and A. D. Rubin. (2003). "Firewalls and Internet Security: Repelling the Wilyhacke". Addison-Wesley Longman Publishing Co., Inc.

[24] S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, Z. Zhipeng. (2020). "Securing a Network: How Effective Using Firewalls and VPNs Are?". In: Arai K., Bhatia R. (eds) Advances in Information and Communication. FICC 2019. Lecture Notes in Networks and Systems. Springer, Cham, Volume 70.

[25] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman. (2019). "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges". Cybersecur 2, doi: https://doi.org/10.1186/s42400-019-0038-7.

[26] S. M. Alqahtani, M. A. Balushi and R. John. (2014). "An Intelligent Intrusion Prevention System for Cloud Computing (SIPSCC)", 2014 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, pp. 152-158, doi: 10.1109/CSCI.2014.161.

[27] G. Xin and L. Yun-jie. (2010). "An New Intrusion Prevention Attack System Model Based On Immune Principle". in e-Business and Information System Security (EBISS), 2010 2nd International Conference on. IEEE, pp. 1–4.

[28] A. Salah, M. Shouman, and H. Faheem. (2010). "Surviving Cyber Warfare with A Hybrid Multiagent-Base Intrusion Prevention System", Potentials, IEEE, no. 1, pp. 32–40, Volume 29.

[29] ITRC. (2015). "Data Breach Reports". Retrieved from: https://www.idtheftcenter.org/images/breach/DataBreachReports_2015.pdf

[30] Edgescan. (2020). "Edgescan's 2020 Vulnerability Stats Report Released". Retrieved from https://www.edgescan.com/edgescans-2020-vulnerability-stats-report-released.

[31] MMH. Alansari, ZM. Aljazzaf, M. Sarfraz. (2019). "On Cyber Crimes and Cyber Security". In M. Sarfraz (Ed.), Developments in Information Security and Cybernetic Wars, pp. 1-41. IGI Global, Hershey, PA, USA. doi:10.4018/978-1-5225-8304-2.ch001.

[32] CERT. (2003). "Incident Note IN-2003-03 W32/Sobig.F". Retrieved from http://www.cert.org/incident_notes/ IN-2003-03.html.

[33] Statista. (2020). "Sources Les Plus Utilisées Pour Usurper Une Identité Numérique En France En 2016''. Retrieved from https://fr.statista.com/statistiques/668387/vol-usurpation-identite-numerique-internet-cyberattaque-francais/#statisticContainer.

[34] Crime Complaint Center. (2019). "2019 Internet Crime Report". Retrieved from https://pdf.ic3.gov/2019_IC3Report.pdf.

[35] IFF Lab. (2016). "Cyber-Crimes in The History of Cyber-Attacks". Retrieved from https://ifflab.org/cyber-crimes-in-the-history-of-cyber-attacks/.

[36] IEEE Organization. (2017). "10 of the Largest Corporate Hacks in Recent History". Retrieved from https://innovationatwork.ieee.org/10-largest-corporate-hacks-recent-history/.

[37] CNN. (2019). "7 of the Biggest Hacks in History". Retrieved from https://edition.cnn.com/2019/07/30/tech/biggest-hacks-in-history/index.html.

[38] Center for Strategic and International Studies (CSIS). (2021). "Significant Cyber Incidents List". Retrieved from https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents.

[39] Symantec. (2019). "Internet Security Threat Report 2019". Retrieved from https://docs.broadcom.com/docs/istr-24-2019-en.

[40] F. Shahzad, M. Pasha, A. Ahmad. (2017)." A Survey Of Active Attacks On Wireless Sensor Networks And Their Countermeasures", arXiv preprint arXiv:1702.07136.

[41] The Cisco organization, Most Common Cyber Attacks, 2020, https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html.

[42] GE. Rodríguez, JG. Torres, P. Flores, DE. Benavides. (2020). "Cross-Site Scripting (XSS) Attacks and Mitigation: A Survey", Computer Networks journal, Elsevier, 106960, Volume 166.

[43] M. Flanders. (2019). "A Simple and Intuitive Algorithm for Preventing Directory Traversal Attacks", arXiv:1908.04502v1 [cs.CR].

[44] J. Johnson. (2021). "Worldwide Digital Population as of January 2021". Retrieved from https://www.statista.com/statistics/617136/digital-population-worldwide/.

[45] MJH. Mughal. (2018). "Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview." International Journal of Advanced Computer Science and Applications .

[46] S.Joni, B. Gurpreet. (2015). "Knowledge Discovery in Data-Mining", International Journal of Engineering Research & Technology (Ijert) Ncetems, Issue 10, Volume 3.

[47] S. Kaur, K. Kaur. (2015). "Web Mining and Data Mining: A Comparative Approach", International Journal of Novel Research in Computer Science and Software Engineering, no. 1, pp. 36-42, Volume 2.

[48] S. Brijendra, HK. Singh. (2010). "Webdata mining research: A survey." Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on. IEEE.

[49] P. Sukumar, L. Robert, S. Yuvaraj. (2016). "Review On Modern Data Preprocessing Techniques in Web Usage Mining (WUM)." 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS) 64-69.

[50] A. Kumar, RK. Singh. (2016). "Web Mining Overview, Techniques, Tools and Applications: A Survey", International Research Journal of Engineering and Technology (IRJET), no. 12, pp. 1543-1547, Volume 3.

[51] V. Bharanipriya, V. Kamakshi Prasad. (2011). "Web Content Mining tools: A Comparative Study in International Journal of Information Technology and Knowledge Management", No. 1, pp. 211-215, Volume 4.

[52] K. Pol, N. Patil, S. Patankar, C. Das. (2008). "A Survey On Web Content Mining and Extraction of Structured and Semistructured Data", Emerging Trends in Engineering and Technology, pp. 543-546.

[53] A. kumar, KR. Singh. (2017). "A Study on Web Content Mining", International Journal of Engineering and Computer Science, no. 1, pp. 20003-20006, Volume 6.

[54] A. Kumar, KR. Singh. (2017). "A Study on Web Structure Mining", International Research Journal of Engineering and Technology (IRJET), no. 1, pp. 715-720, Volume 4.

[55] J. Just. (2013). "A Short Survey of Web Data Mining", WDS'13 Proceedings of the 22nd Annual Conference of Doctoral Students – WDS 2013, pp. 59–62.

[56] Z. Zhu, Q. Peng, Z. Li, X. Guan and O. Muhammad. (2018). "Fast PageRank Computation Based on Network Decomposition and DAG Structure", in IEEE Access, vol. 6, pp. 41760-41770, doi: 10.1109/ACCESS.2018.2851604.

[57] AK. Kassem, AES. AL HAJJAR, B. Daya and P. Chauvet. (2018). "A Proposed Methodology for Cyber Security Mechanism According to the Most Popular Detected Attacks for University Web Application", 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 215-219, doi: 10.1109/WorldS4.2018.8611626.

[58] A. Deepa, P. Raajan. (2015). "An Efficient Preprocessing Methodology of Log File for Web Usage Mining", NCRIIAMI - National Conference on Research Issues in Image Analysis and Mining Intelligence.

[59] A. Kusmakar, S. Mishra. (2013). "Web Usage Mining: A Survey on Pattern Extraction from Web Logs", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, pp:834-838, Issue 9, Volume 3.

[60] SP. Nina, M. Rahman, KI. Bhuiyan, K. Entenam, U. Ahmed. (2009). "Pattern Discovery of Web Usage Mining", In Computer Technology and Development, 2009. ICCTD'09. International Conference on, pp. 499-503. IEEE, Volume 1.

[61] S. Dhawan, S. Goel. (2013). "Web Usage Mining: Finding Usage Patterns from Web Logs", American International Journal of Research in Science, Technology, Engineering & Mathematics pp: 203-207.

[62] Cybersecurity & Infrastructure Security Agency. (2020). "Cybersecurity Overview". Retrieved from www.dhs.gov/cybersecurity-overview.

[63] H. Qin, K. Riehle and H. Zhao. (2017). "Using Google Analytics to Support Cybersecurity Forensics", 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, pp. 3831-3834, doi: 10.1109/BigData.2017.8258385.

[64] Verizon. (2020). "2020 Data Breach Investigations Report". Retrieved from https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf.

[65] R. Patgiri, U. Majhi. (2018)." Big Data Security Analytics: Key Challenges", Proceedings of the 2018 International Conference on Data Science ICDATA'18, 151-154 ISBN: 1-60132-481-2.

[66] Force Point. (2020). "Security Analytics Defined". Retrieved from https://www.forcepoint.com/cyber-edu/security-analytics.

[67] M. Mohammed, MB. Khan, EBM. Bashier. (2016). "Machine Learning: Algorithms and Applications", (1st ed.). CRC Press. Doi: https://doi.org/10.1201/9781315371658.

[68] A. Chauhan, G. Mishra, G. Kumar. (2011). "Survey on Data Mining Techniques in Intrusion Detection", International Journal of Scientific & Engineering Research 2(7) .

[69] AK. Kassem, SA. ARKOUB, B. DAYA, P. CHAUVET. (2019). "A Survey of Methods for the Construction of an Intrusion Detection System", Artificial Intelligence and Applied Mathematics in Engineering Problems. ICAIAME 2019. Lecture Notes on Data Engineering and Communications Technologies. Springer, Cham, pp. 211-225, Volume 43.

[70] Y. Roh, G. Heo, SE. Whang (2019). "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective". IEEE Transactions on Knowledge and Data Engineering. pp. 1-1. 10.1109/TKDE.2019.2946162.

[71] T. Yu, H. Zhu. (2020). "Hyper-Parameter Optimization: A Review of Algorithms and Applications". ArXiv, abs/2003.05689.

[72] A. Navada, A. N. Ansari, S. Patil, B. A. Sonkamble. (2011). "Overview of Use of Decision Tree Algorithms In Machine Learning", 2011 IEEE Control and System Graduate Research Colloquium, Shah Alam, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826.

[73] Guo G., Wang H., Bell D., Bi Y., Greer K. (2003). "KNN Model-Based Approach in Classification". In: Meersman R., Tari Z., Schmidt D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. Volume 2888.

[74] K. Fawagreh, M. M. Gaber, E. Elyan. (2014). "Random Forests: From Early Developments to Recent Advancements", Systems Science & Control Engineering, 2:1, 602-609, doiI: 10.1080/21642583.2014.956265.

[75] Y. Freund. (1995). "Boosting A Weak Learning Algorithm by Majority". Information and Computation, 121(2):256–285.

[76] T. Chen, C. Guestrin. (2016). "XGBoost: A Scalable Tree Boosting System". In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. Doi :https://doi.org/10.1145/2939672.2939785.

[77] S. Wan, H. Yang. (2013). "Comparison among Methods of Ensemble Learning", 2013 International Symposium on Biometrics and Security Technologies, Chengdu, pp. 286-290, doi: 10.1109/ISBAST.2013.50.

[78] P. Mahto. (2020). "Local Outlier Factor: A way to Detect Outliers" https://medium.com/mlpoint/local-outlier-factor-a-way-to-detect-outliers-dde335d77e1a.

[79] A. Mishra. (2018). "Metrics to Evaluate your Machine Learning Algorithm", towards data science, https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[80] AK. Kassem, M. El¬Sayed, B. Daya, P. Chauvet, M. Saadeldine. (2019). "A New Feature Representation Method for Intrusion Detection System", 4th International Conference on Computational Mathematics and Engineering Sciences.

[81] AA. Aburomman, M. B. Ibne Reaz. (2017). "A Survey of Intrusion Detection Systems Based On Ensemble And Hybrid Classifiers", Comput. Secur. 65, 135–152. Doi : https://doi.org/10.1016/j.cose.2016.11.004.

[82] SM. Othman, NT. Alsohybe, FM. Ba-Alwi & AT Zahary. (2018). "Survey On Intrusion Detection System Types". 7. 444-462.

[83]SF Lokman, AT Othman, MH Abu-Bakar. (2019). "Intrusion Detection System for Automotive Controller Area Network (CAN) Bus System: A Review". Journal on Wireless Com Network 2019, 184 doi: https://doi.org/10.1186/s13638-019-1484-3.

[84] SNORT. (2013). Retrieved from http://www.snort.org.

[85] Suricata. (2020). Retrieved from https://suricata-ids.org.

[86] Zeek Website(2020). Retrieved from https://zeek.org.

[87] Ossec. (2020). Retrieved from https://www.ossec.net/.

[88] Ossim. (2020). Retrieved from https://cybersecurity.att.com/.

[89] J.P. Anderson. (1980). "Computer Security Threat Monitoring and Surveillance"; Technical Report; James P. Anderson Company: Philadelphia, PA, USA.

[90] [104] A. Hijazi, JM. Flaus. (2018). "A Deep Learning Approach for Intrusion Detection System in Industry Network", BDCSIntell.

[91] Y. Bouzida, F. Cuppens. (2006). "Neural Networks Vs. Decision Trees for Intrusion Detection", in IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), pp. 81–88.

[92] S. McElwee, J. Heaton, J. Fraley, J. Cannady (2017). "Deep Learning for Prioritizing and Responding to Intrusion Detection Alerts". In Proceedings of the MILCOM 2017—2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25, pp. 1–5.

[93] M. M. Najafabadi, T. M. Khoshgoftaar, N. Seliya. (2016). "Evaluating Feature Selection Methods for Network Intrusion Detection with Kyoto Data", International Journal of Reliability, Quality and Safety Engineering, no. 01, p. 1 650 001, Volume 23.

[94] Sharafaldin, AH. Lashkari, S. Hakak, AA. Ghorbani. (2019). "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", 2019 International Carnahan Conference on Security Technology (ICCST). doi:10.1109/ccst.2019.8888419.

[95] F. Thabtah, RM. Mohammad, L. McCluskey. (2016). "A Dynamic Self-Structuring Neural Network Model to Combat Phishing", in International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 4221–4226.

[96] MS. Elsayed, NA. Le-Khac, S. Dev, AD. Jurcut. (2020). "DDoSNet: A Deep-Learning Model for Detecting Network Attacks". IEEE Xplore. doi:10.1109/WoWMoM49955.2020.00072.

[97] MSI. Mamun, MA Rathore, AH. Lashkari, N. Stakhanova, AA. Ghorbani. (2016). "Detecting Malicious URLs Using Lexical Analysis", Network and System Security. doi:10.1007/978-3-319-46298-1_30.

[98] D. Kapil, A. Bansal, Anupriya, N. Mehra, A. Joshi. (2020). "Machine Learning Based Malicious URL Detection", International Journal of Engineering and Advanced Technology;8(4S):22-26, doi:10.35940/ijeat.d1006.0484s19.

[99] B. Deokar, A. Hazarnis. (2012). "Intrusion Detection System using Log Files and Reinforcement Learning", International Journal of Computer Applications (0975 – 8887), No.19, Volume 45.

[100] M. Xie, J. Hu, J. Slay. (2014). "Evaluating Host Based Anomaly Detection Systems: Application of The One Class SVM Algorithm to ADFA-LD", 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Xiamen, IEEE, pp. 978-982.

[101] R. Meyer, C. Cid. (2008). "Detecting Attacks on Web Applications from Log Files", SANS Institute InfoSec Reading Room.

[102] [115] C. Brown, A. Cowperthwaite, A. Hijazi, A. Somayaji. (2009). "Analysis of the 1999 DARPA/lincoln Laboratory IDS Evaluation Data with Net ADHICT", in Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Piscataway, NJ, 1–7.

[103] M. Xie, J. Hu, J. Slay. (2014). "Evaluating Host-Based Anomaly Detection Systems: Application of The Oneclass Svm Algorithm to ADFA-LD", 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 978–982.

[104] University of Massachusetts Amherst. (2011). "U Mass Trace Repository. Optimistic TCP ACKing". Retrieved from: http://traces.cs.umass.edu,2011.

[105] A. Shiravi, H. Shiravi, M. Tavallaee, AA. Ghorbani. (2012). "Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection", Computers and Security, 357–374, Volume 31.

[106] AH. Lashkari, A. Seo, GD. Gil, AA. Ghorbani. (2017). "CIC-AB: An Online Ad Blocker for Browsers", UNB Research Expo.

[107] A. K. Saxena, S. Sinha and P. Shukla. (2017). "General Study of Intrusion Detection System And Survey Of Agent Based Intrusion Detection System", 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, pp. 471-421.

Part 2: Enhancement of the Defense Level for the Employed Cyber Security Mechanisms in the Lebanese University

Introduction 73

Chapter 5: Web Attacks Penetration Testing and Analysis 76

5.1 General Overview 76

5.2 Applying the Penetration Testing 76

5.2.1 Security Testing and Penetration stages 77

5.3 Experimental Results: Security Suggestions and Solutions 81

5.3.1 Fixing the Vulnerabilities 81

5.3.2 Improving the Fundamental Web Server Security 83

5.3.3 Visitor's Behavior Analysis 84

Chapter 6: Detection of Visitor's Behavior based on Web Mining Techniques 85

6.1 General Overview 85

6.2 Designing and Applying the Web Usage Mining Tools 85

6.2.1 Tools Requirements and Implementation 86

6.2.1.1 Data Collection and Selection 86

6.2.1.2 Tool Selection 87

6.2.1.2.1 Deep Log Analyzer Tool 87

6.2.1.2.2 The Security Analysis Tool 88

6.3 Experimental Results and Analysis 92

6.3.1 The Deep Log Analyzer Tool Result 92

6.3.2 The Security Analysis Tool Result 95

Chapter 7: Host based Intrusion Detection System based on Text Mining and Machine Learning 97

7.1 General Overview 97

7.2 The Proposed HIDS Architecture 97

7.2.1 Data Collection 98

7.2.2 Data Preprocessing 99

7.2.2.1 Data Preparation 99

7.2.2.2 Data Cleaning 100

7.3 Feature Representation Method 101

7.3.1 DOC2VEC Model 103
7.4 The Applied Machine Learning Methods and Classification Preliminaries 105
7.5 Experimental Results and Discussion 108
Conclusion Part II 113
References Part II 114

Introduction

Résumé

Cette partie, intitulée amélioration du niveau de défense pour les mécanismes de cybersécurité employés à l'université libanaise est formée de trois chapitres. A travers ces chapitres, nous avons décrit les différentes études d'évaluation de la sécurité menées à la faculté de technologie de l'université libanaise.

Dans le cinquième chapitre, nous avons proposé un mécanisme d'amélioration de la sécurité du système réseaux au sein du département "génie des réseaux informatiques et télécommunications (GRIT)" de la faculté de technologie. Par conséquent, nous avons appliqué la technique de test de pénétration selon le top 10 OWASP. Cette technique nous a permis de découvrir les vulnérabilités testées en plusieurs pages web concernant les attaques les plus courantes telles que l'injection SQL (SQLi), le cross-site scripting (XSS) et l'exposition de données sensibles. Nous avons délibéré des suggestions de sécurité apportant des solutions aux informaticiens afin de protéger les systèmes contre les cybercriminels. Ainsi, nous avons garanti l'efficacité de notre système en maintenant toutes les vulnérabilités détectées pour atteindre les normes de sécurité essentielles.

Dans le chapitre six, nous avons présenté une méthodologie basée sur les techniques du web mining afin de détecter le comportement du visiteur. Cet objectif a été atteint en proposant deux outils: un analyseur de log (Deep Log Analyzer) et un modèle d'analyse de sécurité. Ces outils ont amélioré l'efficacité du système en identifiant plusieurs approches sur les activités des visiteurs et les actions anormales. D'une part, l'analyseur de log a permis d'identifier plusieurs approches en termes d'activités des visiteurs, de leurs comportements, du contrôle des ressources d'accès et les erreurs de hits, etc. D'autre part, l'outil d'analyse de la sécurité a permis de détecter les menaces qui peuvent faire face au serveur Web où 15 attaques ont été découvertes respectivement: 10 pour SQLi, 3 pour XSS et 2 liés à la traversèrent de répertoires. Ce dernier a été développé en utilisant le moteur de règles et il était basé sur une base de données comprenant les modèles d'attaques les plus importants utilisant l'expression régulière.

Dans le chapitre sept, un système intelligent de détection d'intrusion hôte (HIDS : Host-based Intrusion Detection System) a été développé en utilisant la technique de text mining. Pour cela, nous avons construit un ensemble de données de classification de texte fiables comprenant 6000 enregistrements d'URL malveillantes liées aux attaques de serveurs web les plus populaires. Ainsi, nous avons prouvé qu'il existe une hypothèse compliquée incluant les méthodes de représentation des caractéristiques avec trois limitations majeures: faiblesse dans la récupération des informations sur l'URL, nécessité d'un travail manuel pour extraire les caractéristiques les plus importantes, et incapacité de capturer les informations utiles sur les requêtes HTTP GET invisibles. Pour pallier ces limites, nous avons suggéré le modèle Doc2Vec comme méthode de représentation des caractéristiques dans notre HIDS. En outre, nous avons appliqué plusieurs techniques d'apprentissage automatique telles que KNN, SVM, l'arbre de décision et MLP afin

d'offrir un système efficace de classification. Par conséquent, ce système a atteint la capacité de détecter les attaques SQLi, XSS ainsi que les attaques par traversée de répertoires. De plus, Il est important de noter que, lors de la phase de test, le perceptron multicouche (multilayer perceptron MLP) s'est avéré être le modèle le plus précis à 90.67% pour prévoir ces attaques, suivi par le KNN qui a marqué le deuxième score avec 88,17 %, puis l'arbre de décision avec 86,08 %. Enfin, le SVM a obtenu le plus faible taux de 82,67%.

Overview

This part contains three chapters that explain the enhancement of the defense level for the employed cyber security mechanisms in the Lebanese university.

In the first chapter we will examine the security level and propose a cyber-security mechanism for the faculty of technology at the Lebanese university. We will apply the study on the system of computer and communications networks engineering department. Our target is to protect the system against cybercriminals and cyber-attack. Therefore, we will apply several penetration testing techniques according to some of the most main security risks mentioned in the top 10 OWASP. In this study we will achieve the target of discovering the vulnerabilities concerning the popular attacks such as SQL injection (SQLi), cross-site scripting (XSS) and sensitive data exposure. Furthermore, we will present the fundamental security solutions and suggestions to fix the detected vulnerabilities and to assist the IT administrator to protect the faculty system against the cyber threats.

In the second chapter, we will apply a web mining technique at the faculty of technology in the Lebanese university. We will present the importance of this technique for detecting behavioral approaches related to the web visitors. In addition, we will discuss the implementation phase of two tools using the web usage technique. We will select our data source from the university apache web server. The first tool will permit us to detect several results in terms of the visitor activities, their behaviors and the access resources control. Furthermore, we will develop a security analysis tool using the rule based method. Then we will construct a proposed data base contains more than 100 patterns concerning the most important attacks. This model will achieve the ability to detect the malicious URL such as SQLi, XSS and path traversal that may counter our web server. Ultimately, we will visualize our experimental results by generating output reports. These reports will highlight on the detection of the visitor's behavior of the web server usability and several kinds of cyber-attacks.

In the third chapter, we will propose a host intrusion detection system based on the text mining technique. We will construct a textual dataset that includes 6000 records of malicious URLs related to the most popular web server attacks. One of the challenges of working with a textual data source is suffering from the feature methods, this challenge derives us to propose the DOC2VEC model as a feature representation method. Furthermore, in the final part of this chapter we will explain our attacks' classification preliminaries. We will apply several machine learning techniques to offer the best efficient classification system. In the testing phase, those models will achieve the ability in detecting the SQLi, XSS and directory traversal attacks.

Chapter 5: Web Attacks Penetration Testing and Analysis

5.1 General Overview

The electronic information in the internet network has become a complete part of our daily lives. All the institutions, such as universities, schools, medical institutes and even the financial institutions use this kind of networks for their appropriate functioning. They use the network to gather, treat, store and share massive data.

The internet networks protection presents as a controversial topic. It takes an important concern in many companies especially the financial ones since the protection part is one of the most important points to be observed. Hence the role of cyber security concept is to protect any exchange of electronic information. In addition, the internet information is extracted, collected and shared between the internet users. Therefore, the integrity of cyber data presents as an essential research area in cyber security. The existing cyber-attack generates illegitimate authorization and authentication and big traffic to interrupt services. Therefore, the cyber security mechanism is deployed as a safety for cyber components against threats related to discretion, reliability, and accessibility [1]. Hence, safety mechanism must be complied with several cyber analysis methods to avoid the networks and systems from breaches related to the vulnerabilities appearance. The vulnerabilities can be exploited by attackers over the networks due to several failures of its components like hardware fragility, system bugs, malicious code from local or remote attackers.

In this chapter, we tested the security level of the applied cyber-security mechanism in the faculty of technology at the Lebanese university. The system of the department of computer and communications networks engineering has been used in our study. Therefore, we applied the penetration testing techniques according to the polices suggested by the top 10 web application security risk rules OWASP [2]. In our experimental results we discovered several vulnerabilities concerning the SQL injection (SQLi), cross-site scripting (XSS) and sensitive data exposure. subsequently, we proved the major mechanism that should be applied by the IT administrator to protect the system against the current and the latest threats.

5.2 Applying the Penetration Testing

In this part, we debated the penetration testing techniques that can be applied to search for the most important vulnerabilities that may face the CCNE web system as shown in figure [21]. We tested the vulnerabilities suggested from the top 10 OWASP such as SQL injection, XSS and sensitive data exposure on different web pages. Afterwards, we proposed security suggestions and solutions to be performed against cybercriminals which can threat our web server. Thus, we will be able to prove and enhance the fundamental cyber-security mechanism that should be applied in the university according to the possible coming security risks to that system.

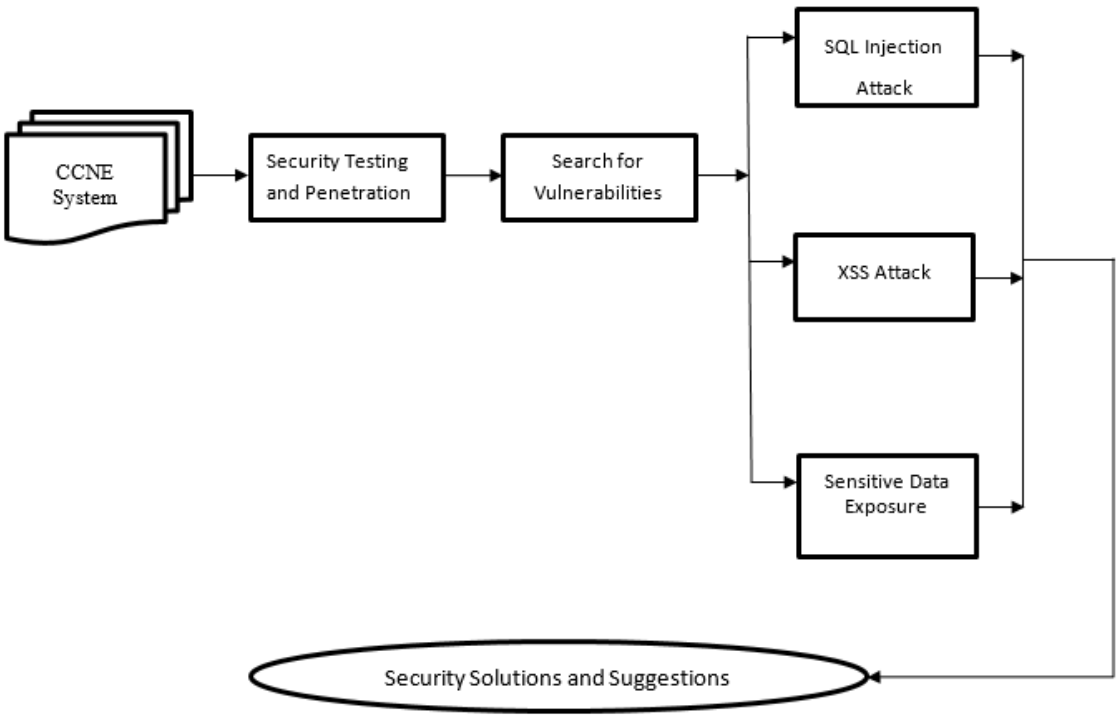


Figure 21: General architecture of the applied penetration testing

5.2.1 Security Testing and Penetration stages

In this part, we will discuss the applied security testing and penetration stage over the selected web pages. In our study, the best interactive pages in which users mostly are working on has been examined. In fact, the CCNE system has three kinds of authentication, each one of them has some features that can be added as an administrator like the head of the department, the instructors and the students. Every specific user has several permissions.

The authentication page provides the users the accessibility of their profiles like the profile page, grades, training files, senior project, academic registration and others. Therefore, we applied several security testing and penetrations in terms of detecting the vulnerabilities related to sql injection, xss and sensitive data exposure. We deduced that these pages aren't well secured and they contain some vulnerabilities listed as follows:

i. Sql Injection:

In this experiment, we studied several pages with respect to the SQL Injection (SQLi) after we applied several pentesting methods. We started by testing whether the log in the page can be vulnerable to SQLi or not and then we achieved easily bypass an authentication by sending special crafted SQL query as shown

in the figure [22] below. We concluded that the login page is not fully protected, we accessed them bypassing several SQLi patterns.

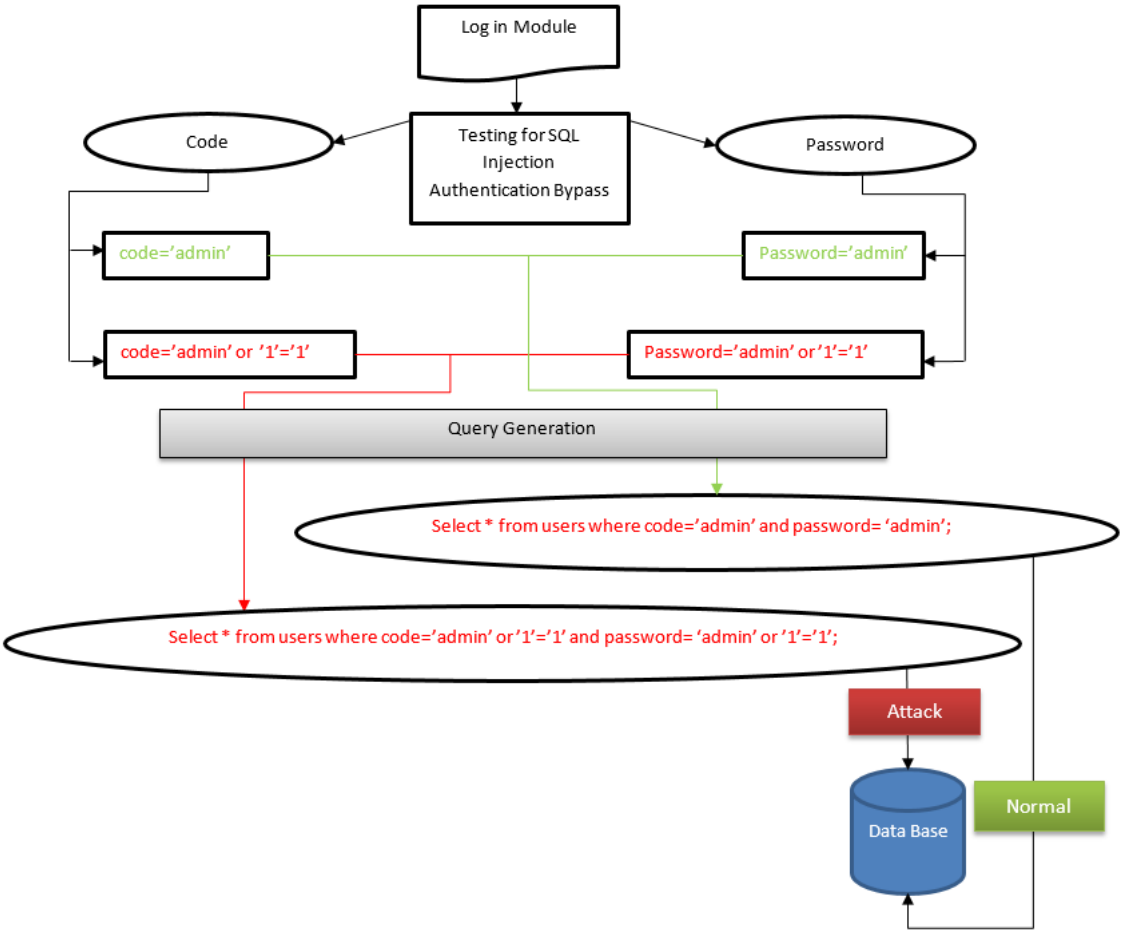


Figure 22: Applying the SQL injection attack on the login module

Moreover, we tested another page (student edit profile) and we noticed that the code source of the edit profile page applies JavaScript validation over the inserted personal information before directing it to be forwarded by an ajax script which interacts dynamically to send the updated data asynchronously to the corresponding server. Referring to figure [23], the detected vulnerability was proven while the inserted data was adopting with the student code. Thus, we can access the personal information and change the phone number and the email address for each student without any authentication. Moreover, the penetration can be attained by sending an SQL injection pattern instead of the code like ('or 1 = 1 - "or 1 = 1 -) in order to change the profile of all students without any permission. Thus, we designed our attack page to steal the student's accounts. After conducting it, the victim page seems to not be able to scan for authentication process, this conflict destroys the student's information and denies all the related services in this page.

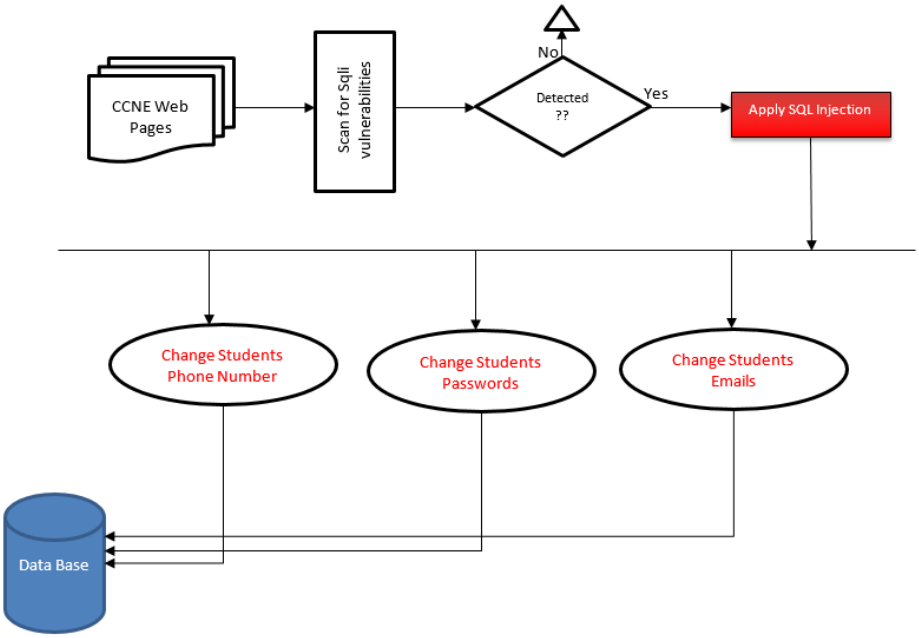


Figure 23: SQL injection attack to destroy the student's accounts

ii. Cross-Site Scripting (XSS)

In this experiment as presented in figure [24], we tested the instructor's drop box page and we concluded that it is not well secured in terms of detecting several vulnerabilities. In fact, the drop box page gives the instructors the permission to download and upload any file from/to the system. We proved that the extensions of the allowed files are (PNG, JPG, JPEG, PDF, XLS, XLSX, DOCX, DOC, ZIP, RAR). After checking and evaluating this page, we found that it includes two kinds of dangerous vulnerabilities related to the failure in validation (XSS, upload file). Therefore, we developed an XSS malicious script to steel the session of each user by adding some penetration code into the file title. Thus, we reached the registered password to display in a prompt message and save it in the browser's security settings as shown in figure [25].

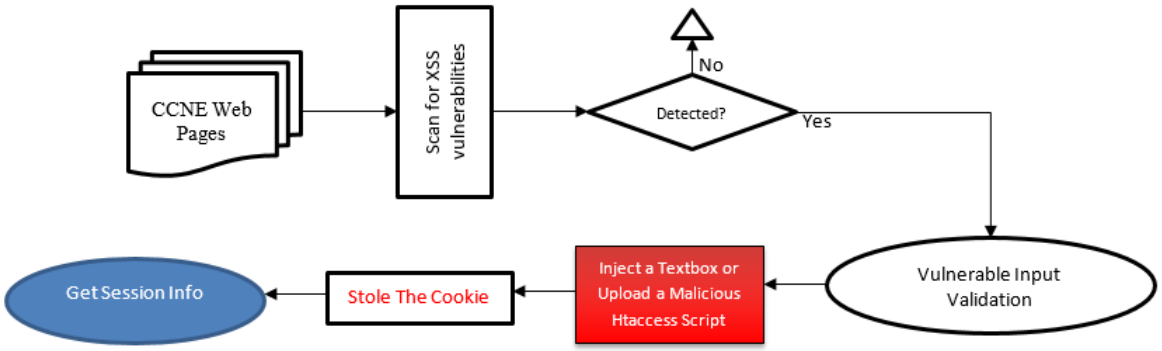


Figure 24: Applying the XSS Attack on the ccne web pages

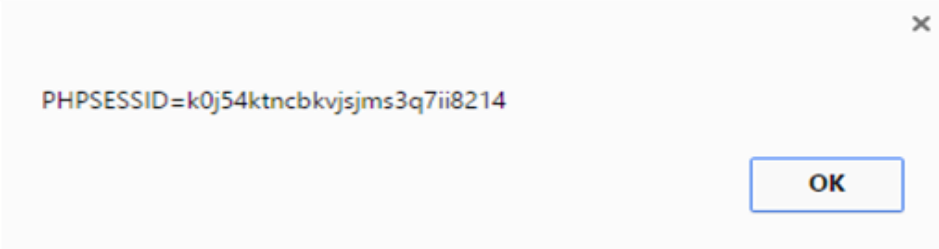


Figure 25: The achieved result using the XSS attack on the drop box page

iii. Sensitive Data Exposure

In this experiment we tested as shown in figure [26] the senior project page and we detected the vulnerability while the page is in the loading phase. We found an AJAX code running at the same time to fetch for the added projects and the selected students for each senior project. Indeed, the AJAX code shouldn't display any result other than the essential information. Subsequently, after analyzing the revealed AJAX output, we deduced that it contains many sensitive information. Thus, we detected the sensitive data exposure that contains the username and the password for all students.

Then, another penetration testing was accomplished while studying the client side of the academic registration page. We concluded that there is a failure in the validation concerning the unchecked courses. Therefore, the page sends the code and the personal content of the students whether they were registered or not without any validation. Thus, there is a capability to exploit the credit's number for the academic registration by increasing the allowed number of registered courses.

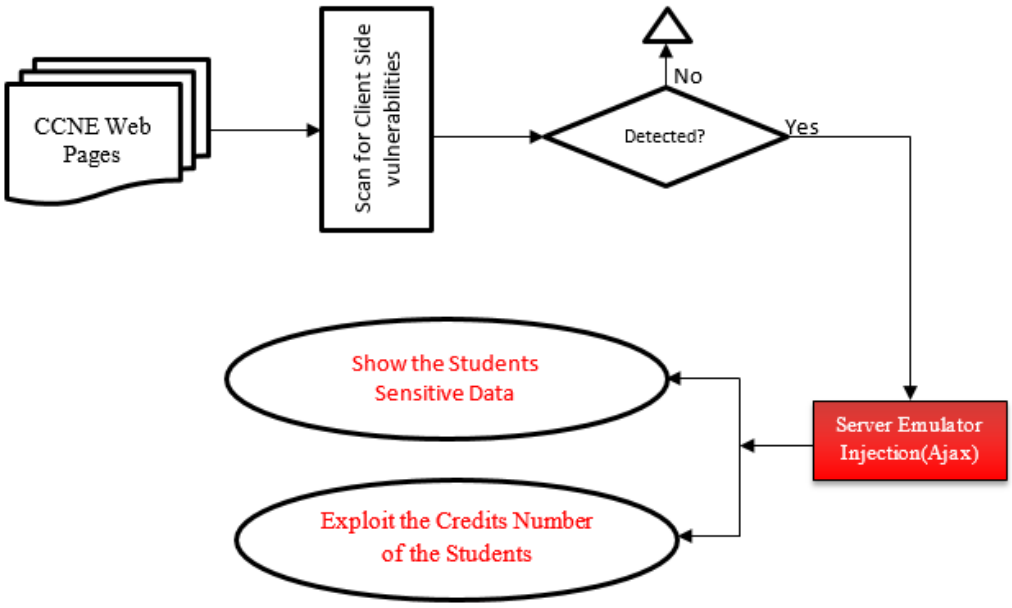


Figure 26: Applying the sensitive data exposure

5.3 Experimental Results: Security Suggestions and Solutions

Since vulnerabilities can be exploited by the attackers that lead to cyber-attack, we can deduce that the threats that face the network is a serious growing problem in the cyber-world and it becomes more complicated. Some threats come from hardware failure, extant vulnerabilities, bugs in a program, malicious code or intrusion from local or remote attacker. In this part, we will present the fundamental security solutions and suggestions to fix the detected vulnerabilities for assisting the IT administrator to protect our system against the cyber threats.

5.3.1 Fixing the Vulnerabilities

i. In case of SQl injection vulnerabilities, the developers must:

A. Use parameterized queries:
   - Declare all the SQL statements to be transmitted in every parameter to the SQL query.
   - Do not create SQL queries dynamically, they have to be used with Statements called Parameterized Queries.
   - Use the bindParam() function by means of the PDO that can be defined with parameterized queries.

B. Use of stored procedures:
   - Group the SQL statements into a logical unique collection to produce an execution schema and let the statements to be parameterized.
   - Use of procedure to be subsequent stored and utilized several times. Thus, when we want to perform the query, we just call the stored procedure instead of coding it many times.

C. Whitelist input validation:
   - Use regular expressions for whitelists input validation such as form, blog entry and controls.
   - Define the value of the drop-down list or radio button input data and limit it to be matched with the desired options.

D. Escaping all user input data:
   - Employ the character-escaping for data input provided by the user and the database management system (DBMS) to restrict the conflict of the SQL code written by the developer.
   - Use the mysql_real_escape_string() function to avoid the inserted characters that drive to an abnormal SQL statement.

E. Errors must not be displayed to the user:
   - The user should not be able to see the any SQL error.
   - An urgent error message must be shown as generated message about the allowed authority of sensitive information.

F. Logged the errors:

- The database errors must be logged and extracted in a secured file or to the server error log.
- The file should be protected and not be attainable to an attacker through the web server.

ii. In case of XSS vulnerabilities, the developers must:

A. Filter and control the input form to be classified as expected or valid input.
B. Encode the output data in a way the user controls it in HTTP responses to deny it from being clarified as active content.
C. Take these obligatory steps depending on the output:
   - Use "HTTP only cookie flag" to encode the HTML in case of unsupported Data which can be inserted into the HTML tags Content.
   - Attribute encode in case of unsupported data would be inserted into the HTML Common Attributes.
   - Use "Auto-Escaping Template System" to encode the CSS in case of unsupported data might be inserted into HTML Style attribute.
   - Use "modern JS frameworks" to encode the JavaScript code in case of unsupported Data would be inserted into JavaScript Data Values.
   - Encode the URL in case of unsupported data shown be inserted into the HTML via URL Parameter Values.
D. Use suitable response for headers via the Content-Type and X-Content-Type-Options headers to ensure the correct responses.
E. Provide a Security Policy to minimize the side effect of any occurred vulnerabilities related to XSS attack.
F. Use authorized locations to insert only trusted Data.
G. Avoid and control any JavaScript URLs.
H. Deny DOM attribute based XSS.

iii. In case of sensitive data exposure, the developers must:

A. Encrypt the sensitive data in the databases.
B. Check the corresponding Type of any variable that can be expected.
C. Cast and validate the data before putting it into any variable.
D. Encode special characters with the corresponding HTML code.
E. Check the presence of all expected arguments.
F. Constrain the numbers between interval of values.
G. Check the allowed value that belongs to the list (select, radio, checkbox...).
H. Restrict or limit the length of the value with minimum and maximum size.
I. Verify the value using regular expressions.
J. Check if the null value must be accepted or not.

K. Accept only by default the alphabetical letters and/or numbers. Thus, all other characters should be refused. In case that other characters must be authorized, they must be limited to a predefined list or replaced by HTML codes.
L. Define CharSet tag that can be declared on the page head.

iv. Authentication fundamental security:

A. Session IDs must have a limit lifetime in a way it is no longer usable.
B. All the pages in a web application must ensure that the user has an authentication permission to be sure that there is no stolen identity.
C. After a predefined period of inactivity, the session ID must be invalidated.
D. The session ID should be destroyed even the automatic log out ones when we close the browser of user agent.
E. Cookies have to be protected by creating two attributes:" Secure" that prevent the sending of a cookie to unencrypted channel and "HTTPonly" that forbid access to JavaScript.

5.3.2 Improving the Fundamental Web Server Security

Enhancing the cyber-security in the web applications is considered as a serious concern to any ecommerce and online activities which manage its operations via the cyber space. In the latest decades, one of the most cyber-victims is the web servers of companies, universities and governmental organization due to the sensitive data that they generally host. Thus, securing web server is considered as significant for securing their components such as website, web pages and web application as well as the main network around it.

The web server can be secured according to specific configurations employed on the "htaccess" file which is a hidden text file used by Apache web server that helps the developer to configure any website without the need of modifying or creating manual server configuration. It is generally placed in the root (ex: www) of the website, this file can be located in other places depending on the configurational needs and in which files and folders should be influenced and related. Therefore, this kinds of configurations usually used to improve and enrich the security by adding some instructions listed as follows:

A. Protect any sensitive folder with a predefined password using ".htpasswd".
B. Enable and disable the "mod_rewrite".
C. Allow or deny an IP address related to malicious user to access the website.
D. Manage the privileges that are assigned to database account.
E. Approve the selected DB Users or Views.
F. Manage the page errors to be redirected to a defined page.
G. Setting/Unsetting Apache server environment variables.
H. Manage the Multipurpose Internet Mail Extensions (MIME) type to be assigned for particular files in the root.

I. Limit the size of uploads and downloads.
J. Setup up the directory index for the website.
K. Configure and control the file types.
L. Manage the cache control which is a complement to the expire headers depending on the server or the used browser by visitors.
M. Enable and disable the server side that includes (SSI) to call CGI scripts or HTML content.
N. Prevent the web directory listing.
O. Changing the default charset and language headers.
P. Redirect the non-www URL to be www URL.
Q. Redirect the entire website to https.
R. Prevent the Image Hot linking.
S. Manage the redirecting process.
T. Re-write the URL's: Redirection Rules.

5.3.3 Visitor's Behavior Analysis

Visitor's behavior analysis is a mining technique that conducts specific study concerning the produced activity in the website. This mechanism takes place to apply several qualitative applications that track the website's performance, recognize the visitors' behavior, identify the illegal experiences, restrict their activities, enhance and monitor the website security. Moreover, we can follow which pages of the web site were visited by the users and may contains vulnerabilities specifically if they were redirected to the page 404 or 500. Thus, for instance we can manage the malicious visitors who's redirected to page 500 or another pages more than once and classified them as attackers in order to deny their accesses and block their IPs.

Chapter 6: Detection of Visitor's Behavior based on Web Mining Techniques

6.1 General Overview

The evolution of the internet in recent decades enlarge the website's reports with user activity records. These data are mostly registered in the web server which are stored in a log file. Aside from detecting the visitors' activities and their behaviors, the web usage mining technique can be effectively used to detect the existing issues and threats. It permits us to extract valuable results related to the detection of cyber-threat approach.

In this chapter, we applied the web mining techniques by selecting our data source from the apache web server in the faculty of technology at the Lebanese university. In addition, we discussed the phases of implementing two tools using the web usage technique. The first tool achieved the detection of several approaches about the visitor's activities and behaviors and the access resources control. On the other hand, we developed a security analysis tool with a proposed data base which includes more than 100 attack patterns. Thus, our model detected several kinds of attacks that faced the web server such as SQLi, XSS and path traversal. Afterwards, we visualized our experimental results related to identifying of the visitor's behavior and the detected attacks.

6.2 Designing and Applying the Web Usage Mining Tools

The web usage mining is one of the significant and fast developing zone. It takes an important part of the latest analysis technology to discover the user's behavior. The feedback concerning the user's activities and any problem including the cyber-security approach represents as a principal raison of applying the web mining technique. In addition, when a user request for specific or particular resources from the web site, each request will be recorded and stored in a log file within the web server and it refers to the browsing behavior of the user.

Therefore, we designed two tools which are the deep log analyzer tool associated with a developed model named as the security analysis tool. These tools require the web server log file to analyze a suitable pattern concerning the visitor's behaviors. They generate statistical and web usage mining reports to identify the detected behaviors. The first tool was employed to identify the approaches for the visitor's activities, their behavior and the access resources control. On the other hand, the security analysis tool was examined for cyber security intent. It detects several kinds of attacks that may face any web server. The next parts will state the steps including the requirements' tools and the implementation phases as shown in the figure [27].

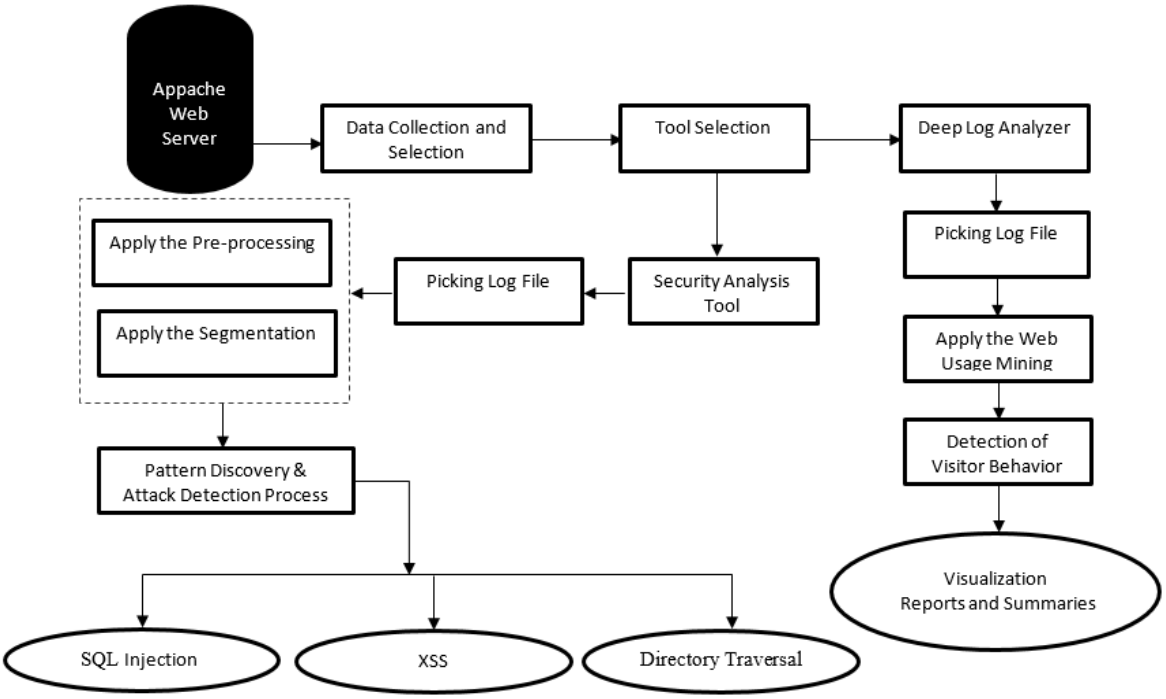


Figure 27: Architecture of the proposed web usage mining tools

6.2.1 Tools Requirements and Implementation

This part consists of representations for each step used in our proposed web usage mining tools as stated in the following parts.

6.2.1.1 Data Collection and Selection

In web usage mining, data can be collected from multiple resources such as: files (image, sound, video and web files), operational databases and server log files. Indeed, when users communicate and interact with any website, the interaction's details and the request activity resulted by the web visitor will be automatically recorded and stored in the web server log file [3]. Thus, in this section, we present the data collection phase that was applied in our research which has been extracted from our faculty web server. The collected data was extracted from log file during a period of four days on February 2018 as shown in the table [8].

Access Log File Details	
File Name	iut.ul-iut.net-Feb-2018
Period	23 Feb 2018 – 26 Feb 2018
Number of entries	6742

Table 8: Data collection description

In addition, the basic information recorded from log file can be summarized in the table [9] below:

Parameters	Description

Username	This identifier will discover who visits the website. The identification of the user principally would be the IP address.
Visiting Path	The path that the user typed while visiting the website.
Path Traversed	It distinguishes the path taken by the user via different links.
Time stamp	The durational time in which the user spends on each web page while surfing through the website, this record recognized as a session.
Last visited Page	The visited web page by the users before the leaving.
Status	It declares the responses, for instance: Successful Response (200), Redirection Message (301, 302) Client Error Message (400,401 ,403, 404) and Server Error Response (500, 503).
Success rate	The number of downloads made and the number of replicating activities experienced by the user that can specifies the success rate of the website.
User Agent	This is the browser that can indicate from where the user sends the request to the web server. It will be formed as a string that characterizes the type and the version of browser software being used.
URL	The resource of the user access.
Request Type	The method chosen for transferring data such as GET, POST.

Table 9: The basic information recorded from log file

Further, the web log stores the visitor's activities with an unstructured format. It can't be used directly for mining purpose, many techniques should be applied as listed in the next parts.

6.2.1.2 Tool Selection

A variety of tools can be proposed to assist the web administrator to apply analysis tasks for mining purpose. In this part, we will present our tools selection used in the web usage mining implementation which are the deep log analyzer [4] and the security analysis tool. The second tool which is our developed security analysis model that permits to detect the visitor's behavior in terms of cyber security approach.

6.2.1.2.1 Deep Log Analyzer Tool

In this part, we will present the applied deep log analyzer as an advanced web analytics tool. It permits to analyze and detect the web visitors' behavior and attains the complemented web usage analytics and statistics. The requirements and the implementation phase of this tool are stated as follow:

i. Picking the log file

We picked the collected log file as input data in the tool to be examined in the implementation phase. This kind of source contains records about all web pages and resources requested by the users.

ii. Applying the web usage mining technique

We applied the web usage mining technique implemented as shown in figure [28]. It consists of several steps such as; data acquisition pre-processing, pattern discovery and pattern analysis. This technique was conducted to employ the picked data to find patterns about the visitor. It permits to grab and extract meaningful behavior related to its activity and other approaches.

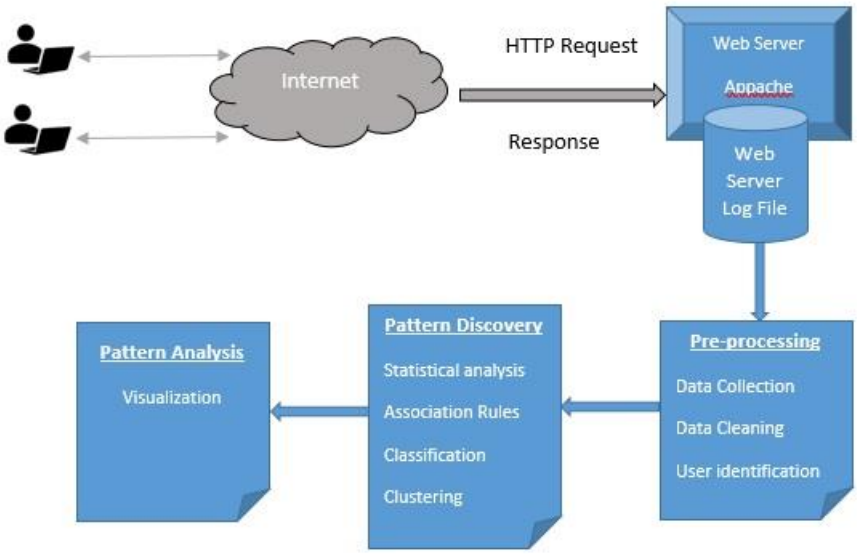


Figure 28: The architecture of web usage mining methodology

iii. The detection of the visitor behavior

Our target by employing this tool was to achieve the aim of detecting the visitor's behavior and the related activity. The detection can be used to produce a dynamic monitoring and extract summaries concerning the detected patterns. These summaries were used to analyze the discovered behavior related to the visitors of our faculty web site. Thus, the IT administrator attains the target to improve the usability of the website. In our experiment, we highlight on some detection services' results such as; general activities, visitor's activities and access resources control.

6.2.1.2.2 The Security Analysis Tool

In this section, we will state the essential requirements used to develop the security analysis tool. In fact, there are two majors of attack detection which are the rule based and the anomaly detection. The log file analysis is considered as a rule based detection for web attacks. Therefore, we develop this model using

python language to detect the discovered malicious patterns related to cyber-attacks. The tool requirement will be discussed by taking each implementation step stated as follow:

i. Picking the Log File

In our proposed security analysis tool, we studied the apache log file as an input data source with the parameters stated below. We converted it into useful and understandable form. Thus, it can be picked in our tool as shown in the figure [29] and represented with the following format as displayed in the table [10]:

"%h %l %u %t \"%r\" %s %b \"%{Referrer}i\" \"%{User-agent}i\"

%h	The host IP address
%l	The "hyphen" in the output indicates that the requested piece of information is not available.
%u	User id of the person requesting the document as determined by HTTP authentication
%t	Date-time and zone
%r	Request
%s	Status
%b	The last entry indicates the size of the object returned to the client, not including the response headers. If there is no content was returned to the client, this value will be "-"
%{ Referrer}	The "Referrer" (sic) HTTP request header gives a feedback for the site that the client reports have been referred from
%{user-agent}	The browser used by the client

Table 10: The proposed parameters used to be matched with the log file

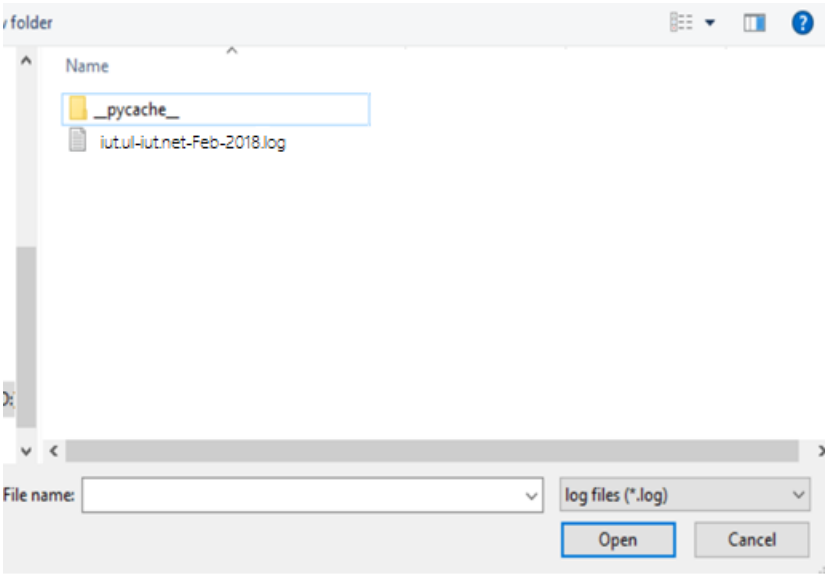


Figure 29: Picking the input data in the selected tool

ii. Applying the preprocessing and segmentation process

The pre-processing technique plays an important role in converting data into suitable and organized form. It helps to precise the pattern discovery and to provide accurate and appropriate information for data mining intent. In our case, we apply the data cleaning by deleting the requests of sounds and pictures' extension, keeping the request related to error status as well as deleting the request who has no parameter with success status.

Below an example of an extracted row that displays the structure to be segmented as presented later.

212.40.149.49 - - [03/Jan/2018:12:52:11 -0700] "GET /ccne.php HTTP/1.1" 200 6772 "http://www.iut.ul.edu.lb/tuition-fees.php" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36"

Indeed, this tool offers the ability to pick a log file shown in figure [29]. Then, it will be processed by passing it through a regular expression which will segment the log file line by line and insert it into a data frame using the proposed pattern below:

(?P<Host>[\d\.]+)\s+

(?P<Identity>\S+)\s+

(?P<User>\S+)\s+

\[(?P<Time>.+?)\]\s+

"(?P<Request>.+?)"\s+

(?P<Status>\d+)\s+

(?P<Bytes>\d+|-)\s+

(?:"(?P<Referrer>.*?)"|\n)\s*

(?:"(?P<useragent>.*?)"|\n)

Therefore, After applying the pattern above, we added the formatted data into pandas data frame which is a two-dimensional size-mutable to produce a csv to be segmented as shown in the figure [30].

Host	Identity	User	Time	Request	Status	Byte	Referrer	User Agent
212.40.149.49	-	-	an/2018:12:52:11 -	"GET /ccne.php HTTP/1.1"	200	6722	ttp://www.iut.ul.edu.lb/tuition-fees.ph	"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36"

Figure 30: Example of results after the segmentation process

iii. Segmentation of the URL

Log file contains a part of the full request and response related to the HTTP protocol. The client asks the essential web server to access the desired resource and the response will be registered as a status code. It allows to attain the requests for instance "GET /ccne.php HTTP/1.1" in the above figure [30]. The GET

request realizes the fundamental requirement of serving the requested URL to the client as a query string. In addition, the URL will be extracted using dynamic step. In this stage, we applied a right split to extract the protocol HTTP/1.1 as well as to examine the left split to select the "GET" method. Thus, we can attain the extracted URL as a query string like "/ccne.php". Finally, our tool will check and analyze this URL according to our proposed attack patterns stated in the next section.

iv. Pattern discovery and attack detection process

Pattern discovery benefits from the preprocessing results. It offers some techniques such as statistical analysis, pattern analysis, association rules, clustering and other techniques. In our tool, we applied the pattern analysis technique to discover each detected pattern with its behavior. Therefore, we constructed our proposed data base as an ajax file. It includes more than 100 patterns related to several attack types such as Sql injection (SQLi), Cross-Site Scripting (XSS) and Directory Traversal (DT). Some of the proposed patterns in our data base can be shown in table [11]. Afterwards, the pattern discovery was executed, we examined the proposed patterns using the regular expression with the ability to identify if the segmented URL is matched with our proposed attack pattern to be detected as attack or not. Thus, we achieved the detection process by identifying the most important attacks that can face any web server.

Attack Type	Attack Behavior	Proposed Pattern
SQLi	SQL authentication bypass attempts	?:union\\s*(?:all|distinct|[(!@]*)\\s*[([]*\\s*select)|(?:\\w+\\s+like\\s+\\\")|(?:like\\s*\"\\%)|(?:\"\\s*like\\W*[\"\\d])|(?:\"\\s*(?:n?and|x?or|not |\\\|\\\||\\&\\&)\\s+[\\s\\w]+=\\s*\\w+\\s*having)|(?:\"\\s**\\s*\\w+\\W+\")|(?:\"\\s*[^?\\w\\s=.,;)(]+\\s*[(@\"]*\\s*\\w+\\W+\\w)|(?:select\\s*[\\[\\]()\\s\\w\\.,\"-]+from)|(?:find_in_set\\s*\\()
XSS	XSS Probings	?:,\\s*(?:alert|showmodaldialog|eval)\\s*,)|(?::\\s*eval\\s*[^\\s])|([^:\\s\\w,.\\/?+-]\\s*)?(?<![a-z\\/_@])(\\s*return\\s*)?(?:(?:document\\s*\\.)?(?:.+\\/)?(?:alert|eval|msgbox|showmod(?:al|eless)dialog|showhelp|prompt|write(?:ln)?|confirm|dialog|open))\\s*(?:[^.a-z\\s\\-]|(?:\\s*[^\\s\\w,.@\\/+-]))|(?:java[\\s\\/]*\\.[\\s\\/]*lang)|(?:\\w\\s*=\\s*new\\s+\\w+)|(?:&\\s*\\w+\\s*\\)[^,])|(?:\\+[\\W\\d]*new\\s+\\w+[\\W\\d]*\\+)|(?:document\\.\\w
DT	Specific directory and path traversal	?:%c0%ae\\/)|(?:(?:\\/|\\\\)(home|conf|usr|etc|proc|opt|s?bin|local|dev|tmp|kern|[br]oot|sys|system|windows|winnt|program|%[a-z_-]{3,}%)(?:\\/|\\\\))|(?:(?:\\/|\\\\)inetpub|localstart\\.asp|boot\\.ini

Table 11: Some of the attack patterns extracted from our configured data base

v. Pattern analysis

This stage in the web usage mining known as the output visualization process. It generates as output graphs, table and report in order to display the discovered pattern and the detected behavior.

6.3 Experimental Results and Analysis

In this part, we will display our experimental results for each tool as stated in the following parts.

6.3.1 The Deep Log Analyzer Tool Result

The results of this tool display the detection of our visitors' behavior. In fact, our interest in this experiment is to mention the reports concerning the detected activities and the most important resources. Therefore, the outputs are suggested in three different categories; the general activity, visitors' activities and the access resources control illustrated in figure [31].

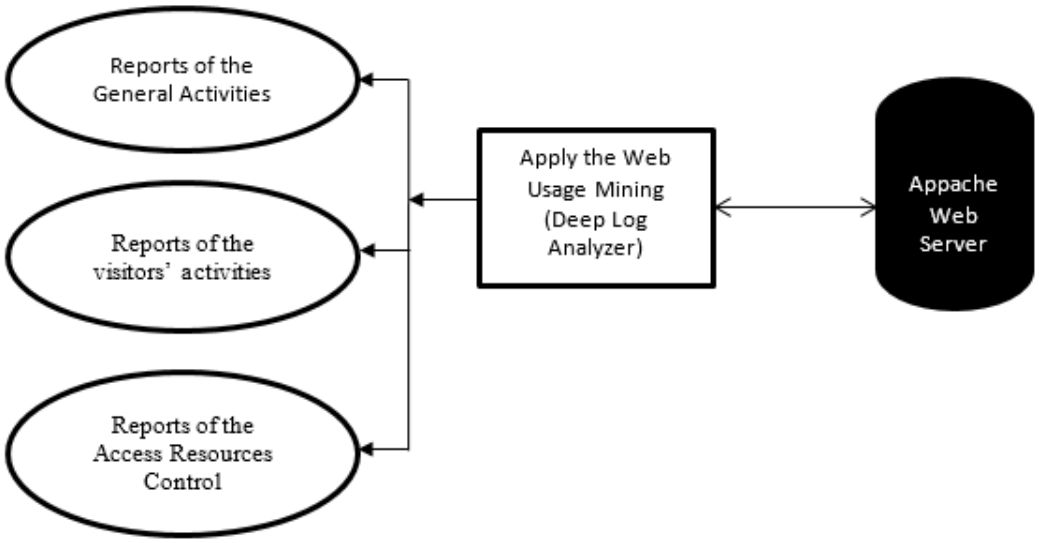


Figure 31: The achieved results using the Deep Log Analyzer tool

The results of this tool shows a report of general activities that detected several information related to the visitors during a selected date. As shown in the figure [32], several summaries have been extracted about the hits, visits, visitors and the page views. In the hits summary, the tool detected 318 hits including 84% that are identified as successful hits. The number of visits were 51 with an average of 13 visits during 7:44 min per day. In addition, the number of unique visitors is 44, 89% from them visited the web once with an average of 1.16 for the number of visits per visitor. Lebanon country attains an average of 36% and was selected as the most visited country to our website. According to the page views summary, during that period the total of viewed pages is 204 where the log in this page shown as the most popular visited page.

Hits Summary [Details...]	Total	Per Day
Number of Hits:	318	80
Number of Successful Hits:	266 (84%)	67

Visits Summary	Total
Number of Visits :	51
Average Number of Visits per Day:	13
Average Visit Duration :	7:44 Min

Visitors Summary	Total
Number of Unique visitors :	44
Visitors who visited once:	39 (89%)
Repeat visitors:	5 (11%)
Average Visits per visitor :	1.16
Most visitors from this Country :	Lebanon (36% visitors)

Page Views Summary		Hits
Total Page Views :		204
Most popular Page :	.../wp-login.php??goto=99999...	15
Most popular Download :	.../33adbf46c4fd78664e915ffa...	9

Figure 32: A summary about the visitor's general activity

In addition, by analyzing the report shown in figure [33], we attained a technical overview reports. The most popular browser used to reach our website is the Mozilla and the "Android OS" detected as the top used operating system. A percentage of 16 % was occurred concerning the error hits that happened during the visits.

Technical Summary	
Most Popular Browser :	Mozilla or other Mozilla based 5.0
Most Popular Operating System :	Android
Error Hits :	52 (16%)

Figure 33: A summary for the technical summary result

Furthermore, by studying the report shown in figure [34], we can precise the duration of the spending time of the visitors. 88% from our visitors were spending just less than of 5 minutes and it can vary from day to another at the same week.

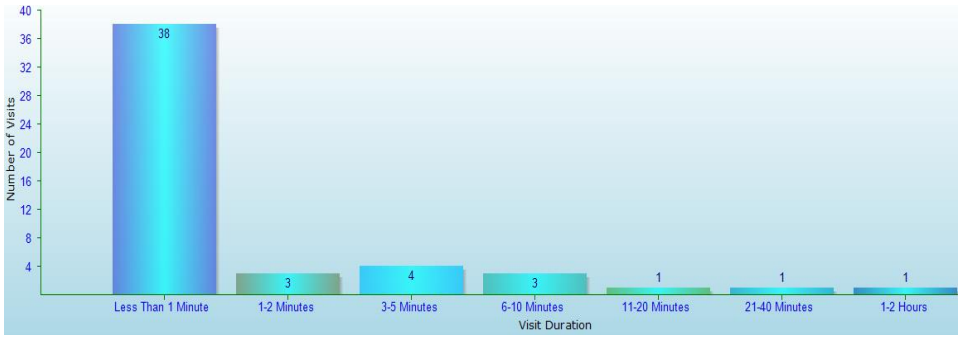


Figure 34: Visitor's spending time on the faculty website

Moreover, by tracking the web resources, we extracted statistical reports about our web server resources. Thus, we detected the behavior of the visitors and their interest with respect to the pages, files, diagnostic and resources errors. Thus, we can monitor the popularity downloaded files with an example shown in the figure [35] to detect the main notice of the visitors in terms of those files.

FileName
/subCat/33adbf46c4fd78664e915ffa2045c1b5.pdf
/schedules/5d08a3612465501c05e787e07f41501f.pdf

Figure 35: Some of the top downloaded files

In addition, as shown in the figure [36], we noticed that the root of our web server "/" achieved the top visited link with 180 hits. Several accessed directories can be used to identify the most visited resources or files located in any directory. The results of this report is ranked by the number of hits and the transferred data.

Directory	Number of Hits	Data Transferred (Kb)
/	180	738
/css/	27	101
/ccne/	19	213
/img/	17	138
/subCat/	9	4,243
/img/news/	7	590
/css/fonts/	7	0
/cap/Update/	6	0
/img/events/	5	394
/schedules/	4	269
/js/	4	130
/img/slider/	4	5
/ccne/plugins/iCheck/square/	3	3
/ccne/plugins/bootstrap-wysihtml5/	3	166
/ccne/plugins/iCheck/	3	1
/ccne/plugins/iCheck/flat/	3	3
/ccne/plugins/iCheck/futurico/	3	1
/ccne/plugins/iCheck/line/	3	5
/ccne/plugins/iCheck/polaris/	3	1
/css/themes/	3	3
/ccne/plugins/iCheck/minimal/	3	3
/ccne/c:/	2	0
Total	318	7,007

Figure 36: A summary about the top accessed directories

Further, our tool achieved the target of controlling the accessed links that permits to detect some of the occurred errors during the visits. For instance, after analyzing the report in figure [37], we can conclude that "404" is the most happening error with 52 hits during that period. Therefore, this study will help the web system administrator to identify if there are vulnerabilities within the web server. Moreover, we can observe the targeted pages to determine the best solution to fix the discovered errors.

Error	Number of Hits
404 - File Not Found	52

Page	Number of Errors
/wp-login.php??goto=999999.9+%2f**%2fuNiOn%2f**%2faLl+%2f**%2fsElEcT(%2f**%2fsElEcT+%2f**%2fcOnCaT(0x217e21,count(t.%2f**%2ftAbLe_nAmE),0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftAbLeS+as+t+on+t.%2f**%	15
/favicon.ico	8
/css/ajax-loader.gif	6
/wp-login.php??goto=999999.9+%2f**%2fuNiOn%2f**%2faLl+%2f**%2f%2f**%2%2f**%2%2f**%2sElEcT(%2f**%2fsElEcT+%2f**%2fcOnCaT(0x217e21,count(t.%2f**%2ftAbLe_nAmE),0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftA	5
/wp-login.php??goto=999999.9+%2f**%2%2f**%2%2f**%2fuNiOn%2f**%2faLl+%2f**%2fsElEcT(%2f**%2fsElEcT+%2f**%2fcOnCaT(0x217e21,count(t.%2f**%2ftAbLe_nAmE),0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftAbLeS+as	5
/css/fonts/slick.woff	4
/css/fonts/slick.ttf	3
/ccne/c:/arbre%20de%20Haffman	1
/ccne/c:/Abre%20de%20Huffman	1
/apple-touch-icon-precomposed.png	1
/apple-touch-icon-120x120-precomposed.png	1
/apple-touch-icon-120x120.png	1
/apple-touch-icon.png	1
Total	52
Total	52

Figure 37: Results about the occurred web server errors

6.3.2 The Security Analysis Tool Result

The results of the security analysis tool reveal the detection process as illustrated in the figure [38]. Our tool achieved the ability to detect three kinds of attacks which are the SQL injection, XSS and path traversal that faced our faculty web site.

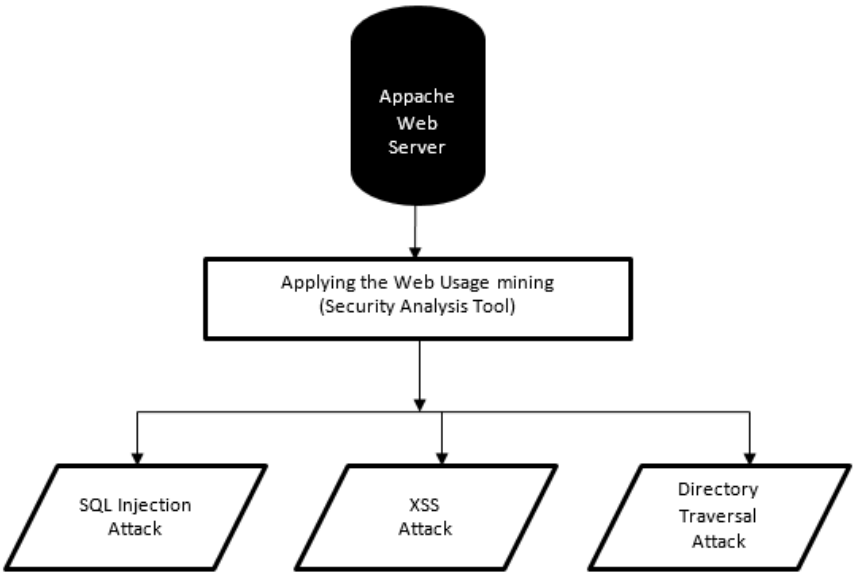


Figure 38: The security analysis tool with our achieved results

We can observe figure [39] after executing our security analysis tool based on our proposed attack patterns that presents a summary of results about the discovered attacks with their types per each visited link. In addition, the results show that our tool proves a cyber-security mechanism. It classified the detected attack were its behavior can be identified as malicious URLs determined as SQL injection, Cross-Site Scripting and Directory Traversal attacks. Ultimately, our proposed tool detected among 15 malicious URLs; 10 attacks have been classified as SQL injection, 3 attacks for XSS and 2 related to the path traversal attacks.

Access Pattern	Attack Type
/wp-login.php??goto=999999.9+%funNiOn%f**%2fal+%2f**%2fsEIEcT(%2f**%2fsElEcT+%2f**%2fcOnCaT(0x217e21,count(t.%2f**%2ftAbLe_nAmE),0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftAbLeS+as+t+on+t.%2f**%	Sql injection
/??goto=SELECT%20CHAR(0x66)	Sql injection
/??goto=10%3B%20DROP%20TABLE%20members%20%2F*	Sql injection
/??goto=ASCII()	Sql injection
/??goto=10%3B%20DROP%20TABLE%20members%20--	Sql injection
/wp-login.php??goto=999999.9+%2f**%2funiOn%2f**%2faLI+%2f**%2f%2f**%2%2f**%2%2f**%2sEIEcT(%2f**%2fsEiEcT+%2f**%2fcOnCaT(0x217e21.count(t.%2f**%2ftAble_nAme).0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftA	Sql injection
/wp-login.php??goto=999999.9+%2f**%2%2f**%2%2f**%2fuNiOn%2f**%2faLI+%2f**%2fsEIEcT(%2f**%2fsEIEcT+%2fcOnCaT(0x217e21.count(t.%2f**%2ftAble_nAmE).0x217e21)+%2f**%2ffRoM+information_schema.%2f**%2fsChEmAtA+as+d+join+information_schema.%2f**%2ftAbLeS+as	Sql injection
/schedule.php?file=/etc/	path traversal
/schedule.php?javascript%3Aalert%28%27%27%29	XSS
/schedule.php?javascrip:alert(")	XSS
/ccne/adminPFE.php?Login='%20and%20'1'='1~~~&Password='%20and%20'1'='1~~~&ret_page='%20or%20'1'='1~~~&querystring='%20%2B%20(SELECT%20FieldName%20FROM%20TableName%20LIMIT%201.1)%20%2B%20'~~~&FormAction=login&FormName=Login	Sql injection
/index.php?Login='%20and%20'1'='1~~~&Password='%20and%20'1'='1~~~&ret_page='%20and%20'1'='1~~~&querystring=';%20EXEC%20master..sp_makewebtask%20"\\10.10.1.3\share\output.html"".%20';%20SELECT%20*%20FROM%20INFORMATION_SCHEMA.TABLES""~~~&FormAction=login&20SELECT%20*%20FROM%20INFORMATION_SCHEMA>TABLES""~~~&FormAction=login&F	Sql injection
/schedule.php?file=%3Cscript%3Ealert%28%27%27%29%3C%2Fscript%3E	XSS
/schedule.php?file=/etc/shadow	path traversal
/ccne.php?goto=AND%20%20%27a%27%3D%27a%27	Sql injection
TOTAL	**15**

Figure 39: The results of the security analysis tool with the detected cyber-attacks

Chapter 7: Host based Intrusion Detection System based on Text Mining and Machine Learning

7.1 General Overview

In our days, the information technology within the cyber space appears in every aspect of our lives. The methods of protecting networks, systems and sensitive information are known as cyber-security service which can be built on the HTTP/HTTPS protocol. Jose Barahona da Fonseca et al. reveal that among the various web attacks, the code injection on web pages increases each year and conduct up to 96.15% of web attacks in last few years [5]. In addition, according to the WAAR which concentrates on the most common types of injection attacks [6], the Cross-Site Scripting (XSS) attack counts 49.09% of the whole web attacks, SQLI attack counts 28.32% of the whole web attacks and finally Path Traversal counts 9.82% of the whole web attacks (DT) [7]. Furthermore, the user inputs acted as an injection method for the web application attacks. Thus, these inputs appear in query string of HTTP GET request. Depending on this assumption, the malicious query can be considered as one of the core of web injection attacks [8]. Therefore, cyber scientists present various techniques in order to detect malicious activities. Hence, these activities are discovered as queries in the web requests using both of signature and anomaly based detection systems [9].

In this chapter, we will implement a host intrusion detection system based on the text mining technique. Our dataset has been selected in a way it matches our web server log file. It contains 6000 textual records of SQLi, XSS and path-traversal related to the query string of HTTP GET URLs. We will also discuss the difficulties that may face these data source such as suffering from the complicated features method. We have applied four models using the machine learning techniques; the Decision Tree, Multilayer Perceptron (MLP), Support Vector Machine and KNeighbors. Finally, we will validate the applied models, these methods will offer the classification in detecting the SQL injection, XSS and the path traversal attacks as well as exporting for each model its performance measurements.

7.2 The Proposed HIDS Architecture

In this section, we will state the process of developing the host based intrusion detection system. The data will be chosen in a way it matches the log file that was extracted from our university web server. In addition, a preprocessing technique with the proposed feature representation method will be performed. Each raw of the text will be converted into vector of features by adding all unique words to a dictionary. Thus, each word in the dictionary becomes a feature in the vector to represent the input text. Therefore, we developed our IDSs to realize the attacks as a detection process using several machine learning techniques shown in the figure below [40]. Every process in the development phase will be discussed in the following parts.

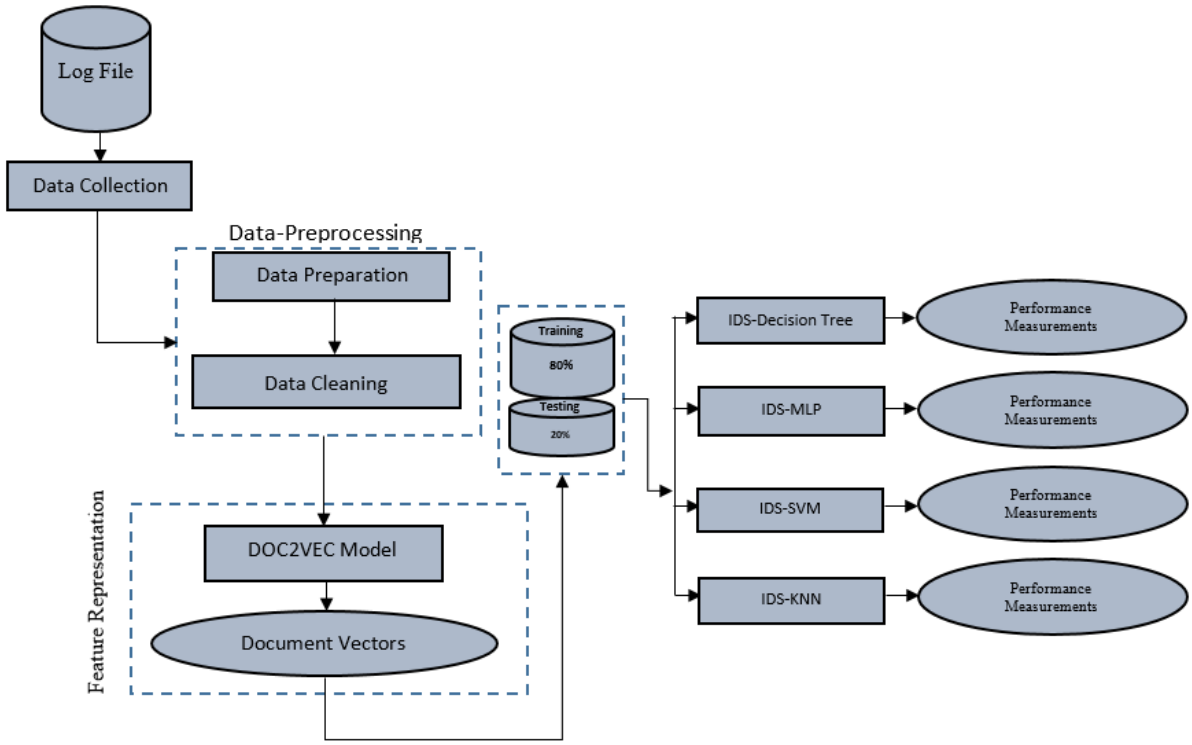


Figure 40: The Proposed HIDS architecture with the implementation phase

7.2.1 Data Collection

Log files are extended by almost all real-world web servers. Thus, having these log files provide us with a great advantage for our interpretation. Thus, in our study we used the apache log file presented in CLF format as shown in figure [41]. This data source offers security-related information. Each log line is a combination of static and dynamic information which consists of date and time information, user information, event information, and application information. The used log files were gathered from vulnerable sites which include several resources and consist of both dynamic and static web pages. These sites were hosted on the faculty of technology in the Lebanese based web Server-Linux platform. Therefore, we performed large quantity of different attacks during one week on the web server using the faculty web system via manual and dynamic injection methods. First we did some tasks by scanning the vulnerabilities using OpenVas tool. Moreover, we executed potential attacks using developed shell scripts by using penetration testing tool for web applications and online vulnerability reports such as US-CERT and CERT/CC Advisories. In addition, we extracted valuable information from several security-related mailing lists. We also used automatic attacking tools such as Pangolin, Havij in order to perform SQLI attacks, LOIC, and XSS proxy for initiating attack string and legal requests. Moreover, these log files were combined with an open source log files and were extracted from a testbed [10] which is a set of web applications that are vulnerable to SQL injection attacks and several log files from various sources [11].

Common Log Format (CLF)

The configuration of the common log format is given below [11]

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

The entries give details about the client who had requested for the web site to the web server

- 127.0.0.1 (%h) - This is the IP address of the client which made the request to the server.

- (%l) - The hyphen present in the log file entry next to the IP address indicates that the requested information is not available.

-frank (%u) - The user id of the person requesting the document as determined by HTTP Authentication

- [10/Oct/2000:13:55:36 -0700] (%t) -The time format resembles like [day/month/year: hour: minute: second zone]

-"GET /apache_pb.gif HTTP/1.0" (\"%r\") - The request sent from the client is given in double quotes. GET is the method used. apache_pb.gif is the information requested by the client. The protocol used by the client is given as HTTP/1.0

- 200 (%>s) - This is the status code sent by the server. The codes beginning with 2 for successful response, 3 for redirection, 4 for error caused by the client, 5 for error in the server

-*"http://www.example.com/start.html" (\"%{Referer}i\")* - This gives the site that the client reports having been referred from. (This should be the page that links to or includes /apache_pb.gif).

- *"Mozilla/4.08 [en] (Win98; I ;Nav)" (\"%{User-agent}i\")* - This is the information that the client browser reports about itself to the server.

Figure: 41 The log file of CLF format

7.2.2 Data Preprocessing

User navigation on a web site is represented as a set of lines in web server log files, some of these lines do not reveal any useful information. They are unnecessary for the analysis process and could cause noise in this stage, thus it affects the performance during the attack detection. Therefore, preprocessing stage should be applied before carrying any learning algorithms on the data source. Unfortunately, many studies in this field did not mention the preprocessing steps [12]. They just present the implicitly of the log parse process that is responsible for converting the log file into a specific format. In our study, we discuss this issue. Figure [44] and Table [12] illustrate the steps that were involved in the preprocessing process. It includes log segmentation and data cleaning on the generated format. The following subsection will be mentioned and clarified in the next steps.

7.2.2.1 Data Preparation

Due to the fact that log file is unstructured text, we need to parse the log entries into structured representation. The retrieval of user navigation pattern becomes possible to train a learning model over this formal representation [13]. The previous log analysis methods neglected timestamp log entry values and use the static log key to detect anomaly values [14]. In our study, all the entries found in the log file will be stored for the purpose of applying the following:

- Create CSV file named dataset where the columns number and names in this file are equal to the log entries presented in the collected log files.
- Apply regular expressions on the collected log files to group each log entry values into a separate group. Each group's name is the same as the header obtained from log files.
- Map the outputted group after applying regular expressions to their associated columns in the CSV file.

After this stage, we applied further preprocessing techniques presented later in the outputted CSV file as shown in figure [42] in order to compare among different learning algorithms.

host	time	request	status	bytes	user_agent
92.184.100.14	31/Dec/2017:05:13:57	GET /css/fonts/slick.woff HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:57	GET /css/fonts/slick.ttf HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:57	GET /css/ajax-loader.gif HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:58	GET /favicon.ico HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:14:34	GET /css/ajax-loader.gif HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
54.36.150.183	31/Dec/2017:05:14:35	GET /robots.txt HTTP/1.1	404	-	Mozilla/5.0 (compatible; AhrefsBot/5.2; +http://ahrefs.com/robot/)
92.184.100.14	31/Dec/2017:05:14:47	GET /css/fonts/slick.woff HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:14:47	GET /css/fonts/slick.ttf HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
207.46.13.205	31/Dec/2017:05:17:18	GET /robots.txt HTTP/1.1	404	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)

Figure 42: An example about the data during the preprocessing stage

7.2.2.2 Data Cleaning

The data cleaning process is about altering the storage resource in which the created CSV file is from the previous stage, in this way we make sure that we capture the needed data only. In this stage, first we started by decoding the HTTP GET request into ASCII characters to transform it to lowercase and to remove numerical values. In addition, a new technique was suggested to discard HTTP GET requests shown as follows:

- Returning the raw with status code that have the number 200 with no query string
- Cleaning the static request such as (eg.'jpg','png','gif','webp','cr2','tif' and others)

Next, the remaining HTTP GET requests were parsed to extract a query from the URL that appears after the "?" mark for instance "GET /ccne.php?id=2 HTTP/1." Finally, duplicated text should be showed as one entry in our dataset since different queries may appear to be identical after applying the previous cleaning techniques. The data collected from this step as well as the resources will be used to generate a new dataset shown in figure [43] that consists of the three types of attacks {XSS, SQLI, Path-Traversal}.

payload	attack_type
/../../../../../../../../../../../../etc/passwd	path-traversal
/etc/passwd	path-traversal
file:/etc/passwd	path-traversal
.....etcpasswd	path-traversal
file:.....etcpasswd	path-traversal
..../..../..../..../..../..../..../..../..../..../.	path-traversal
/..../..../..../..../..../..../..../..../..../..../	path-traversal
//../..../..../..../..../..../..../..../..../..../.	path-traversal
1' where 6406=6406;select count(*) from rdb$f	sqli
1) and 8514=(select count(*) from domain.dom	sqli
-3136%') or 3400=6002	sqli
1) where 7956=7956 or sleep(5)#	sqli
-7387'))) order by 1--	sqli
1')) as gfzb where 7904=7904;begin dbms_lock.	sqli
1))) union all select null,null,null#	sqli
confirm(2)>/	xss
open()>	xss
location=?javascript:alert(1)>click	xss
<svg><script>alert(/1/)</script>	xss
</script><script>alert(1)</script>	xss
5rt(0);'>rhainfosec	xss

Figure 43: Part of the generated dataset including the three kinds of attacks

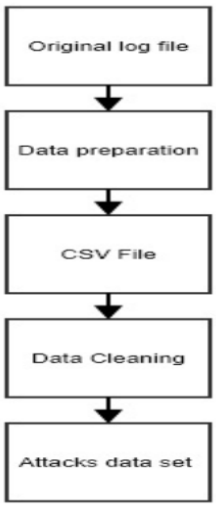


Figure 44: A log file with preprocessing steps

Data Pre-processing	
Data Cleaning	- The requests of sounds and pictures extension should be deleted - Keeping the request related to error series status - Deleting the request who has no parameter with success status
After preprocessing	Attack will be generated and used as an input to the learning algorithms

Table 12: Applying the preprocessing process

7.3 Feature Representation Method

In this section we will discuss the process of learning the word and documents vector from the text, this is known as feature extraction. Feature representation on textual data is an active research area in which

many different architecture models are proposed in natural language processing (NLP) to extract information from context and represent the extracted knowledge by a vector. These features are used by the set of classifiers to be trained upon, after which a meta-learning technique is introduced to specify which of the pre-trained classifier is reliable during the training process.

We used the NLP technique for proposing our feature method that is based on paragraph vector by using the word vector approaches for the HTTP GET Request. In addition, this approach was fulfilled when the word vectors pass through a task for predicting the following word in a sentence (malicious URLs). Then, although the word vectors initially are in arbitrary state, the word vectors can realize the semantics in a form of indirect result during the prediction task.

In figure [45] we can observe the framework for the paragraph vector in an SQL injection example. Each paragraph has a different vector characterized by a matrix "D" as a column and each word (from the query string) has a different vector characterized by matrix "W" as a column. After that, the process of averaging or concatenating was performed for word and paragraph vector to apply the word prediction related to the context. The paragraph symbol can be determined as a different word. We can relate that to a memory which depends on the paragraph's topic or the current context in remembering the omitted ones [15].

The stochastic gradient descent is one of the main factors that is used in the training stage for both paragraph and word vectors. The gradient will be gained using the backpropagation method. In each phase of the stochastic gradient training, the fixed length context will be sampled via a random paragraph to compute the gradient error that may occur in a selected network. Thus, it can be used for dynamically updating the parameters in the prediction model. Indeed, it is a necessity at any textual prediction process. An important step known as the inference which is employed to compute paragraph vector for a new paragraph, this can be performed via the gradient descent factor. In addition, the word vectors (W) and the Soft Max weights which are used in the prediction task must be fixed as parameters in the model.

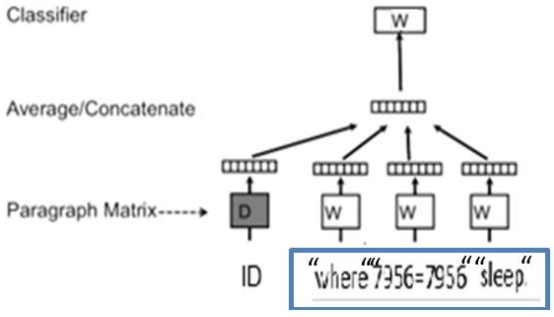


Figure 45: An example for the proposed Doc2vec model [15]

Assume that we have N as the total number of paragraphs "the extracted URL" and M as the number of words "malicious code" in the vocabulary. Our aim is to learn paragraph vectors in a way that every paragraph is mapped to P dimensions and every word is mapped to Q dimensions. After that the model will contains a total of (N × P + M × Q) parameters (excluding SoftMax parameters [16]). Although when N

increases, the parameter's number will also increase. While we performed the updating in the training phase, the result will be effective.

When the training phase is accomplished, we can represent the features in the paragraph using the paragraph vectors. Lately, the features can be added to machine learning techniques such as decision tree, logistic regression, support vector machines or KNN etc.

This method should have two key stages:

- A stage of getting word vectors W by the training phase as well as the SoftMax weights and paragraph vectors D based on already known paragraphs.
- A stage of getting Paragraph Vectors D by the inference step in order to predict new paragraphs that were never seen before. This stage should be done by appending a lot of columns and gradient descent in D.

To accomplish the prediction process, we use D for several specific labels based on machine learning prediction classifier. The advantage of applying this method is that the model can learn using unlabeled-data, in addition this can provide prediction tasks in case there is no sufficient labeled data type.

We apply this method on the generated attack data set discussed previously to convert the URL attack from text to numerical vectors. Due to the fact that machine learning can be applied on numerical data, this step is essential in order to classify HTTP GET requests into different attacks type using machine learning algorithms.

7.3.1 DOC2VEC Model

In this section we present the methodology for building doc2vec for detecting attacks on HTTP GET request. Our goal is to learn the embedding that gets the properties about the order in which the words appear in a request. To achieve that, we first identify doc2vec hyper-parameters that should be fed to this model which are:

- Dm=1: To define a training algorithm like 'distributed memory' (PV-DM).
- size=300: Feature vectors Dimension.
- Window=10: The maximum distance between the current and predicted word within a sentence.
- Alpha=0.025: The initial learning rate.
- min_alpha=0.025: Learning rate will linearly drop to min_alpha as training progresses.
- min_count =5: For ignoring all the words with total frequency lower than this.
- Sample=10e-5: The threshold for configuring which higher-frequency words are randomly down sampled, the useful range is (0, 1e-5).
- Negative=5: Negative sampling will be used, the value for negative specifies how many "noise words" should be drawn (usually between 5-20).

- dm_concat=1: The usage of concatenation of context vectors rather than sum/average.
- Document= Training dataset generated previously.

After we applied this model in our dataset, we obtained the 300−dimensional vector representation for each document along with a dictionary (word embedding) that contains all unique words found in the training corpus of the dataset. Table [13] shows the number of document vectors and unique word that were obtained from this model.

Word embedding	**Number of document vectors**
1172	6000

Table 13: Word embedding and document vectors numbers resulted from the doc2vec model

Furthermore, in this section we visualized the embedding documents of HTTP GET request extracted from the generated dataset. We used the whole data set of 6000 requests taken from our dataset which consist of balanced distribution classes that have been selected from our data as presented by table [14]. We extracted the feature vectors of these requests from matrix D that was generated when we trained doc2Vec model (trained on 6000 requests). The selected feature vectors were obtained by concatenating of the word vectors outputted from the words appeared in the same document. We obtained a 300-dimensional vector for each document in the training corpus and for each unique word collected from dataset during the training process. For the baseline features, the distributed memory in doc2vec embedding the use of CBOW while generating the word vectors from the dataset. In addition, from the extracted feature vectors, we applied t-SNE [17] to reduce the feature dimensions and plot the HTTP GET request on a 2-dimensional embedding space. The representation of the embedded requests can be seen in figure [47].

XSS	**SQLI**	**Path-Traversal**
1999	2003	1998

Table 14: Number of document vectors for each class

As it can be seen in figure [46], for requests, the XSS, SQLI and Path-Traversal requests are clearly separated into three groups of requests. Most of SQLI requests are located in the right area of the plot while XSS request are in the central area and Path-Traversal are in the left area of the plot. Very few data points of different attack requests are overlapping with each other. Also, we observed that several clusters appeared in the plot. We further analyzed some of these clusters to identify the potential patterns in the desired string which may possibly be indicative for the nature of this attack. Analysis of such patterns could be useful for a deeper understanding in the HTTP GET request properties. For instances, requests that contain phrases such as 'select *', 'where' in the query string, or 'from', were clustered together. For phrases that appeared like '<script>', they were grouped in another cluster. We were still able to distinguish between two clusters of requests; one for '<script>' in the query string and the other is for

'select *' in the query string. Therefore, the distinction between different text during training was clear. On the other hand, the employing of our features method, doc2vec proves and obtains meaningful representation and embeds those requests in sparse matrix where similar vector request get clustered together. Finally, with the chosen number of dimensions and without the need to obtain expert features, feature vector is efficient to represent the requests. For data size of 6000 requests doc2vec model does not suffered from the memory constraints to process and store the vectors which can be used for further downstream tasks.

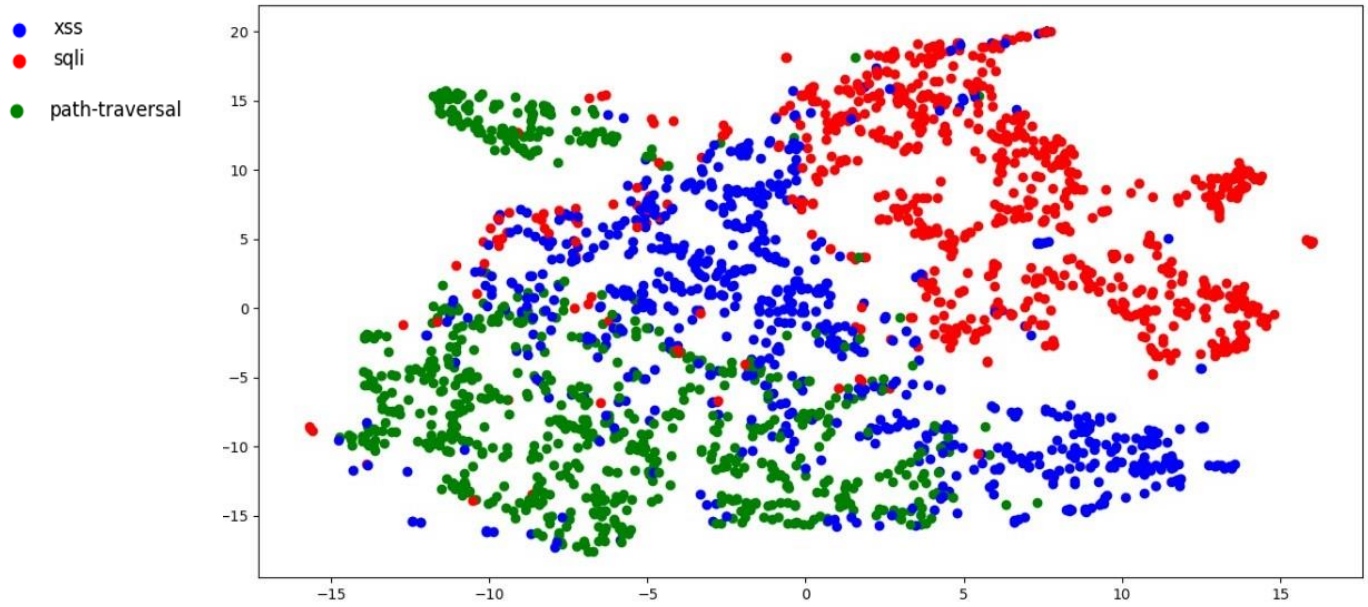


Figure 46: Document vector space plot

7.4 The Applied Machine Learning Methods and Classification Preliminaries

The aim of this part is to find the preferable machine learning methods to be applied on our data set in order to develop our proposed IDSs to detect the most popular web server attacks.

In our approach, we employed the documents-embedding technique (doc2Vec model) by extracting 6000 documents vectors learned by doc2vec. Furthermore, we used four different algorithms (Decision Tree, Multilayer Perceptron (MLP), Support Vector Machine and KNeighbors Classifier to classify the Extracted URLs presented in HTTP GET request that permits us to detect these malicious attacks. Therefore, to evaluate the applied models, we started by splitting the data set into two parts; 80 % for training and 20 % for the testing proportion as presented in table [15]. Moreover, the intelligent models shown subsequently were tested with numerous estimations metrics such as the accuracy, precision specificity, recall, area under curve (ROC) and confusion matrix. The metrics were generated for each machine learning classifier which include important evidence about the existing and detecting attack classes.

	Training data set	**Testing data set**

Created Data set	80%	20%

Table 15: The suggested splitting proportion for the training and testing stage

Furthermore, during the learning stage and in each model, we applied several performing parameters for the machine learning classifier stated as follows.

i. Decision Tree Model:

A decision tree is similar to a tree structure which composed of nodes that shapes a rooted tree. The basic aim of ID3 algorithm is to build a decision tree by using flowchart starting from top to down. It tests each parameter at any node like the entropy for exploratory analysis. This property will separate the training phase taking into consideration their target classification. To achieve that, the entropy has been tested with the formula:

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Where i is the number of all classes and pi is the probability of class I. In this model we selected the parameters below:

- Criterion= Entropy
- Splitter=Best
- Max_depth=None
- Class_weight= Balanced
- Max_features= None

ii. MLP Model:

Artificial neural network (ANN) is considered as one of the latest technology in the field of artificial intelligence (AI). The Neural network can treat any information in a similar way like the human brain can process. The network is collected from a huge number of processed interconnected neurons that work in a parallel way. It may solve many scientific problems to take artificial decisions. Thus, we used the Multi-Layer Perceptron (MLP) in the ANN which is a network that have numerous layers. Those layers play several roles and functionalities. The network includes the input layer, hidden layer and network output that is called the output layer. In addition, mathematically the MLP classifier shown in figure [47] is calculated according to the Rectified Linear Unit activation function:

"Relu(x) = max(x,0)"

Where x is an element and the Relu(x) is the function that attains the maximum of that element with zero. Therefore, our model was examined with the word vectors as an input layer, hidden layer and the output layer according to the following parameters:

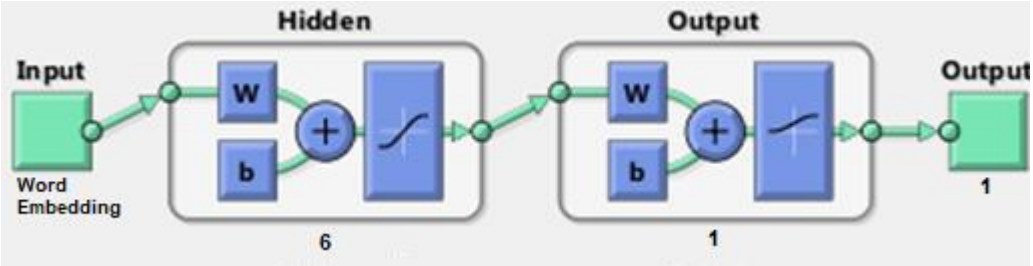


Figure 47: Artificial neural network: the MLP model

- Activation=Relu as the rectified linear unit function
- Solver=adam as a solver for weight optimization
- Hidden_layer_sizes =6 which is number of neurons in hidden layer
- Learning rate= 0.001 schedule for weight updates
- random_state= batch sampling to precise the random number for weights and bias initialization
- Momentum =0.9 as a gradient descent update
- Alpha=0.0001 for avoiding overfitting by penalizing weights
- Max_iter =1000 which is the Maximum number of iterations; the solver iterates until convergence by this number of iterations.

iii. SVM Model:

The Support Vector Machine classifier (SVM) was introduced by Vladimir Vapnik in 1998. It is a supervised learning algorithm. It works based on the separation by a margin of linearly separable classes, then it generalizes it into nonlinear boundaries by changing the space.

In this model we computed the hyperplane separators (decision boundaries) by the formula: $h(x) = x^T\beta + \beta_0 = 0$. Then we calculated the distance from a point to the hyperplane by using the following formula:

$\mathrm{d(x)} = \frac{|x^T\beta+\beta_0|}{||\beta||}$ knowing that $||\beta|| = \sqrt{\beta_1^2 + \cdots + \beta_p^2}$ maximizing the margin amounts to minimize the norm of the vector of parameters β.

The SVM can be employed in two approaches which are the classification and regression problems. In our model we used during the training and testing phases the following parameters:

- Kernel= linear: with the equation: $k(x_i, x_j) = x_i . x_j$
- Tol=0.001 Tolerance for stopping criterion: 1e-3
- gamma='scale' that uses 1 / (n_features * X.var()) as a value of gamma

- cache_size=200 as size of the kernel cache
- max_iter=- 1 for no limit.

iv. KNN Model:

In this model we applied a supervised learning which is the KNN technique that is used for classification problems. It works by assumption of the data points of identical classes that are closer to each other. We applied the following steps to detect the attack classes:

- Step 1: K value should be determined
- Step 2: We calculated the distances between the points of the testing and training data using the metric of Minkowski
- Step 3: We sorted the distances to set the k nearest neighbors according to the lowest distance values
- Step 4: We analyzed the neighbors to assign the related category for new data (test) using the majority vote
- Step 5: We attained the detected class.

In our approach, we tested our trained model with the following parameters:

- Weight= uniform the used function in the detection process
- Metric= minkowski with p=2 which is the standard Euclidean metric
- n_neighbors=2 Number of neighbors to use
- algorithm='auto' the Algorithm used to compute the nearest neighbors
- n_jobs=1 the number of parallel jobs that permits to search for neighbors

In the next section we will begin the discussion of the results of each model to make a comparative study Thus, we will prove the best accurate and effective model among the chosen ML techniques.

7.5 Experimental Results and Discussion

In this part, we will evaluate the performance of our proposed HIDS using four machine learning techniques: the Decision Tree, Multilayer Perceptron (MLP), Support Vector Machine and K-Neighbors Classifier. Our target is to classify the most malicious attacks that may face the web server of our faculty. Therefore, to evaluate the applied HIDSs, we examined different estimations metrics such as the detection rate, accuracy, precision, recall, area under curve (ROC) and confusion matrix. We noticed that on each time we employ the proposed algorithm, the generated results of each model are clearly different.

Thus, the table [16] illustrates the detection rate for each attack that were tested using our proposed IDSs. We can conclude that the SVM classifier accomplished the highest rate by 95.41% in detecting the Path-Traversal attack. The MLP classifier achieved the highest detection rate for both SQLI and XSS stated respectively by 94.69%, 84%. From another perspective, SVM achieved highest detection rate for path traversal where it reached the lowest rate in detecting the XSS with 63.47%.

		Classifier			
		Decision Tree	**MLP**	**SVM**	**KNN**
Attack Type	Path-Traversal	0.8214	0.9260	0.9541	0.9184
	SQLI	0.9215	0.9469	0.8776	0.9330
	XSS	0.8320	0.8400	0.6347	0.7840

Table 16: The highest detection rate achieved by the classifiers for each attack

Figures 48,49,50,51 show the area under the Receiver Operating Characteristic (ROC) for each applied model which were calculated based on true positive and false positive. The large value of ROC shows the ability of a model to detect the malicious while the lower value presents the weakness of that model. Regarding the area under the ROC of each model, we can conclude that MLP classifiers logged the highest rate 93% while KNN recorded with 91%, decision tree (90%) and SVM presented the lowest value (87%).

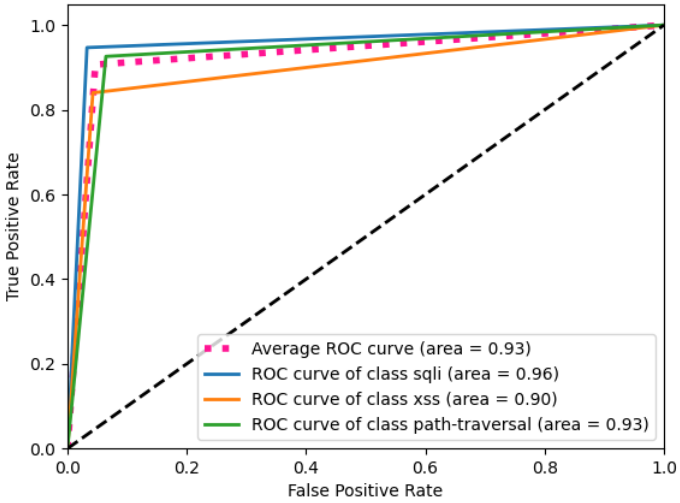


Figure 48: MLP ROC curve

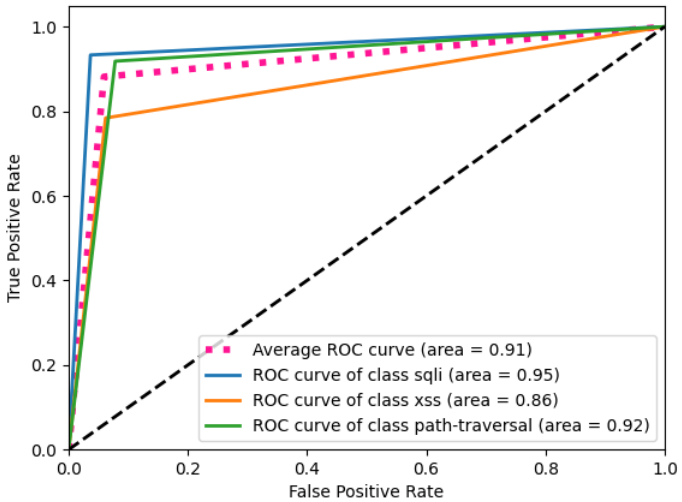


Figure 49: KNN ROC curve

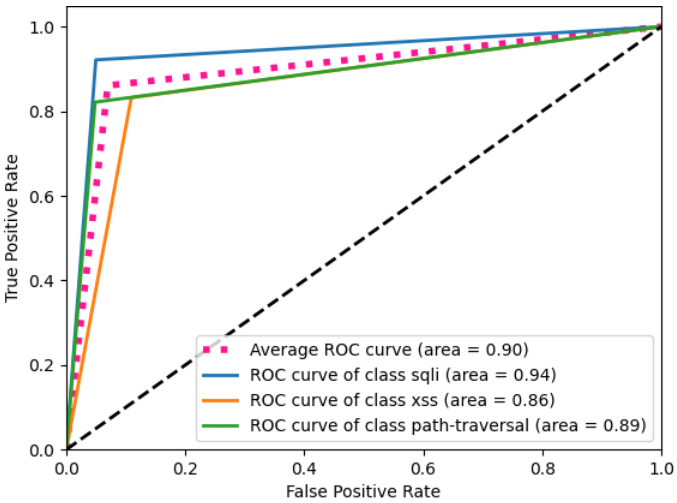


Figure 50: Decision Tree ROC curve

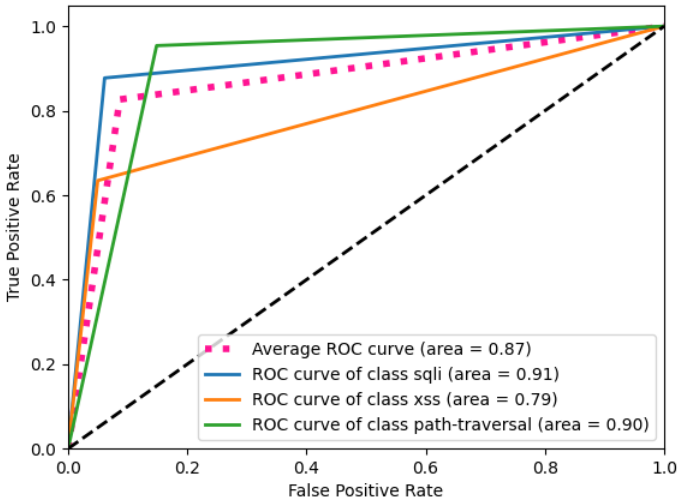


Figure 51: SVM ROC curve

Among the validation of 6000 records existing in our dataset. The average accuracy rate for each selected classifier are presented in table [17]. We can conclude that the MLP classifier is the most accurate model with an accuracy of 90.67% while the SVM is the lowest with 82.67%.

Machine Learning Classifiers	**Accuracy**
MLP	0.9067
KNN	0.8817
Decision Tree	0.8608
SVM	0.8267

Table 17: Accuracy results of the different ML Techniques

By using the decision tree model and after analyzing the table [18] and the figure [52], we found that the decision tree model recorded average values in testing data. The path traversal attack had a precision rate of 89.20%, 82.14% as a recall rate, the SQLI takes the highest precision score with 91.30%, 92.15% for the recall rate and finally the XSS attains the lowest precision with 77.61% while the recall attains the 83.20%. In addition, the decision tree model recorded a precision average of 86.04 % and the area under ROC achieves the average of 90%.

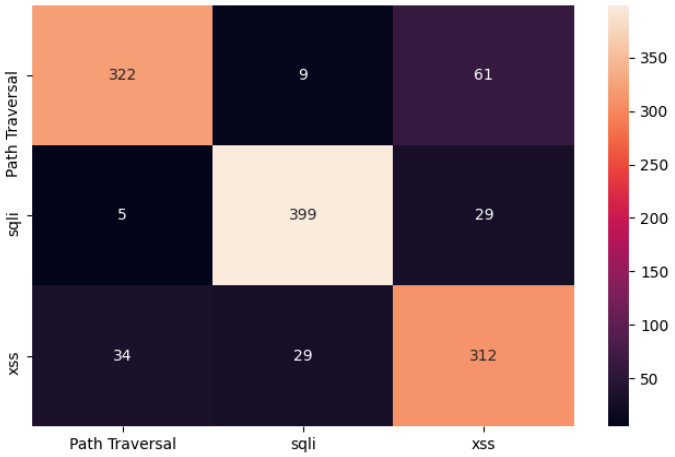
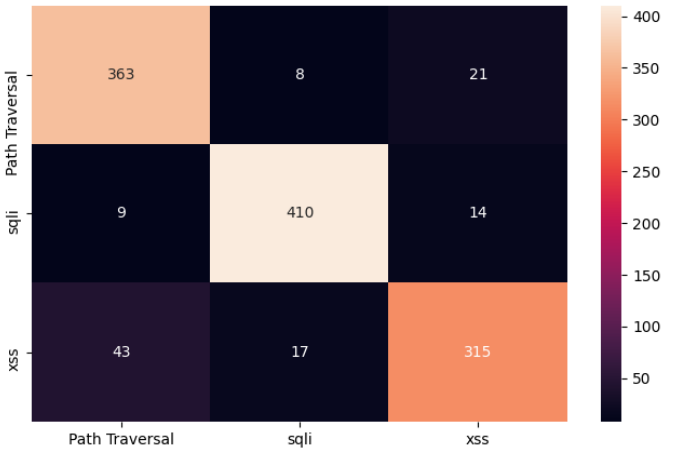


Figure 52: Confusion matrix for the Decision Tree model

	Precision	Recall	
Path Traversal	0.8920	0.8214	
SQLi	0.9130	0.9215	
XSS	0.7761	0.8320	
Accuracy			0.8608
Macro avg	0.8604	0.8583	
Weighted avg	0.8634	0.8608	

Table 18: Classification report for the Decision Tree model

In addition, according to the MLP model and after analyzing the table [19] and figure [53], we concluded that this model achieved enhancement values in the testing phase where the path traversal attack had a low precision rate (87.47%), the SQLI takes the highest precision score with (94.25%) and finally the XSS achieves a rate of 90%. The recall results for the path traversal, SQLI and XSS attacks were 92.60%, 94.69% and 84% sequentially. This model recorded an increase in the precision average value in the detection phase 90.57% and similarly the area under ROC obtains the average of 93%.



	Precision	Recall	
Path Traversal	0.8747	0.9260	
SQLi	0.9425	0.9469	
XSS	0.9000	0.8400	
Accuracy			0.9067
Macro avg	0.9057	0.9043	
Weighted avg	0.9071	0.9067	

Table 19: Classification report for the MLP model

Figure 53: Confusion matrix for the MLP model

Furthermore, we found that the SVM model has been increased in some classes in the testing phase after studying the table [20] and figure [54]. For instance, the path traversal attack had a low precision rate of 75.71%, the SQLI achieves the precision score of 88.99% and the XSS is about 85.30%. The recall results for the path traversal, SQLI and XSS attacks were 95.41%, 87.76% and 63.47% respectively. Moreover, this model recorded 83.34 % as an average precision value and 87% as an average of the area under ROC rate.

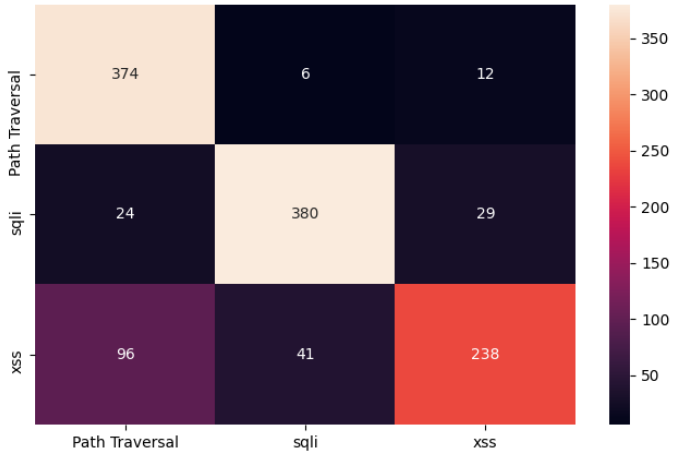


Figure 54: Confusion matrix for the SVM model

	Precision	Recall	
Path Traversal	0.7571	0.9541	
SQLi	0.8899	0.8776	
XSS	0.8530	0.6347	
Accuracy			0.8267
Macro avg	0.8334	0.8221	
Weighted avg	0.8350	0.8267	

Table 20: Classification report for the SVM model

Finally, after analyzing the table [21] and figure [55], we concluded that the KNN model achieved normal values in testing data, the path traversal attack had a low precision rate (85.11%). The SQLI attains a high precision score of 93.52% while the XSS attack takes 85.22%. The recall results for the path traversal, SQLI and XSS attacks were 91.84%, 93.30% and 78.40% sequentially. This model recorded an average of 87.95% as a precision value in the detection phase and the area under ROC obtains the average of 91%.

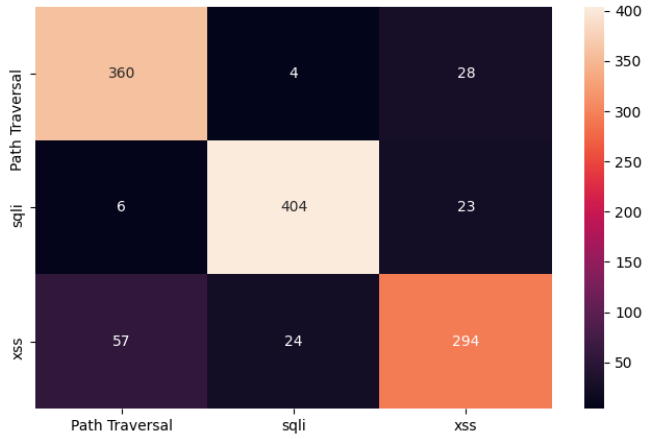


Figure 55: Confusion matrix for the KNN model

	Precision	Recall	
Path Traversal	0.8511	0.9184	
SQLi	0.9352	0.9330	
XSS	0.8522	0.7840	
Accuracy			0.8817
Macro avg	0.8795	0.8785	
Weighted avg	0.8818	0.8817	

Table 21: Classification report for the KNN model

Conclusion Part II

We presented three chapters to reveal the different security assessment studies that were carried out at the faculty of technology in the lebanese university.

In chapter five, we proposed a mechanism for security enhancement for the system of the department of computer communications and networks engineering in the faculty of technology. Therefore, we applied the penetration testing technique according to the top 10 OWASP. This technique permits us to discover the vulnerabilities concerning the most popular attacks such as SQL injection (SQLi), cross-site scripting (XSS) and sensitive data exposure. We deliberated security solutions and the suggestions that can be used as a guide line by the IT administrator to protect the system against cybercriminal threats. Thus, we guarantee the efficiency of our system by fixing all the detected vulnerabilities to achieve the essential safety standards.

In the chapter six we presented a methodology to detect the visitor's behavior based on the web mining techniques. Our target has been accomplished by developing and applying our suggested tools; the deep log analyzer and the security analysis model. These tools enhanced the efficiency to identify several approaches about the visitor activities and the abnormal actions. Hence, the deep log analyzer used to identify the approaches related to the visitors' activities, their behaviors, access resources control, errors of hits and others. On the other hand, the security analysis tool was employed to detect several malicious URLs where 15 attacks which discovered stated severally; 10 of SQLi, 3 for XSS and 2 related to the path traversal.

Moreover, in the chapter seven, we developed a host based intrusion detection system (HIDS) using the text mining technique. We constructed a textual dataset which includes 6000 records of malicious URLs related to the HTTP GET request. We proved that there is a complicated hypothesis including the features representation methods. It has three major limitations; weakness in retrieving information about the URL, manual work is needed to extract the most important features and inability to capture useful information about the unseen HTTP GET request. To overcome these limitations, we suggested the Doc2Vec model as a feature representation method in our HIDS. Furthermore, four different machine learning techniques (KNN, SVM, Decision Tree and MLP) were employed to offer the best effective classification model. Our HIDS achieved the ability to detect the SQLi, XSS and directory traversal attacks. Hence, MLP recorded the best accuracy of 90.67% Subsequently, the KNN attained the second accuracy score with 88.17 % followed by the decision tree of 86.08 %. Finally, the SVM retrieved the lowest accuracy rate of 82.67%.

References Part II

[1] S. Singh, S. Silakari. (2013). "An Ensemble Approach for Cyber Attack Detection System: A Generic Framework", 14th ACIS, IEEE 2013. pp 79-85.

[2] OWASP. (2017)." Top Ten Web Application Security Risks". Retrieved from https://owasp.org/www-project-top-ten/.

[3] AK. KASSEM, B. DAYA, P. CHAUVET. (2018). "A Proposed Methodology on Predicting Visitor's Behavior Based on Web Mining Technique", (IJACSA) International Journal of Advanced Computer Science and Applications, No. 12, Volume 9.

[4] Deep Software (2018). "Deep Log Analyzer", Available from https://www.deep-software.com/.

[5] J. Fonseca, M. Vieira, H. Madeira. (2013). "Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection", IEEE Transactions on Dependable & Secure Computing, 440–453, Volume 11.

[6] Imperva. (2015). "Web Application Attack Report WAAR ". Retrieved from https://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed6.pdf.

[7] I. Hydara, ABM. Sultan, H. Zulzalil, N. Admodisastro. (2015). "Current State of Research on Cross-site Scripting (XSS) - A Systematic Literature Review", Information and Software Technology, 170–186, Volume 58.

[8] Y. Donga, Y. Zhanga. (2017). "Adaptively Detecting Malicious Queries in Web Attacks", ArXiv, Volume 1701.0777.

[9] AK. Kassem, SA. Arkoub, B. Daya, P. Chauvet. (2019). "A Survey of Methods for the Construction of an Intrusion Detection System", Artificial Intelligence and Applied Mathematics in Engineering Problems. ICAIAME 2019. Lecture Notes on Data Engineering and Communications Technologies. Springer, Cham, pp. 211-225. Volume 43.

[10] BCF. (2020). "SQL-Injection-Application-Testbed". Retrieved from http://www.bcf.usc.edu/~halfond/testbed.html, University of Southern California.

[11] Elastic. (2019). Retrieved from https://raw.githubusercontent.com/ elastic/examples/master /Common%20Data%20Formats/apache_logs/apache_logs.

[12] S. Singh, S Silakari. (2013). "An Ensemble Approach for Cyber Attack Detection System: A Generic Framework", 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 79–84.

[13] JG. Lou, Q. Fu, S. Yang, Y. Xu, J. Li. (2010). "Mining Invariants from Console Logs for System Problem Detection", USENIX Annual Technical Conference, 24–24.

[14] X. Yu, P. Joshi, J. Xu, G. Jin, H. Zhang, G. Jiang. (2016). "CloudSeer: Workow Monitoring of Cloud Infrastructures via Interleaved Logs", ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS).

[15] Q. Le, T. Mikolov. (2014). "Distributed Representations of Sentences and Documents", The 31st International Conference on International Conference on Machine Learning, Volume 32, 88–96

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. (2013). "Distributed Representations of Words and Phrases and their Compositionality", ArXiv, Volume 1310.4546.

[17]L. Van der Maaten, G. Hinton. (2008). "Visualizing Data using t-SNE", Journal of Machine Learning Research 9, Volume 2579-2605.

Part 3: Security Intelligent System Based-Intrusion Detection using Machine Learning Techniques (SIS-ID)

Introduction 118

Chapter 8: Materials and Development Mechanism 120

8.1 General Overview 120

8.2 The SIS-ID Requirements 120

8.2.1 Data Gathering: Canadian Institute for Cyber-Security Datasets 120

8.2.1.1 DB-MALCURL Dataset Preparation 121

8.2.1.2 DB-DDOS Dataset Preparation 121

8.3 Data and Features Engineering 122

8.3.1 Data Preprocessing 122

8.3.2 Features' Technique 123

8.3.3 Selected Features 124

8.4 The Learning Methodology for SIS-ID System 125

8.4.1 The Applied Machine Learning Methods 126

8.4.2 Learning Implementation 130

8.4.3 Learning Optimization Method 131

Chapter 9: Results: SIS-ID Performance Evaluation 132

9.1 General Overview 132

9.2 Experimental Results and Discussion 132

9.2.1 Applying the SIS-ID System on DB-MALCURL 132

9.2.1.1 Supervised Learning 132

9.2.1.2 Ensemble Techniques 137

9.2.1.3 Evolving the Ensemble Techniques 140

9.2.2 Applying the SIS-ID system on DB-DDOS 144

9.2.2.1 Supervised Learning 144

9.2.2.2 Ensemble Techniques 148

9.2.2.3 Evolving the Ensemble Techniques 152

9.2.2.4 Unsupervised Learning 156

9.3 General Discussion and Evaluation 157

9.4 Hardware-Based Real-Time Simulation 160

Conclusion Part III 163

References Part III 164

Introduction

Résumé

Cette partie, intitulée système intelligent de sécurité pour la détection des intrusions basé sur des techniques de l'apprentissage automatique est composée des deux chapitres. Dans cette partie, le système « SIS-ID », que nous avons développé, a été décrit. Ce système est basé sur les techniques d'apprentissage automatique. Dans le premier chapitre, nous avons suggéré tous les matériaux utilisés pour détecter les attaques d'URL malveillantes et les DDOS. Par conséquent, nous avons configuré nos ensembles de données, recueillies auprès de l'Institut canadien de la cybersécurité (CIC), les DB-MALCURL et DB-DDOS. De plus, nous avons proposé notre méthode d'ingénierie des données employées pendant la phase d'apprentissage du système. Nous avons utilisé l'étape " preprocessing" pour le nettoyage des données, le " under-sampling" et la technique de "data transformation". De plus, la méthode de caractéristiques a été sélectionnée en utilisant la technique « recursive feature elimination ». Ensuite, nous avons appliqué onze techniques d'apprentissage automatique ; l'apprentissage supervisé, l'apprentissage non supervisé et les techniques d'ensemble. Finalement, nous avons proposé une méthode d'optimisation de l'apprentissage qui utilise les techniques "Hyperparameter et GridSearchCV" avec la validation croisée K-Fold. Ceci a notamment permis une évolution des mesures de performance du SIS-ID. Dans le deuxième chapitre, nous avons présenté et analysé les résultats de notre système. Ainsi, le SIS-ID avec la méthodologie appliquée sur les méthodes d'ensemble offre les meilleurs résultats. Il a atteint la précision (98.57%), rappel (98.55%), F1-mesure (98.56%) et accuracy (98.52%) en utilisant le modèle « voting » via le DB-MALCURL. De plus, comparant au plus bas résultat obtenu en utilisant l'"arbre de décision", ce modèle a permis une amélioration de la performance de 3 % (précision), de 2,86 % (rappel), de 2,93 % (f1-mesure) et de 2,97 % (accuracy). D'autre part, en examinant le DB-DDOS, et en appliquant la technique « stacking », notre système SIS-ID a atteint des meilleurs résultats : précision (79.77%), rappel (77.07%), f1-mesure (76.28%) et accuracy (77.04%). De cette façon, SIS-ID a permis une amélioration de la performance comparant au plus bas résultat obtenu par le modèle « KNN » comme suit: précision (de 6,26 %), rappel (de 4,55) %, f1-mesure (de 4,29%) et accuracy (de 4,56%). Par conséquent, nous avons conclu que notre méthodologie, basée sur des modèles d'ensemble avec les méthodes d'optimisation suggérée, a amélioré les performances du SIS-ID pour détecter les attaques d'URL malveillantes et les DDOS. En outre, avec notre système, nous avons obtenu des performances plus élevées que celles présentés par le laboratoire CIC en termes de détection d'attaques à la fois par URL malveillantes et DDOS. Finalement, nous avons validé notre modèle SIS-ID en l'implémentant dans un matériel intelligent au sein de l'université libanaise en tant que système de prévention des intrusions. Nous avons testé le système en utilisant le model « facteur de valeur aberrante locale (LOF) ». Ainsi, il a atteint une bonne précision avec l'efficacité d'éviter une attaque par déni de service (DOS) effectuée sur une scène en temps réel.

Overview

In this part, we will present the general architecture of our security intelligent system based on the machine learning techniques called "SIS-ID". In fact, our system will be conducted for detecting both of malicious URL and DDOS attacks. Therefore, we will present our configured datasets that were gathered from the Canadian Institute for Cyber-Security. Moreover, we will discuss our data engineering and features' method employed during the learning stage. Further, we will sate the applied machine learning techniques with our suggested learning optimization method for enhancing the performance of each model. Consequently, we will display the performance measurement of our proposed SIS-ID for detecting the malicious URLs using DB-MALCURL and the DDOS attacks based on the DB-DDOS. Ultimately, the validation of this system will be implemented as an intelligent hardware and simulated on real time stage with the efficiency for avoiding a performed DOS attack.

Chapter 8: Materials and Development Mechanism

8.1 General Overview

Currently, the growing of network threats presents as a sophisticated challenge against the defense mechanism in the networks for the protection of privacy data and its components. Therefore, security intelligent systems will be recommended to deploy it via the machine learning techniques against ever-increasing cyber threats. The intrusion detection systems (IDSs) and Intrusion Prevention Systems (IPSs) are considered as one of the most important considerations of cyber-researchers due to its potency in detecting and preventing any novel cyber-attacks in real-labeled network against different set of attacks and abnormal activities.

In this chapter, we will introduce the development mechanism of our proposed IDS named as "SIS-ID" concerning the used materials in detecting both of malicious URL and DDOS attacks. We will present the datasets gathered from the Canadian Institute for Cyber-Security by preparing all the required techniques to apply our planned data engineering and features method that can be employed during the learning stage. Moreover, we will state the applied machine learning techniques that were used on our SIS-ID system as well as we will propose the learning and optimization methods during the training phase.

8.2 The SIS-ID Requirements

In this part, the materials and development mechanism are suggested in order to develop the SIS-ID system based on the machine learning techniques. Therefore, the data gathering, feature and data engineering, proposed learning and optimization techniques are stated in the following parts.

8.2.1 Data Gathering: Canadian Institute for Cyber-Security Datasets

The appraisal of input cyber-datasets takes an essential consideration in the effectiveness of any intrusion detection system approach. It allows us to evaluate the suggested techniques that is qualified in detecting and preventing cyber-attacks. The preparation of datasets for network based IDS may be not facilely obtainable because of the privacy and confidentiality matters. Moreover, the biggest challenge is the lack of publicly, availability and labeled datasets that will be used in the training phase that is based on machine learning techniques. In this part we will highlight on the selection of reliable datasets that were decided to be used in the development of SIS-ID gathered from the Canadian Institute for Cybersecurity (CIC) in the University of New Brunswick which is an inclusive multidisciplinary that specialized in cyber security filed.

In our thesis, our SIS-ID system was conducted by referring to the datasets extracted from the CIC organization selected and named as follows: DB-MALCURL using the ISCX-URL-2016 dataset [1] and DB-DDOS using the DDOS2019 dataset [2].

8.2.1.1 DB-MALCURL Dataset Preparation

The cyber space has become an essential source for web malicious activities. In fact, the URLs are utilized as one of the main techniques in this field. The security scientists put their efforts on developing intelligent techniques for collecting malicious URLs such as Alexa top websites, WEBSPAM-UK2007, Open Phish and many other resources [1].

The DB-MALCURL was prepared using the ISCX-URL-2016 dataset to implement our SIS-ID system for detecting the most important malicious URLs. This Data base dataset covers 80 features extracted from the CICFlowMeter framework [3] which is a network traffic flow generator and analyzer used to generate biflows from the pcap files as well as extracting features from these flows. DB-MALCURL contains five classes of attacks which are the defacement, malware, phishing, benign and spam distributed as shown in the table [22].

DB-MALCURL	
Attack Type	**Number of rows**
Benign	7530
Spam	7735
Phishing	7945
Malware	6670
Defacement	6820

Table 22: The distribution of the classes for DB-MALCURL

8.2.1.2 DB-DDOS Dataset Preparation

Distributed Denial of Service (DDoS) attack is considered as one of the most threatened attacks for network security field. It aims to exhaust networks with large amount of malignant traffic. Although several tools have been intended in detecting the DDoS attack, the weak computational performance is currently considered as essential concerns for the cyber security-scientists to develop intrusion detection systems based on the machine learning techniques that heavily relies on reliable dataset. Therefore, the DB-DDOS has been prepared using the CICDDoS2019 dataset to implement our SIS-ID models for detecting the DDOS attacks. This dataset covers a collection of the latest DDOS network flows' attack listed in the table [23] with 80 features extracted from the CICFlowMeter-V3 framework that contains 13 DDOS attack classes proposed in the distribution as shown in following table.

DB-DDOS	
Attack Type	**Number of rows**
BENIGN	55665
DrDoS_DNS	55975
DrDoS_LDAP	55875
DrDoS_MSSQL	56050
DrDoS_NetBIOS	55635
DrDoS_NTP	56325
DrDoS_SNMP	56310
DrDoS_SSDP	56625

DrDoS_UDP	55930
Syn	55910
TFTP	55890
UDP-lag	56220
WebDDoS	55590

Table 23: The distribution of the classes for DB-DDOS

8.3 Data and Features Engineering

In this part, we will present the data and projected features' engineering methods used in the development of our SIS-ID system. Indeed, several factors can affect the results of a machine learning algorithm. Therefore, data and features engineering were employed in the implementation phase to improve the performance of intelligent model based ML as stated in the following parts.

8.3.1 Data Preprocessing

Data preprocessing method is a critical phase in machine learning that intended to enhance the data quality to improve the extraction of well insights from desired data. This phase prepares the instances by treating it to be in a proper form. Thus, in this part our pre-processing step were examined in several techniques concerning the transformation of subsets into readable, comprehensible and cleaned format since the data was gathered from the sources and may contains noisy, empty or irrelevant data that could potentially decrease the performance of our SIS-ID system. The Data preprocessing steps are specified in figure [56] and listed as follows:

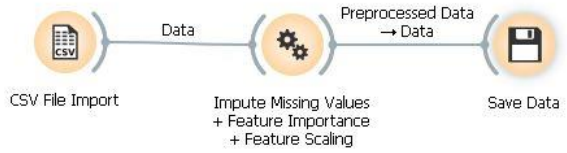


Figure 56: Data Preprocessing Workflow

i. The data Cleaning step in order to remove and handle missing values by replacing the missing value with meaningful ones.
ii. Balancing the subset by using the under-sampling and over-sampling methods clarified as seen below.

Imbalanced data typically refers to the dataset where the classes are having a big difference in size. Several datasets have an unbalanced number of instances in their classes. To overcome this problem, the sampling method was applied using the scikit-learn library [4] with two main algorithms stated as follows:

A. Over-sampling: duplicating samples from the under-represented classes using "imblearn.over_sampling import RandomOverSampler"
B. Under-sampling: removing samples from the over-represented classes using "imblearn.under_sampling import RandomUnderSampler".

iii. Data Transformation by encoding the categorical features as well as scaling them

Generally, most of IDSs data set includes features with different volume and range. Indeed, dataset includes features with several dimensions and range. The features with high magnitudes can be measured with the Euclidean distance between two data points. Therefore, the scaling methods among the data should be applied to overcome this issue and they must be brought to the same scale of magnitudes [5]. Our proposed scaling method has been applied for the SIS-ID system using the "StandardScaler" method. It standardizes the features by eliminating the mean and scaling it to unit variance where the standard score of a sample x calculated according to this formula "Z= (X - U) /S"; where U is the mean of the trained instance(x) and S is considered as the standard deviation of x($\frac{xi-mean(x)}{stdev(x)}$) as outputted for instance in table [24].

	Querylength	domain_token_count	path_token_count	avgdomaintokenlen	avgpathtokenlen	charcompvowels
1	-0.232407712	0.158097788	-0.223720996	2.148138384	0.28908439	0.151258571
2	-0.030905003	0.158097788	0.010092389	-0.537537372	-0.0203342	0.076355921
3	-0.232407712	-0.946569866	0.243905774	1.308864569	0.651546168	0.151258571
4	0.000911215	1.262765441	0.010092389	-0.705392185	-0.668639807	0.151258571
5	-0.211196901	1.262765441	-0.691347767	-0.453610091	-0.815982004	-0.672670581
6	-0.232407712	0.158097788	-1.392787923	-0.537537372	-1.14013479	-1.047183831
7	-0.179380684	1.262765441	-0.457534382	-0.831283232	0.156476423	-0.223254679
8	-0.168775278	-0.946569866	1.880599472	0.301736192	0.321499683	1.574408925
9	-0.232407712	-0.946569866	-1.392787923	1.560646663	2.749698778	-0.897378531

Table 24: The example about the scaling method result

8.3.2 Features' Technique

In this part, we proposed the feature selection method to be employed in our SIS-ID system to improve the outcome performance. Therefore, we suggested the feature ranking based recursive feature elimination that used the cross validation method "feature_selection.RFECV" with the parameters to attain the best accrued features using the following parameters:

- estimator=DecisionTreeClassifier
- max_depth=20
- step=1
- cv=StratifiedKFold(5)
- scoring=accuracy

8.3.3 Selected Features

Our recommended features technique assigns an importance score or weight to each selected feature in the SIS-ID system. Thus, the features with the lowest importance score and variance will be eliminated from the data and the model will be trained according to that technique shown as below:

i. DB-MALCURL Dataset

Below is our features selection that were suggested on the DB-MALCURL using the recursive feature elimination based on the cross validation (RFECV). In this dataset, our proposed technique selected 39 features stated as follow:

- 'Querylength',
- 'domain_token_count',
- 'path_token_count',
- 'avgdomaintokenlen',
- 'avgpathtokenlen',
- 'charcompvowels',
- 'ldl_domain',
- 'ldl_filename',
- 'dld_url',
- 'domainlength',
- 'fileNameLen',
- 'this.fileExtLen',
- 'pathurlRatio',
- 'ArgUrlRatio',
- 'executable',
- 'NumberofDotsinURL',
- 'CharacterContinuityRate',
- 'host_DigitCount',
- 'Directory_DigitCount',
- 'File_name_DigitCount',
- 'Directory_LetterCount',
- 'Filename_LetterCount',
- 'Domain_LongestWordLength',
- 'Path_LongestWordLength',
- 'sub-Directory_LongestWordLength',
- 'Arguments_LongestWordLength',
- 'URLQueries_variable',
- 'spcharUrl',

- 'delimeter_Domain',
- 'delimeter_path',
- 'NumberRate_URL',
- 'NumberRate_DirectoryName',
- 'NumberRate_FileName',
- 'NumberRate_Extension',
- 'NumberRate_AfterPath',
- 'SymbolCount_URL',
- 'Entropy_URL',
- 'Entropy_Domain',
- 'Entropy_Extension',

ii. DB-DDOS Dataset

Below we can observe our features selection that were employed on the DB-DDOS using the recursive feature elimination based on the cross validation (RFECV). In this dataset, our proposed technique selected 7 features stated as follow:

- ' Flow Duration',
- ' Total Length of Fwd Packets ',
- ' Fwd Packet Length Max',
- ' Fwd Header Length',
- ' act_data_pkt_fwd',
- ' min_seg_size_forward',
- ' Inbound',

8.4 The Learning Methodology for SIS-ID System

The proposed SIS-ID is an intelligent system developed using python language. It was deployed to detect the latest cyber-attacks using the BDD- MALCURL and DB-DDOS that were utilized in the learning phase for our implemented system. The learning method will be stated in the next part using 80% of each data source for training stage which is definitely takes the biggest part that can be used to train or fit the model to find the optimal parameters that affect it. However, the 20% of remaining data were used for the testing stage that are needed to test the SIS-ID for an unbiased evaluation while this set will be permanently excluded from the training instances to compare their results with the actual classes. Indeed, to achieve a high detection rate, the proposed SIS-ID uses the training and testing sets as indicated in the following methodology illustrated in figure [57]:

- We break down the whole of each dataset into five subsets (20% for each instance, we choose one of them for the testing stage and the remaining four instances for the training one.

- One of our selected machine learning algorithms should be selected.
- We apply our learning technique to produce a result for the selected model.
- This process should be repeated with five times as we attain each time another new subset for the testing and the remaining for training.
- The average of the five results (performance measurements) are calculated and obtained for the selected model.
- This process will be repeated for all other models.
- By applying this methodology, we will minimize the side effects of the instances distribution within the two main sets (testing and training), since this distribution can be considered as arbitrary and will produce for each time different results.

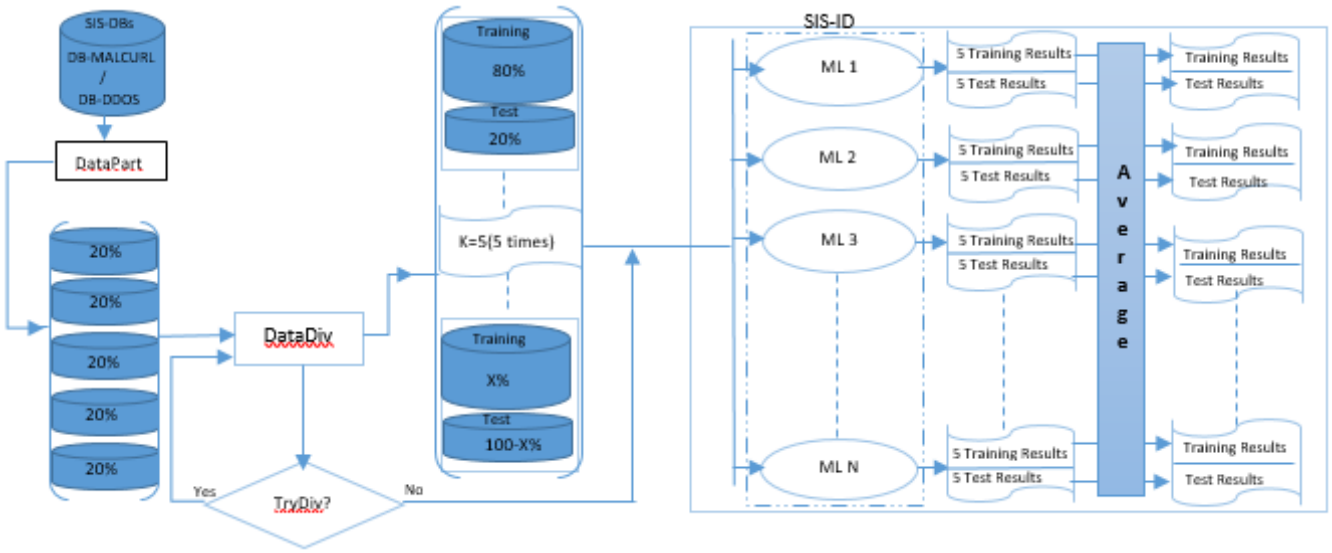


Figure 57: General architecture of the SIS-ID learning methodology based on the applied machine learning techniques

8.4.1 The Applied Machine Learning Methods

We will describe below the learning methods in proposing the SIS-ID system using several machine learning techniques that were tested on the datasets described above which are BDD- MALCURL and DB-DDOS. Thus, our machine learning models have been selected based on the supervised learning using the k-nearest neighbors (KNN), Decision Tree and two Multi-classes techniques which are the OneVsRest and OneVsOne models. We also evolved the ensemble techniques using five models such as Voting, Stacking, Bagging, XGBoost, Random Forest and Adaboost based on a defined methodology that was suggested to optimize our system which is applied on each dataset as stated below:

i. Supervised Learning

In this part, our proposed SIS-ID were examined with the supervised learning which provides powerful models to classify the attacks using four classifiers in order to prove our detection; the Decision Tree, KNN, OneVsRest and OneVsOne.

A. Decision Tree

Decision Tree is a category of trees used frequently in machine learning for the data classification problem represented in attributes and values. The objective of this technique comes from generating of a hierarchical sequence of tests, as short as possible, which successively divides the set of training data into disjoint subsets. Therefore, our model has been trained with the selected optimal parameters using the GridSearchCV function as follow:

- class_weight: none
- criterion: gini
- max_depth: 23
- splitter: best
- Max_features= None

B. KNN

The k-nearest neighbor's algorithm is a supervised learning algorithm which classifies an instance following a plurality vote for its related neighbors. The target is to be assigned to a common class among its k-nearest neighbors. Generally, this method is based on similarity of characteristics and the proximity of them, the samples will be classified in relation with the learning set. Our proposed KNN model requires several optimal parameters selected with GridSearchCV stated as follow:

- Algorithm: auto
- n_neighbors: 2
- weights: uniform
- Metric= minkowski
- n_jobs=1

C. Multi-Classes Techniques

Normally, when we apply a classification problem while having more than two classes of instances on the trained data, it might face a complex phase during the data analysis, model training and accurate result prediction. To overcome this approach, the scientists use the multi-classes techniques which is a classification technique that classifies the testing data into several labels of classes presented in the trained instances for forecasting purpose. Thus, we will discuss the idea of applying the two kinds techniques; the One vs. All (one-vs-rest) and One vs. One.

- **One Vs One Classifier**

One-Vs-One is heuristic classification technique which employs a classification algorithm fundamental for multi-class classification problem. The one-vs-one approach plays an important role by splitting the input

data into one reliable data such for every class against each other class. In this model, the Classifier has been trained with an estimator that were previously learnt using the random forest model.

- **One Vs Rest Classifier**

One-vs-rest is also considered as heuristic method which uses the classification algorithm for multi class classification problems. It includes the splitting of instances related to multi classes dataset into various binary classification approaches. This method in demand when the classifier trained every classification outcome. Thus, the prediction will be involved based on a predefined powerful model. In this model, the Classifier has been trained with an estimator that were previously learnt using the random forest model.

ii. Ensemble Techniques

In this part, we proposed the ensemble methods (ET) to be applied in our system. This technique joins some weak base models in order to generate a powerful and effective model. In this experiment six ET models were deployed which are the extreme gradient boosting (XGBoost), random forest, adaboost, bagging, voting and stacking.

A. XGBoost

XGBoost is an ensemble method based on the decision tree algorithm that improves the gradient boosting and used for both classification and regression problems. Gradient boosting sequentially adds the predictors in order to correct the former weak estimators to minimize the detection loss. This model was performed using several optimal parameters selected with GridSearchCV; the maximum depth of the individual regression estimators was selected as max_depth= 23 and the number of estimators were fixed at n_estimators=500 in order to select the number of boosting stages to perform it later.

B. Random Forest

Random Forests classifier includes several decision trees used to classify an input feature vector where its vector is the input of every tree. Every tree produces a selected prediction based on the majority of the vote result. In this model our selected optimal parameters for the GridSearchCV were stated as follow; bootstrap=False, class_weight= none, criterion= gini, n_estimators= 500, max_depth= 23 and random_state= 42.

C. Adaboost

The AdaBoost classifier is considered as a meta-estimator that fits the model on the input data. The main goal is to adapt further copies of the classifier where the weights of wrongly classified instances are adjusted like when the dependent classifiers are examined on intricate learning cases. This model was

performed with a base estimator which is the random forest classifier in which n_estimators=10 and the best selected learning_rate is equal to 0.1.

iii. Evolving the Ensemble Techniques

A. Bagging

Bagging is an ensemble method that uses the variations of samples to decrease the variance of the detected accuracy by tuning the expect result for the outcomes. It generates supplementary data for training phase using the combinations with recurrence to output multi instances based on the same cardinality of the original data. In this model, we suggest the training phase using the random forest model as a base estimator.

B. Voting

The Voting classifier trains several models to predict the outcomes based on the top achieved probability of desired class to be selected as output. It aggregates the results of every predefined models that should be passed into voting technique to forecast the outcome based on the highest majority result of the voting. Thus, it evolves the ML model performance. In this model, we used the soft voting method by suggesting further learning models trained according to the selection of the top achieved performance resulted from the three applied ensemble techniques with their selected parameters which are XGBoost, Random Forest and Adaboost.

C. Stacking

Stacking model combines various classification or regression models via meta classifier or regressor. The base level of this model involves various algorithms and it is often applicable and heterogeneous. Moreover, the essential characteristic of this model is to train the base level of the model based on the entire training set and subsequently trains the meta-model with the resulted base level model like the feature. In this model, we deployed the stacking model to be trained based on the selection of the top achieved performance resulted from the three applied ensemble techniques with their selected parameters which are XGBoost, Random Forest and Adaboost.

iv. Unsupervised Learning

In this part, we selected our model that has been examined in the SIS-ID system during the validation stage of our proposed hardware discussed in the next part. In fact, the proposal of an intelligent model which is deployed on a real time stage should take several concerns. Therefore, we suggested the local outlier factor algorithm that was implemented as follow:

A. LocalOutlierFactor (LOF):

The Local Outlier Factor (LOF) is considered as one of the most used model in the anomaly detection method, due to the importance of overcoming problem of unknown the coming traffics to the network. Therefore, our interest was to propose the SIS-ID to be well configured in order to avoid any attack that can face the server. Hence, we extracted from the DB-DDOS 60000 records related to benign instances to train our model of normal traffics. Thus, our system learned using the benign traffics (inlier) and can be tested in a real time stage against the coming anomalous actions (outliers). On the other hand, we extracted 40000 records from several DDOS instances to prove its effectiveness in the testing phase for detecting any advent attack. Since the LOF is an unsupervised learning technique, the local density deviation should be measured according to a data point computed with its neighbors. Consequently, the sample with ultimately lower intensity than its neighbors will be considered as outlier (attack) where the LOF is the score of each anomaly sample. The model selected the locality for a data point by the k-nearest neighbors' method in order to calculate the distance and to evaluate the local intensity.

Eventually, our model has been trained with the selected optimal parameters stated as follow:

- n_neighbors=49 for kneighbors queries
- novelty=True for novelty detection (unseen attack)
- p=1 to use manhattan_distance.
- algorithm="auto" for the selection of the most proper algorithm that fits the method

8.4.2 Learning Implementation

In our proposed SIS-ID system, we applied several machine learning classification models during the learning phase as stated in the above part. To overcome the long time during the validation of each chunk, we applied our learning methodology according to the suggested pseudocode presented in figure [58]. Thus, we proposed in the learning implementation; the training and evaluation process for each model using the cross-validation technique. It consists of two loops for the training and the evaluation phases [6]. At the beginning of our suggested learning system, both of DB- MALCURL and DB-DDOS were divided into k folds (k = 5 in our study) which should approximately be in similar sizes. Then in the outer loop, for each iteration we took 20% as one data fold that must be reserved for testing stage. Thus, the remaining (k-1) folds get across to the inner loop in order to execute the hyper parameter tuning as an automated model and should be systematically chosen separately from the evaluation instances. In addition, for each model, the inner loop includes a grid search over the parameters and their values must be evaluated while every parameter is evaluated with a (k-1) fold cross validation step. Furthermore, the hyper parameters that achieved the superior average cross-validation were picked. Consequently, the SIS-ID will be trained on the selected data within the (k-1) folds based on the best achieved parameters to be evaluated according to its detection performance for the withstand fold into the outer loop. Eventually, this procedure should be

repeated with k times (k=5) so that each data fold of the outer loop is used once which leads to k evaluation of our system performance.

1. Define set of hyper-parameter combinations, C, for current model. If model has no hyper-parameters, C is the empty set.
2. Divide data into K folds with approximately equal distribution
3. (outer loop) For fold ki, in the K folds:
    1. Set fold ki as the test set
    2. For parameter combination c in C:
        1. (inner loop) For fold ki in the remaining K-1 folds:
            1. Set fold ki as the validation set
            2. Train model on remaining K-2 folds
            3. Evaluate model performance on fold ki,
        2. Calculate average performance over K-2 folds for parameter combination c
        3. Train model on K-1 folds using hyper-parameter combination that yielded best average performance over all steps of the inner loop
    4. Evaluate model performance on fold ki.
4. Calculate average performance over K folds

Figure 58: The proposed pseudocode that was applied in the SIS-ID learning implementation

8.4.3 Learning Optimization Method

As our proposed security intelligent system based on the machine learning technique, we faced a rigorous challenge in developing the SIS-ID system which is the selection of the model optimization technique. Our priority was to find the best theory for the modeling optimization technique to be dedicated during the learning stage. Thus, the hyperparameter optimization technique was intended in our learning stage which is a meta-optimization task, each trial of a particular hyperparameter setting will involve the model training and it can be an inner optimization process. The outcome of this technique achieved the best parameters combination set that affects our chosen algorithms to fulfill the best performance in the validation set stated as follow:

Hyperparameter has an immediate effect on the learning phase of our proposed system. Therefore, we used the GridSearchCV that loops through a predefined dictionary of hyperparameters to fit the model for attaining the powerful affected parameters and specifying the number of times for the cross-validation parameter per each set of hyperparameters. The selected Parameters for the GridSearchCV Object are listed as seen below:

- Estimator: The selected model.
- Params_grid: The parameters of the dictionary model that holds the hyperparameter.
- Scoring: Evaluation metric.
- N_jobs: Number of processes to be executed in parallel.
- cv: Number of the cross-validation technique which is 5 in our system.
- Verbose: This is fixed to 1 to get the detailed output while we are fitting the data to the GridSearchCV object.

Chapter 9: Results: SIS-ID Performance Evaluation

9.1 General Overview

It is an undeniable verity that sensitive information has now an important presence for all companies, organizations and institutions. Therfore, enhancing the cyber-security by developing an intrusion detection system (IDS) based on machine learning techniques is a crucial fact. It can be considered as one of the most important cyber-defense tools over the networks. Nowadays, scientists are interested in machine learning which is a subfield of artificial intelligence. It gives the computers the ability to learn using intelligent algorithms to learn and make prediction based on given data.

In this chapter, we will explain the obtained results of our suggested SIS-ID based on the machine learning techniques. Moreover, we applied several techniques based on the ensemble models that achieved an evolving performance of our SIS-ID. Thus, our suggested system was implemented to detect the latest malicious and DDOS attacks after applying several optimization methods during the learning phase. Untimely, our proposed SIS-ID has been validated based on intelligent hardware with the effectiveness to avoid the coming cyber-attacks on real time stage. Hence, for further results' validation, we calculated the recall value (sensitivity) and the precision value (specificity). These two performance measurements were employed to provide a harmonic value in order to calculate the F1 score (F-measure) as well as obtaining the accuracy rate.

9.2 Experimental Results and Discussion

9.2.1 Applying the SIS-ID System on DB-MALCURL

In this part, we will state the obtained results of our SIS-ID system applied on the DB-MALCURL using the supervised learning and the ensemble techniques.

9.2.1.1 Supervised Learning

In this part, our proposed models were examined with several supervised learning algorithms which are; the decision Tree, k-nearest neighbors (KNN) and the multi-classes techniques via the OneVsRest and OneVsOne models.

As shown in the table [25] which represents a summary of the performance measurements extracted from each classification report from the applied models. We found that the multi classes techniques achieved the top performance, the OneVSRest attained a precision of 98.28%, recall (98.21%), F1-score (98.24%) and an accuracy of 98.20 %. The OneVsOne model achieved a precision (98.13%), recall (98.05%), F1-score (98.09%) and an accuracy of 98.05 %. Furthermore, we noticed that the KNN model achieved a precision

of 96.42%, recall (96.55%), F1-score (96.45%) and an accuracy of 96.40%. However, the decision tree model recorded a precision (95.58%), recall (95.68%), F1-score (95.62%) and an accuracy of 95.55 %.

Model	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Accuracy ⇩
OneVsRest	0.982861	0.982111	0.982456	0.982016
OneVsOne	0.981371	0.980555	0.980926	0.980518
KNN	0.964221	0.965574	0.964551	0.964033
Decision Tree	0.955805	0.956841	0.956249	0.955586

Table 25: Results of the applied supervised learning techniques tested via the DB-MALCURL

In addition, the table [26] displays the detection per class as well as the measurements coefficient using the supervised learning models as follows:

Model	Nb. Instance	Coefficient	One Vs Rest		One Vs One		KNN		Decision Tree	
			Rate	Detect	Rate	Detect	Rate	Detect	Rate	Detect
Defacement	1547	Precision	98.96%	**1527**	98.96%	1526	95.50%	1527	95.50%	1502
		Recall	98.71%		98.64%		98.71%		98.71%	
		F1-score	98.84%		98.80%		97.08%		97.08%	
Benign	1506	Precision	97.83%	1485	97.76%	**1487**	96.96%	1469	96.96%	1454
		Recall	98.61%		98.74%		97.54%		97.54%	
		F1-score	98.21%		98.25%		97.25%		97.25%	
Malware	1334	Precision	99.39%	1302	99.23%	1297	96.13%	**1315**	96.13%	1292
		Recall	97.60%		97.23%		98.58%		98.58%	
		F1-score	98.49%		98.22%		97.34%		97.34%	
Phishing	1589	Precision	95.84%	**1544**	95.53%	1540	96.27%	1421	96.27%	1431
		Recall	97.17%		96.92%		89.43%		89.43%	
		F1-score	96.50%		96.22%		92.72%		92.72%	
Spam	1364	Precision	99.41%	**1350**	99.19%	1347	97.25%	1344	97.25%	1335
		Recall	98.97%		98.75%		98.53%		98.53%	
		F1-score	99.19%		98.97%		97.89%		97.89%	

Table 26: A summary about the results for detecting each class with the measurements coefficient using the supervised learning models that tested via the DB-MALCURL

After analyzing the table [26], we observed that the one vs rest attained the top accurate model as well as the best classifier for detecting the defacement, phishing and spam classes. The results were obtained from the confusion matrix shown in the figure [59]. Hence, among 1547 records classified as defacement, we detected 1527 defacement, 1 benign, 0 malware, 17 phishing and 2 spam with a precision rate of 98.96%, recall (98.71%) and f1-score (98.84%). Moreover, among the 1506 records classified as benign, we obtained 2 defacement, 1485 benign, 4 malware, 15 phishing and 0 spam with a precision rate of 97.83%, recall (98.61%) and f1-score (98.21%). Among 1334 records classified as malware, we identified 0 defacement, 6 benign, 1302 malware, 25 phishing and 1 spam with a precision rate of (99.39%), recall (97.60%) and f1-score (98.49%) while among 1589 records classified as phishing, we obtained 10 defacement, 26 benign, 4 malware, 1544 phishing and 5 spam with a precision rate 95.84%, recall

(97.17%) and f1-score (96.50%). Finally, among 1364 related to the spam attack, we attained 4 defacement, 0 benign, 0 malware, 10 phishing and 1350 spam with the high coefficient measurements; precision rate by 99.41%, recall (98.97%) and f1-score (99.19%).

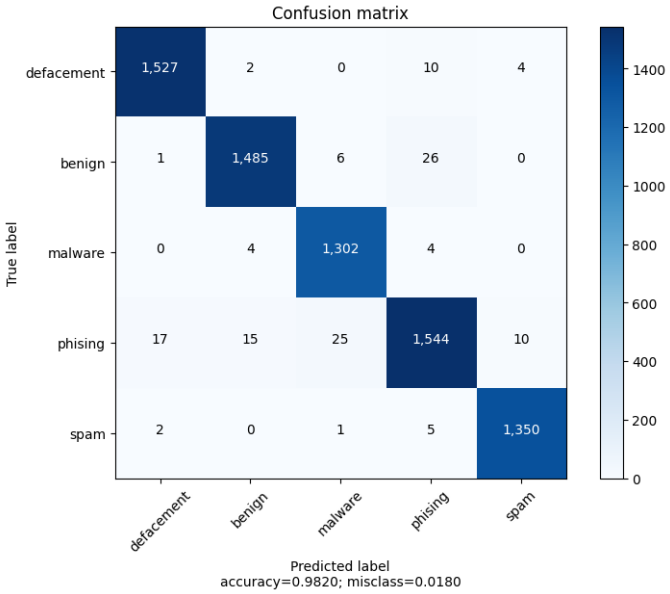


Figure 59: Confusion matrix for OVR model on the DB- MALCURL

Moreover, the one vs one was classified as the best classifier in detecting the benign class. The obtained results were extracted from the confusion matrix shown in figure [60]. Thus, among the 1506 records which classified as benign, we detected 1 defacement, 1487 benign, 4 malware, 13 phishing and 1 spam with a precision of 97.76%, recall (98.74%) and f1-score (98.25%). Furthermore, among 1547 records classified as defacement, we obtained 1526 defacement, 1 benign, 0 malware, 18 phishing and 2 spam. Besides, among 1334 records classified as malware, we identified 0 defacement, 8 benign, 1297 malware, 28 phishing and 1 spam. However among 1589 records classified as phishing, we obtained 11 defacement, 25 benign, 6 malware, 1540 phishing and 7 spam. Finally, among 1364 related to the spam attack, the system recorded 4 defacement, 0 benign, 0 malware, 13 phishing and 1347spam.

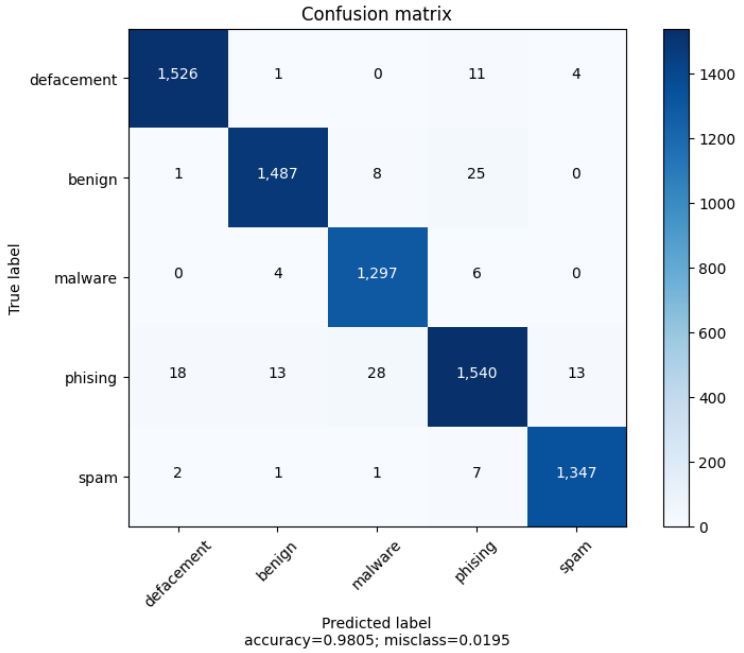


Figure 60: Confusion matrix for OVO model on the DB- MALCURL

Furthermore, the KNN was classified as the best classifier in detecting the malware class. The obtained results were extracted from the confusion matrix shown in figure [61]. Among 1334 records classified as malware, we detected 0 defacement, 5 benign, 1315 malware, 13 phishing and 1 spam with a precision rate of 96.13%, recall (98.58%) and f1-score (97.34%). Moreover, among 1547 records classified as defacement, we obtained 1527 defacement, 1 benign, 2 malware, 12 phishing and 5 spam. Among the 1506 records that classified as benign, we attained 8 defacement, 1469 benign, 11 malware, 17 phishing and 1 spam while among 1589 records classified as phishing, we reached 58 defacement, 40 benign, 39 malware, 1421 phishing and 31 spam. Finally, among 1364 related to the spam attack we got 6 defacement, 0 benign, 1 malware, 13 phishing and 1344 spam.

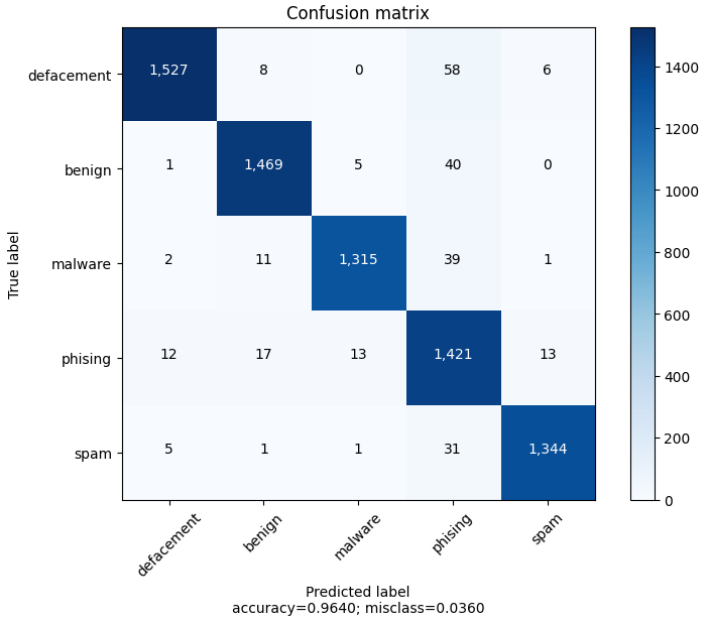


Figure 61: Confusion matrix for KNN model on the DB-MALCURL

Ultimately, the results of the decision tree classifier showed the lowest performance in detecting all the classes. As shown in the confusion matrix in figure [62], among 1547 records classified as defacement, we obtained 1502 defacement, 3 benign, 7 malware, 28 phishing and 7 spam. Further, among 1506 records classified as benign, we attained 9 defacement, 1454 benign, 10 malware, 32 phishing and 1 spam. Moreover, among 1334 records classified as malware, we achieved 1 defacement, 4 benign, 1292 malware, 35 phishing and 2 spam while among 1589 records classified as phishing, we obtained 44 defacement, 36 benign, 47 malware, 1431 phishing and 31 spam. Finally, among 1364 for related to the spam attack we attained 7 defacement, 1 benign, 6 malware, 15 phishing and 1335 spam.

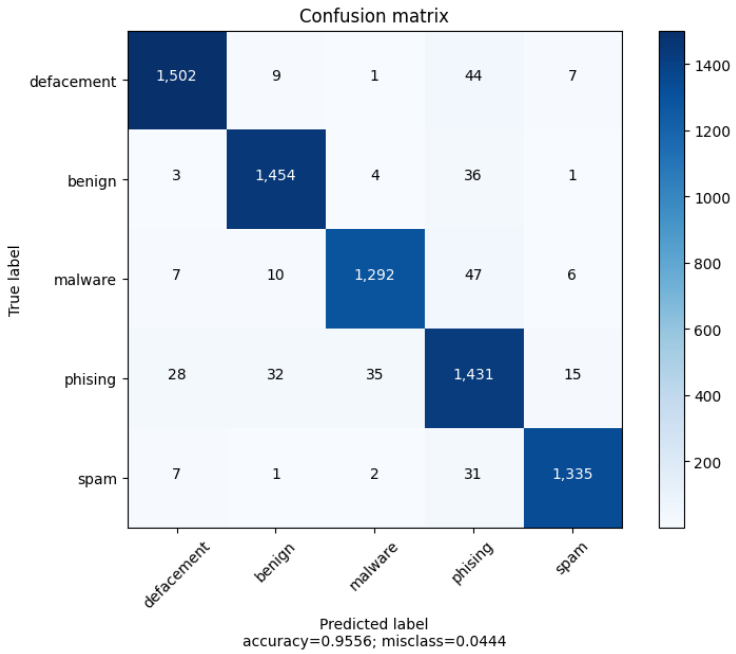


Figure 62: Confusion matrix for Decision Tree model on the DB-MALCURL

9.2.1.2 Ensemble Techniques

In this part, our proposed models were examined with three ensemble techniques. We will discuss the performance measurements outputted from the extreme gradient boosting (XGBoost), random forest and Adaboost models. As shown in table [27], we found that the ensemble techniques achieved high performance than the supervised learning. Further, the XGBoost attained the best model with a precision of 98.46%, recall (98.46%), F1-score (98.46%) and an accuracy of 98.43 %. The Random Forest model recorded a precision of 98.26%, recall (98.18%), F1-score (98.21%) and an accuracy of 98.17% while the Adaboost model obtained a precision (98.18%), recall (98.10%), F1-score (98.13%) and an accuracy of 98.09 %.

Model	Precision Weighted Average	Recall Weighted Average	F1-Score Weighted Average	Accuracy ⇩
XGBoost	0.98465	0.984643	0.984629	0.984332
Random Forest	0.982644	0.981804	0.982191	0.981744
Adaboost	0.981808	0.981014	0.981383	0.980926

Table 27: Results of the applied ensemble techniques tested via the DB-MALCURL

In addition, the table [28] displays the detection for each class as well as the measurements coefficient using the ensemble techniques as follows:

Attack	**Nb. Instance**	**Coefficient**	**XG-boost**		**Random Forest**		**Adaboost**	
			Rate	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**
Defacement	1547	Precision	98.53%	**1538**	98.77%	1527	98.58%	1526
		Recall	99.42%		98.71%		98.64%	
		F1-score	98.97%		98.74%		98.61%	
Benign	1506	Precision	97.96%	1489	97.83%	**1487**	97.89%	1485
		Recall	98.87%		98.74%		98.61%	
		F1-score	98.41%		98.28%		98.25%	
Malware	1334	Precision	98.72%	1313	99.31%	1300	99.24%	**1301**
		Recall	98.43%		97.45%		97.53%	
		F1-score	98.57%		98.37%		98.37%	
Phishing	1589	Precision	97.70%	**1532**	95.78%	1543	95.71%	1541
		Recall	96.41%		97.11%		96.98%	
		F1-score	97.05%		96.44%		96.34%	
Spam	1364	Precision	99.41%	**1353**	99.63%	1349	99.48%	1347
		Recall	99.19%		98.90%		98.75%	
		F1-score	99.30%		99.26%		99.12%	

Table 28: A summary about the results for detecting each class with the measurements coefficient using the ensemble techniques that tested via the DB-MALCURL

After analyzing the table [28], we deduced that the XGBoost is the best classifier for detecting the defacement, phishing and spam classes. The results were obtained from the confusion matrix shown in figure [63]. Thus, among 1547 records classified as defacement, we detected 1538 defacement, 1 benign, 0 malware, 7 phishing and 1 spam with a precision rate of (98.53%), best recall (99.42%) and f1-score

(98.97%). Among the 1506 records classified as benign, we obtained 0 defacement, 1489 benign, 10 malwares, 6 phishing and 1 spam with a precision rate of 97.96%, recall (98.87%) and f1-score (98.41%). Moreover, among 1334 records classified as malware, we identified 0 defacement, 3 benign, 1313 malware, 18 phishing and 0 spam with a precision rate by 98.72%, recall (98.43%) and f1-score (98.57%). However, among 1589 records classified as phishing, we obtained 17 defacement, 27 benign, 7 malware, 1532 phishing and 6 spam with a precision rate of 97.70%, recall (96.41%) and f1-score (97.05%). Finally, the system recorded among 1364 related to the spam attack 6 defacement, 0 benign, 0 malware, 5 phishing and 1353 spam with the best precision rate of 99.41%, recall (99.19%) and the top f1-score rate (99.30%).

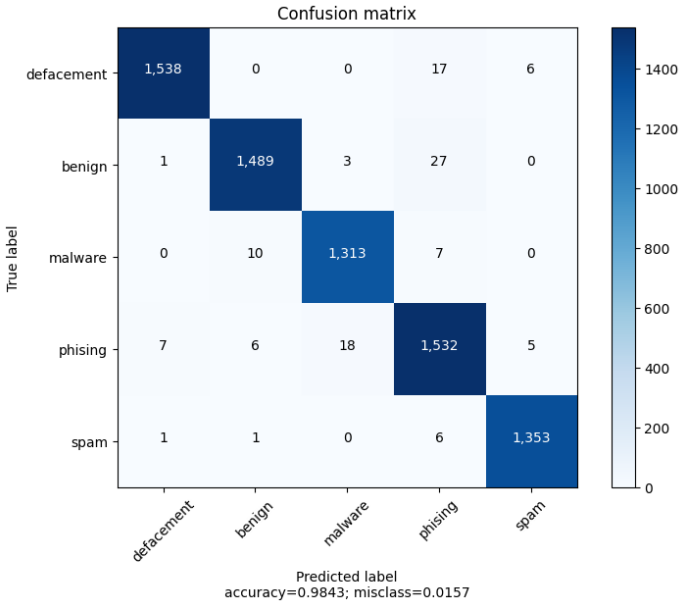


Figure 63: Confusion matrix for XGBoost model on the DB-MALCURL

In addition, the random forest model was classified as a best classifier in detecting the benign class. The obtained results were extracted from the confusion matrix shown in figure [64]. Thus, among the 1506 records classified as benign, we detected 1 defacement, 1487 benign, 4 malware, 14 phishing and 0 spam with a precision rate of 97.83%, best recall (98.74%) and f1-score(98.28%). Moreover, among 1547 records classified as defacement, we obtained 1527 defacement, 1 benign, 1 malware, 17 phishing and 1 spam. Further, among 1334 records classified as malware, we identified 0 defacement, 7 benign, 1300 malware, 26 phishing and 1 spam while among 1589 records classified as phishing, we obtained 14 defacement, 25 benign, 4 malware, 1543 phishing and 3 spam. Finally, the system recorded among 1364 related to the spam attack 4 defacement, 0 benign, 0 malware, 11 phishing and 1349.

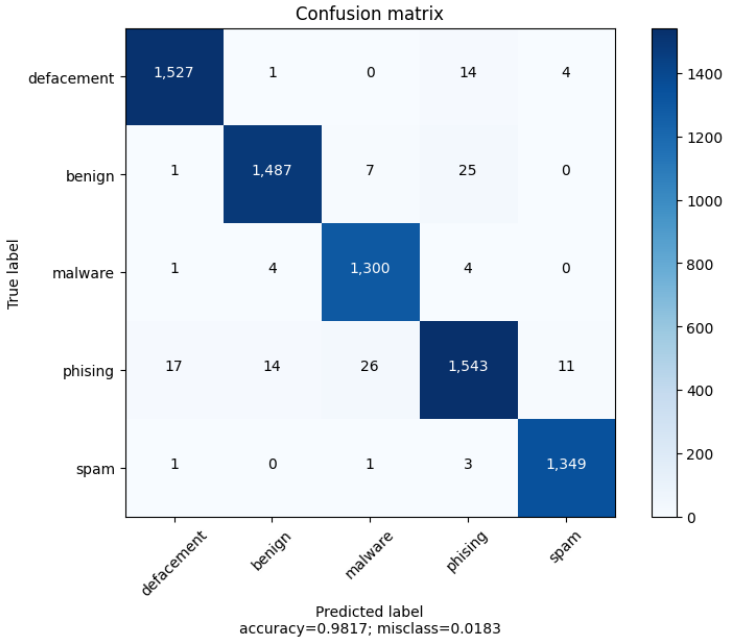


Figure 64: Confusion matrix for Random Forest model on the DB- MALCURL

Furthermore, by employing the Adaboost model, we concluded that it was selected as the best classifier in detecting the malware class. The reached results were mined from the confusion matrix as seen in figure [65]. Hence, among 1334 records classified as malware, we detected 0 defacement, 8 benign, 1301 malware, 23 phishing and 2 spam with a precision of 99.24%, recall (97.53%) and f1-score (98.37%). Moreover, among 1547 records classified as defacement, we attained 1526 defacement, 1 benign, 1 malware, 18 phishing and 1 spam. Among the 1506 records classified as benign, we obtained 1 defacement, 1485 benign, 4 malware, 16 phishing and 0 spam while among 1589 records classified as phishing, we achieved 16 defacement, 23 benign, 5 malware, 1541 phishing and 4 spam. Finally, among 1364 related to the spam attack we attained 5 defacement, 0 benign, 0 malware, 12 phishing and 1347 spam.

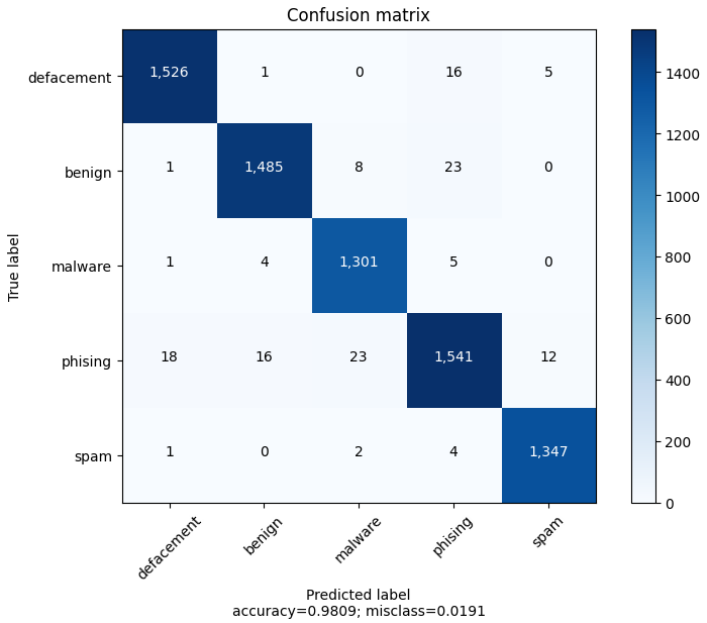


Figure 65: Confusion matrix for Adaboost model on the DB- MALCURL

9.2.1.3 Evolving the Ensemble Techniques

In this part, we will show the enhancement results of the ensemble techniques (ET) by suggesting further ensemble models (EM). These models were trained according to the ET estimators that achieved the top performance shown in the part above. Therefore, our expectation is to produce powerful and accurate results than the previous models. As presented in table [29], we found that the voting and the stacking models enhanced the results. These models outputted the best performance measurement during the detection phase. Further, the voting model achieved a high performance with a precision of 98.57%, recall (98.55%), F1-score (98.56%) and an accuracy of 98.52 %. The stacking recorded a precision of 98.48%, recall (98.47%), F1-score (98.48%) and an accuracy of 98.44 % while the bagging classifier resulted the low performance measurements among the proposed ensemble models with a precision of 97.99%, recall (97.90%), F1-score (97.94%) and an accuracy of 97.90 %.

Model	Estimator	Precision Weighted Average	Recall Weighted Average	F1-Score Weighted Average	Accuracy
Voting	- XGBoost - Random Forest - Adaboost	0.98571	0.985518	0.985602	0.985286
Stacking	- XGBoost - Random Forest - Adaboost	0.984885	0.984728	0.984804	0.984469
Bagging	- RandomForest	0.979931	0.979032	0.979441	0.979019

Table 29: Results of the proposed ensemble models tested via the DB-MALCURL

In addition, the table [30] displays the detection of each class as well as the coefficient measurements using our proposed ensemble models as stated below.

Model	Nb. Instance	Coefficient	Voting		Stacking		Bagging	
			Rate	Detect	Rate	Detect	Rate	Detect

Defacement	1547	Precision	98.84%	1535	99.03%	**1536**	98.77%	1528
		Recall	99.22%		99.29%		98.77%	
		F1-score	99.03%		99.16%		98.77%	
Benign	1506	Precision	98.03%	**1493**	98.54%	**1482**	97.37%	1483
		Recall	99.14%		98.41%		98.47%	
		F1-score	98.58%		98.47%		97.92%	
Malware	1334	Precision	99.09%	**1312**	98.87%	1311	99.08%	**1293**
		Recall	98.35%		98.28%		96.93%	
		F1-score	98.72%		98.57%		97.99%	
Phishing	1589	Precision	97.41%	1539	96.74%	**1541**	95.40%	1536
		Recall	96.85%		96.98%		96.66%	
		F1-score	97.13%		96.86%		96.03%	
Spam	1364	Precision	99.49%	1353	99.27%	**1356**	99.34%	1346
		Recall	99.19%		99.41%		98.68%	
		F1-score	99.34%		99.34%		99.01%	

Table 30: A summary about the results for detecting each class with the coefficient measurements using our proposed ensemble models that tested via the DB-MALCURL

Hence, after analyzing the table [30], we concluded that the voting is the best classifier for detecting the benign and malware classes. The results were obtained from the confusion matrix shown in figure [66]. Hence, among 1547 records classified as defacement, we detected 1535 defacement, 1 benign, 0 malware, 10 phishing and 1 spam with a precision rate of 98.84%, the best recall (99.22%) and f1-score (99.03%). Among the 1506 records classified as benign, we obtained 1 defacement, 1493 benign, 6 malware, 6 phishing and 0 spam with a precision rate of 98.03%, recall (99.14%) and f1-score (98.58%). Moreover, among 1334 records classified as malware, we identified 0 defacement, 3 benign, 1312 malware, 18 phishing and 1 spam with a precision rate of 99.09%, recall (98.35%) and f1-score (98.72%) while among 1589 records classified as phishing, we obtained 13 defacement, 26 benign, 6 malware, 1539 phishing and 5 spam with a precision rate of 97.41%, recall (96.85%) and f1-score (97.13%). Finally, the system recorded among 1364 instances of spam; 4 defacement, 0 benign, 0 malware, 7 phishing and 1353 with the best precision rate (99.49%), recall (99.19%) and the top f1-score rate (99.34%).

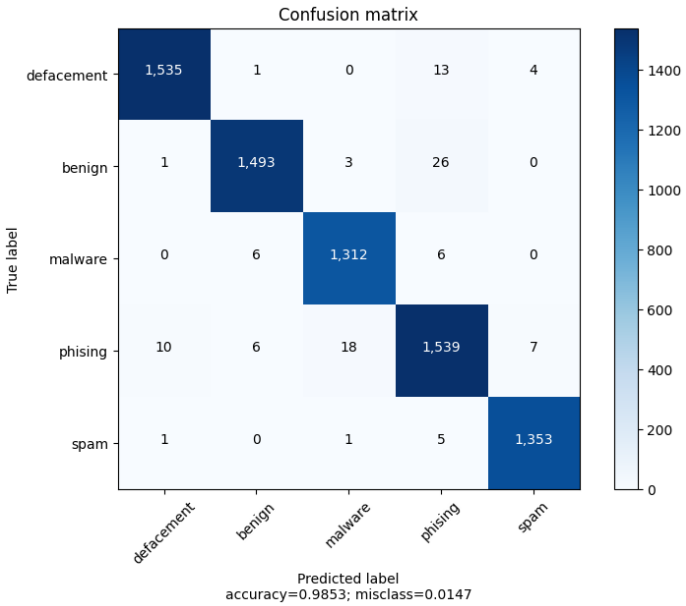


Figure 66: Confusion matrix for Voting model on the DB-MALCURL

In addition, the stacking model was classified as the best classifier in detecting the benign, phishing and spam classes. The obtained results were extracted from the confusion matrix shown in the figure [67]. Thus, among 1547 records classified as defacement, we obtained 1536 defacement, 1 benign, 1 malware, 7 phishing and 2 spam. Among the 1506 records classified as benign, we detected 1 defacement, 1482 benign, 6 malware, 17 phishing and 0 spam with a precision rate of 98.54%, recall (98.41%) and f1-score (98.47%). Moreover, among 1334 records classified as malware, we identified 0 defacement, 2 benign, 1311 malware, 21 phishing and 0 spam while among 1589 records classified as phishing, we obtained 13 defacement, 19 benign, 8 malware, 1541 phishing and 8 spam with a precision rate of 96.74%, recall (96.98%) and f1-score (96.86%). Finally, the system recorded among 1364 instances of spam attack 1 defacement, 0 benign, 0 malware, 7 phishing and 1356 with a precision rate of 99.27%, recall (99.41%) and f1-score (99.34%).

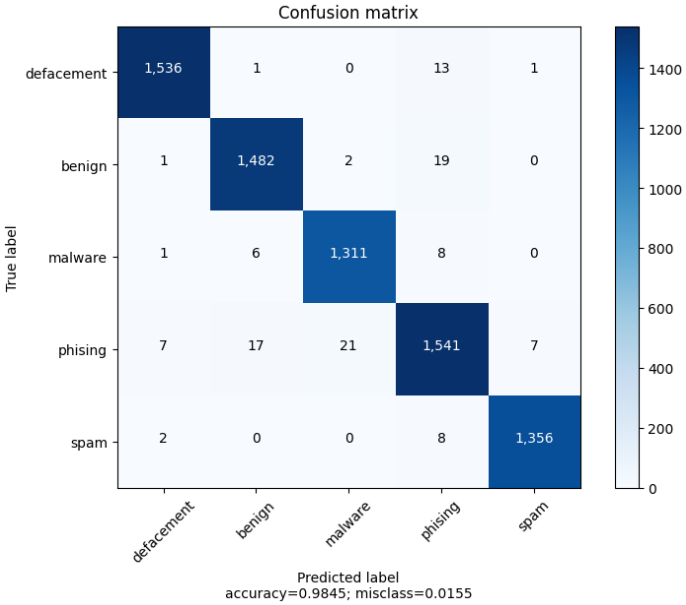


Figure 67: Confusion matrix for Stacking model on the DB-MALCURL

Eventually, the bagging model is classified as the best classifier in detecting the malware class. The obtained results were extracted from the confusion matrix shown in the figure [68]. Thus, among 1547 records classified as defacement we detected 1528 defacement, 1 benign, 1 malware, 16 phishing and 1 spam. Among the 1506 records classified as benign, we obtained 1 defacement, 1483 benign, 6 malware, 16 phishing and 0 spam. Moreover, among 1334 records classified as malware, we identified 0 defacement, 10 benign, 1293 malware, 29 phishing and 2 spam with a precision rate of 99.08%, recall (96.93%) and f1-score (97.99%) while among 1589 records classified as phishing, we got 13 defacement, 29 benign, 5 malware, 1526 phishing and 6 spam. Finally, the system recorded among 1364 of spam attacks 5 defacement, 0 benign, 0 malware, 13 phishing and 1346 spam.

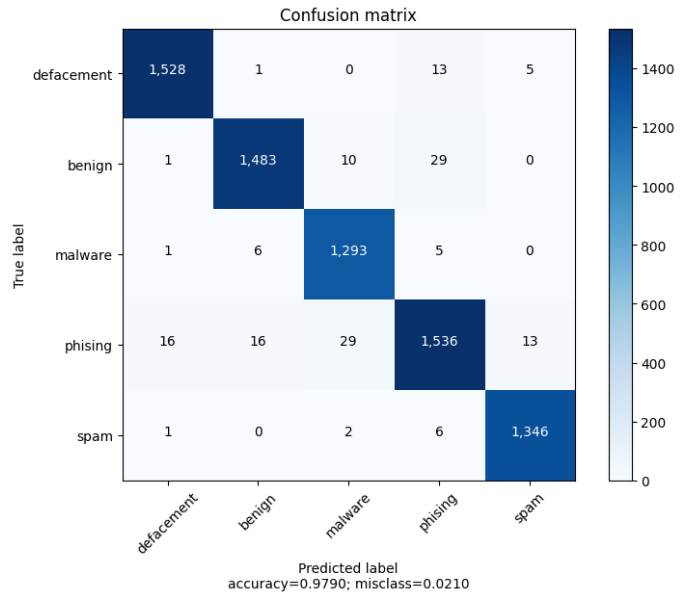


Figure 68: Confusion matrix for Bagging model on the DB-MALCURL

9.2.2 Applying the SIS-ID system on DB-DDOS

In this part we will state the obtained results of our SIS-ID system that was employed on the DB-DDOS using the supervised learning and the ensemble techniques.

9.2.2.1 Supervised Learning

Our models were validated using several supervised learning algorithms which are; the decision Tree, k-nearest neighbors (KNN) and the multi-classes techniques via the OneVsRest and OneVsOne models. As revealed in the table [31] which represents a summary of the performance measurements extracted from each classification report of those models. We found that the multi classes techniques achieved the top performance, the OneVSRest attained a precision of 79.60%, recall (76.85%), F1-score (76.03%) and an accuracy of 76.82% while the OneVsOne model recorded a precision (79.43%), recall (76.81%), F1-score (76.02%) and an accuracy of 76.78 %. Furthermore, we noticed that the Decision Tree model attained a precision (79.29%), recall (76.74%), F1-score (75.97%) and an accuracy of 76.71 % and finally the KNN model achieved a precision (73.51%), recall (72.52%), F1-score (71.98%) and an accuracy of 72.48 %.

Model	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Accuracy ⇓
OneVsRest	0.7961	0.7685	0.7604	0.7682
OneVsOne	0.7944	0.7682	0.7603	0.7679
Decision Tree	0.793	0.7675	0.7597	0.7672
KNN	0.7352	0.7252	0.7199	0.7248

Table 31: Results of the applied supervised learning techniques tested via the DB-DDOS

As we can observe, table [32] displays the detection stage for each class as well as the coefficient measurements using the supervised learning models as follows:

Attack	Nb. Instance	Coefficient	One Vs Rest		One Vs One		Decision Tree		KNN	
			Rate	Detect	Rate	Detect	Rate	Detect	Rate	Detect
BENIGN	11133	Precision	99.54%	**10998**	99.56%	10995	99.49%	10995	93.65%	10337
		Recall	98.79%		98.76%		98.67%		92.85%	
		F1-score	99.16%		99.16%		99.08%		93.25%	
DrDoS_ DNS	11195	Precision	81.99%	5493	81.85%	5500	79.97%	5479	47.37%	**8214**
		Recall	49.07%		49.13%		48.94%		73.37%	
		F1-score	61.39%		61.40%		60.72%		57.57%	
DrDoS_ LDAP	11175	Precision	52.02%	8355	52.02%	**8362**	51.94%	8285	50.86%	4763
		Recall	74.77%		74.83%		74.14%		42.62%	
		F1-score	61.35%		61.37%		61.09%		46.38%	
DrDoS_ MSSQL	11210	Precision	71.39%	4905	71.58%	4923	71.90%	4928	58.26%	**5011**
		Recall	43.76%		43.92%		43.96%		44.70%	
		F1-score	54.26%		54.43%		54.56%		50.59%	
DrDoS_ NetBIOS	11127	Precision	97.52%	10550	97.57%	10551	97.28%	**10549**	95.57%	10185
		Recall	94.81%		94.82%		94.81%		91.53%	
		F1-score	96.15%		96.18%		96.03%		93.51%	

DrDoS_ NTP	11265	Precision	79.04%	8076	78.86%	8071	79.15%	**8091**	76.50%	7745
		Recall	71.69%		71.65%		71.82%		68.75%	
		F1-score	75.19%		75.08%		75.31%		72.42%	
DrDoS_ SNMP	11262	Precision	89.72%	**11156**	89.73%	11155	89.68%	11151	89.67%	11114
		Recall	99.06%		99.05%		99.01%		98.69%	
		F1-score	94.16%		94.16%		94.12%		93.96%	
DrDoS_ SSDP	11325	Precision	61.65%	**9072**	62.55%	8759	61.76%	8828	68.44%	6435
		Recall	80.11%		77.34%		77.95%		56.80%	
		F1-score	69.68%		69.16%		68.92%		62.08%	
DrDoS_ UDP	11186	Precision	70.04%	10085	69.92%	10078	70.52%	**11186**	68.84%	9209
		Recall	90.16%		90.09%		89.04%		82.33%	
		F1-score	78.84%		78.73%		78.71%		74.98%	
Syn	11182	Precision	96.05%	4714	95.83%	4709	95.52%	4729	79.44%	**5158**
		Recall	42.16%		42.11%		42.29%		46.13%	
		F1-score	58.60%		58.51%		58.63%		58.36%	
TFTP	11178	Precision	61.98%	**10953**	61.95%	10942	61.99%	10928	61.90%	10076
		Recall	97.99%		97.89%		97.76%		90.14%	
		F1-score	75.93%		75.88%		75.87%		73.40%	
UDP-lag	11244	Precision	77.13%	6400	74.31%	6657	74.63%	**6702**	68.96%	6255
		Recall	56.92%		59.20%		59.61%		55.63%	
		F1-score	65.50%		65.90%		66.28%		61.58%	
WebDDoS	11118	Precision	96.85%	11099	96.95%	**11101**	97.04%	11088	96.28%	11036
		Recall	99.83%		99.85%		99.73%		99.26%	
		F1-score	98.32%		98.38%		98.37%		97.75%	

Table 32: A summary about the results for detecting each class with the coefficient measurements using the supervised learning models that tested via the DB-DDOS

After analyzing the table [32], we proved that the one vs rest is the best classifier in detecting the BENIGN, DrDoS_SNMP, DrDoS_SSDP and TFTP classes. The results were obtained from the confusion matrix shown in the figure [69]. Hence, among 11133 records classified as benign, we discovered 10998 benign and 135 instances for the remaining attack classes as false positive with a precision rate of 99.54%, recall (98.79%) and f1-score (99.16%). Further, among the 11262 records classified as DrDoS_SNMP, we detected 11156 DrDoS_SNMP and 106 instances for the remaining classes as false positive with a precision rate of 89.72%, recall (99.06%) and f1-score (94.16%). Moreover, among the 11325 records classified as DrDoS_SSDP, we noticed 9072 of DrDoS_SSDP and 2253 instances for the remaining classes as false positive with a precision rate of 61.65%, recall (80.11%) and f1-score (69.68%). Finally, among the 11178 records classified as TFTP, we found 10953 of TFTP and 225 instances for the remaining classes as false positive with a precision rate of 61.98%, recall (97.99%) and f1-score (75.93%).

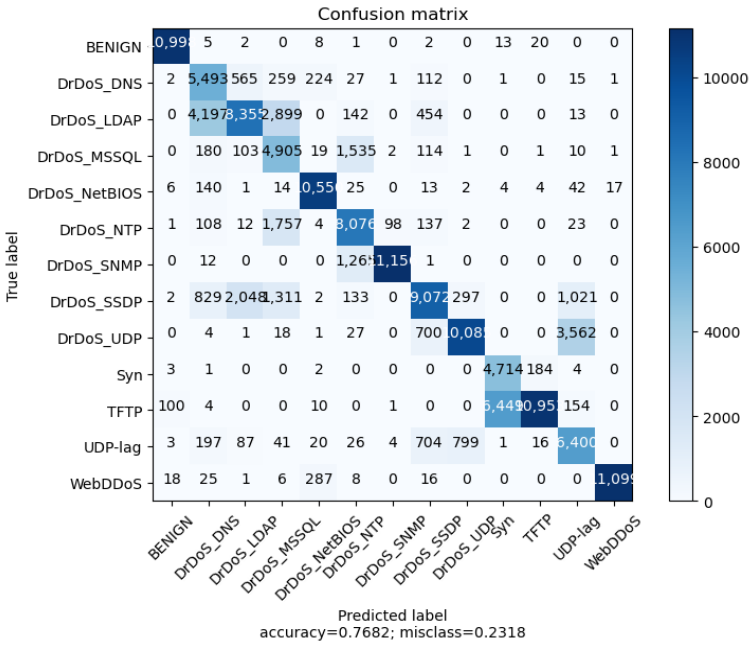


Figure 69: Confusion matrix for OVR model on the DB-DDOS

In addition, the one vs one presents as the best classifier in detecting the DrDoS_LDAP and WebDDoS classes. The results were obtained from the confusion matrix shown in figure [70]. Thus, among the 11175 records classified as DrDoS_LDAP, we detected 8362 DrDoS_LDAP and 2813 instances for the remaining classes as false positive with a precision rate of 52.02%, recall (74.83%) and f1-score (61.37%). Finally, among the 11118 records classified as WebDDoS, we detected 11101 WebDDoS and 17 instances for the remaining classes as false positive with a precision rate of 96.95%, recall (99.85%) and f1-score (98.38%).

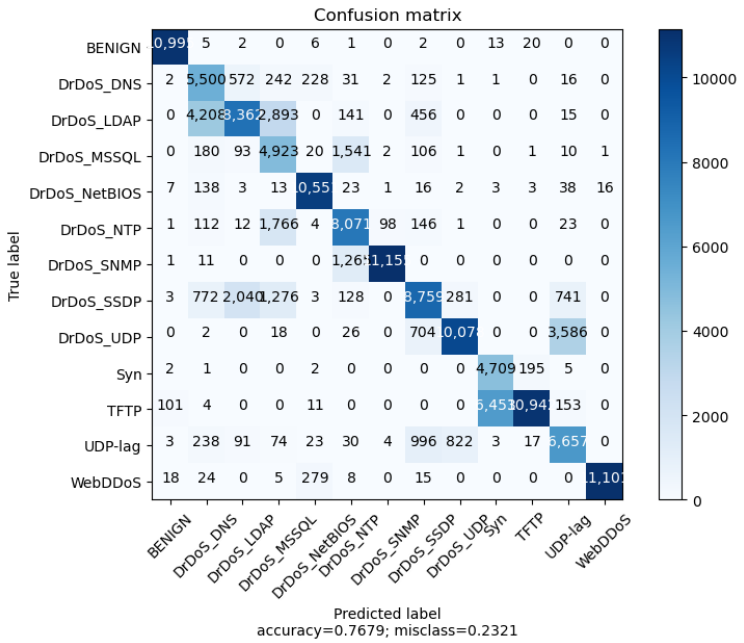


Figure 70: Confusion matrix for OVO model on the DB-DDOS

Furthermore, the decision tree model is the best in detecting the DrDoS_NetBIOS, DrDoS_NTP, DrDoS_UDP and UDP-lag classes. The outputted results were extracted from the confusion matrix seen in the figure [71]. Hence, among the 11127 records classified as DrDoS_NetBIOS, we obtained 10549 DrDoS_NetBIOS and 578 instances for the remaining classes as false positive with a precision rate of 97.28%, recall (94.81%) and f1-score (96.03%). Moreover, among the 11265 records classified as DrDoS_NTP, we got 8091 DrDoS_NTP and 3174 instances for the remaining classes as false positive with a precision rate of 79.15%, recall (71.82%) and f1-score (75.31%). Further, among the 11186 records classified as DrDoS_UDP, we attained 9960 DrDoS_UDP and 1226 instances for the remaining classes as false positive with a precision rate of 70.52%, recall (89.04%) and f1-score (78.71%). Finally, among the 11244 records classified as UDP-lag, we obtained 6702 UDP-lag and 4542 instances for the remaining classes as false positive with a precision rate of 74.63%, recall (59.61%) and f1-score (66.28%).

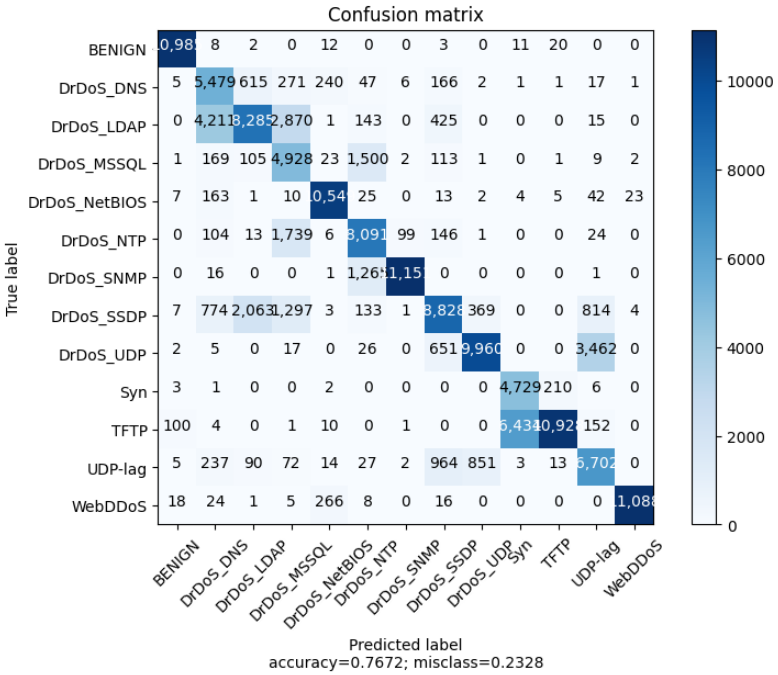


Figure 71: Confusion matrix for Decision Tree model on the DB-DDOS

Ultimately, the KNN model earns to be the best classifier in identifying the DrDoS_DNS, DrDoS_MSSQL and Syn classes. The obtained results were extracted from the confusion matrix shown in figure [72]. Hence, among the 11195 records classified as DrDoS_DNS, we got 8214 DrDoS_DNS and 2981 instances for the remaining classes as false positive with a precision rate of 47.37%, recall (73.37%) and f1-score (57.57%). Further, among the 11210 records classified as DrDoS_MSSQL, we obtained 5011 DrDoS_MSSQL and detected 6199 instances for the remaining classes as false positive with a precision rate (58.26%), recall (44.70%) and f1-score (50.59%). Finally, among the 11182 records classified as Syn, we obtained 5158 Syn and 6024 instances for the remaining classes as false positive with a precision rate (79.44%), recall (46.13%) and f1-score (58.36%).

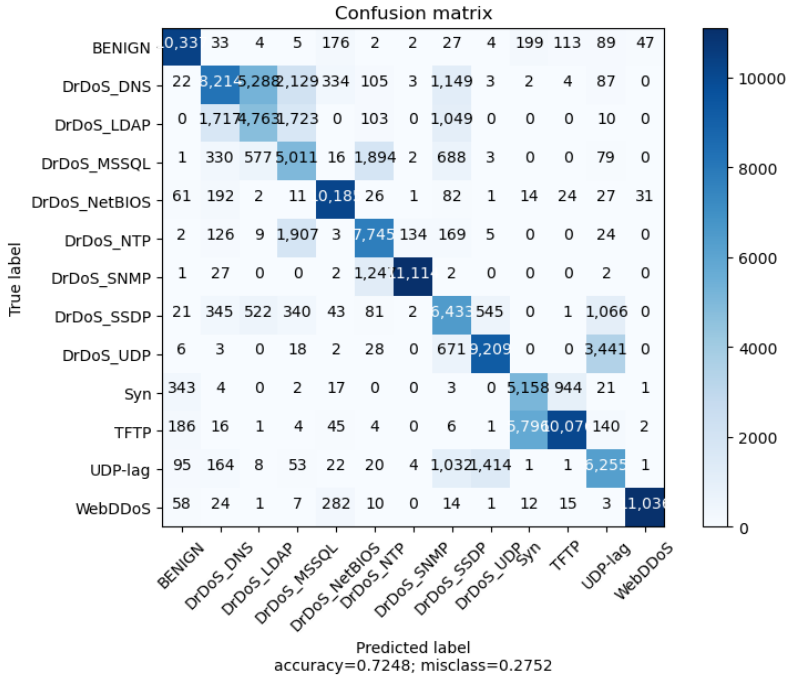


Figure 72 Confusion matrix for KNN model on the DB-DDOS

9.2.2.2 Ensemble Techniques

In this part, our proposed models were examined with three ensemble techniques. Therefore, we will discuss the performance measurements outputted from the extreme gradient boosting (XGBoost), random forest and Adaboost models.

As seen in the table [33] that shows the performance measurements extracted from each applied model, we found that the ensemble techniques achieved a high performance than the supervised learning. Further, the XGBoost attained the best model with a precision of 79.45%, recall (76.89%), F1-score (76.13%) and an accuracy of 76.86%. The Random Forest model recorded a precision (79.64%), recall (76.86%), F1-score (76.05%) and an accuracy of 76.83%. However, the adaboost model obtained a precision (79.34%), recall (76.77%), F1-score (76.02%) and an accuracy of 76.73%.

Model	Precision Weighted Average	Recall Weighted Average	F1-Score Weighted Average	Accuracy ⇩
XGBoost	0.794551	0.768937	0.761347	0.768626
Random Forest	0.796485	0.768653	0.760505	0.768365
Adaboost	0.793434	0.767708	0.760299	0.767397

Table 33: Results of the applied ensemble techniques tested via the DB-DDOS

In addition, the table [34] displays the detection for each class as well as the coefficient measurements using the ensemble techniques as follows:

Attack	Nb. Instance	Coefficient	XG-Boost		Random-Forest		Adaboost	
			Rate	Detect	Rate	Detect	Rate	Detect
BENIGN	11133	Precision	99.53%	**10997**	99.57%	**10997**	99.82%	10967
		Recall	98.78%		98.78%		98.51%	
		F1-score	99.15%		99.17%		99.16%	
DrDoS_DNS	11195	Precision	82.33%	**5545**	81.56%	5477	81.15%	5527
		Recall	49.53%		48.92%		49.37%	
		F1-score	61.85%		61.16%		61.39%	
DrDoS_LDAP	11175	Precision	52.06%	**8357**	51.93%	8340	52.06%	8328
		Recall	74.78%		74.63%		74.52%	
		F1-score	61.38%		61.25%		61.30%	
DrDoS_MSSQL	11210	Precision	71.44%	5015	72.74%	4858	70.23%	**5030**
		Recall	44.74%		43.34%		44.87%	
		F1-score	55.02%		54.31%		54.76%	
DrDoS_NetBIOS	11127	Precision	97.23%	**10584**	97.72%	10545	97.85%	10536
		Recall	95.12%		94.77%		94.69%	
		F1-score	96.16%		96.22%		96.25%	
DrDoS_NTP	11265	Precision	79.79%	8032	78.61%	**8205**	79.56%	7944
		Recall	71.30%		72.84%		70.52%	
		F1-score	75.31%		75.62%		74.77%	
DrDoS_SNMP	11262	Precision	89.76%	**11162**	89.70%	11152	89.69%	11150
		Recall	99.11%		99.02%		99.01%	
		F1-score	94.20%		94.13%		94.12%	
DrDoS_SSDP	11325	Precision	62.87%	8734	61.23%	**9069**	62.21%	8730
		Recall	77.12%		80.08%		77.09%	
		F1-score	69.27%		69.40%		68.85%	
DrDoS_UDP	11186	Precision	70.16%	10003	70.31%	**10061**	70.22%	10005
		Recall	89.42%		89.94%		89.44%	
		F1-score	78.63%		78.92%		78.67%	
Syn	11182	Precision	95.46%	**4729**	95.88%	4726	95.77%	4728
		Recall	42.29%		42.26%		42.28%	
		F1-score	58.61%		58.67%		58.66%	
TFTP	11178	Precision	61.97%	10924	62.00%	10949	61.91%	**10963**
		Recall	97.73%		97.95%		98.08%	
		F1-score	75.85%		75.93%		75.91%	
UDP-lag	11244	Precision	73.33%	**6735**	77.30%	6398	74.01%	6732
		Recall	59.90%		56.90%		59.87%	
		F1-score	65.94%		65.55%		66.19%	
WebDDoS	11118	Precision	97.00%	11095	96.87%	**11097**	96.98%	11093
		Recall	99.79%		99.81%		99.78%	

		F1-score	98.38%		98.32%		98.36%	

Table 34: A summary about the results for detecting each class with the coefficient measurements using ensemble techniques that tested via the DB-DDOS

After analyzing the table [34], we noticed that the XGBoost is best classifier in detecting 7 classes; the BENIGN, DrDoS_DNS, DrDoS_LDAP, DrDoS_NetBIOS, DrDoS_SNMP, Syn and UDP-lag. The results were obtained from the confusion matrix shown in figure [73]. Thus, among 11133 records classified as benign, we resulted 10997 benign and detected 136 instances for the remaining attack classes as false positive with a precision rate of 99.53%, recall (98.78%) and f1-score (99.15%). Further, among the 11195 records classified as DrDoS_DNS, we detected 5545 DrDoS_DNS and 5650 instances for the remaining classes as false positive with a precision rate of 82.33%, recall (49.53%) and f1-score (61.85%). In addition, among the 11175 records classified as DrDoS_LDAP, we obtained 8357 DrDoS_LDAP and detected 2818 instances for the remaining classes as false positive with a precision rate of 52.06%, recall (74.78%) and f1-score (61.38%). Among the 11127 records classified as DrDoS_NetBIOS, we detected 10584 DrDoS_NetBIOS and 543 instances for the remaining classes as false positive with a precision rate of 97.23%, recall (95.12%) and f1-score (96.16%). Moreover, among the 11262 records classified as DrDoS_SNMP, we obtained 11162 DrDoS_SNMP and detected 100 instances for the remaining classes as false positive with a precision rate of 89.76%, recall (99.11%) and f1-score (94.20%). Finally, among the 11244 records classified as UDP-lag, we detected 6735 UDP-lag and attained 4509 instances for the remaining classes as false positive with a precision rate of 73.33%, recall (59.90%) and f1-score (65.94%).

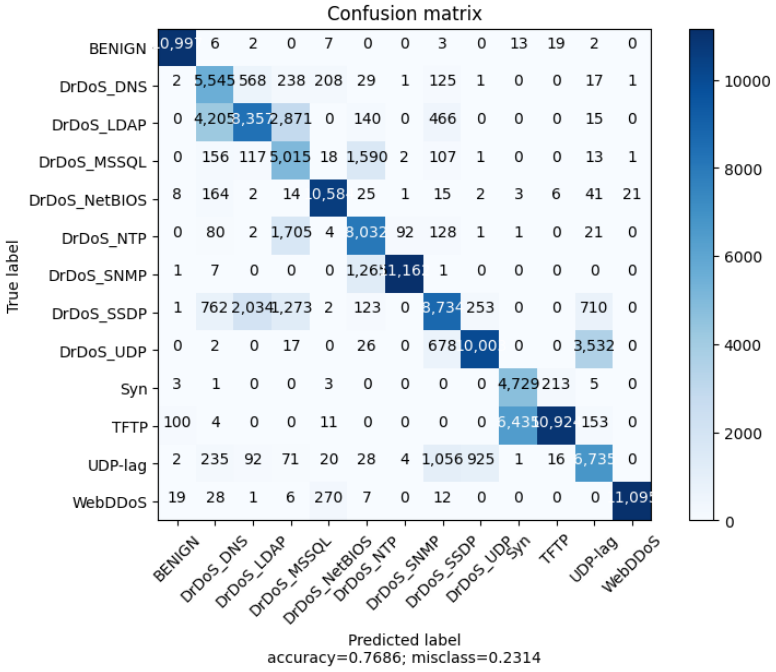


Figure 73: Confusion matrix for XGBoost model on the DB-DDOS

In addition, the random forest model is considered as the best classifier in detecting 5 classes; benign, DrDoS_NTP, DrDoS_SSDP, DrDoS_UDP, WebDDoS classes. The obtained results were extracted from the

confusion matrix shown in the figure [74]. Hence, among 11133 records classified as benign, we got 10997 benign and detected 136 instances for the remaining attack classes as false positive with a precision rate of 99.57%, recall (98.78%) and f1-score (99.17%). Moreover, among the 11265 records classified as DrDoS_NTP, we detected 8205 DrDoS_NTP and 3060 instances for the remaining classes as false positive with a precision rate of 78.61%, recall (72.84%) and f1-score (75.62%). Further, among the 11325 records classified as DrDoS_SSDP, we obtained 9069 DrDoS_SSDP and detected 2056 instances for the remaining classes as false positive with a precision rate of 61.23%, recall (80.08%) and f1-score (69.40%). Moreover, among the 11186 records classified as DrDoS_UDP, we identified 10061 DrDoS_UDP and 1183 instances for the remaining classes as false positive with a precision rate of 70.31%, recall (89.94%) and f1-score (78.92%). Finally, among the 11118 records classified as WebDDoS, we obtained 11097 WebDDoS and detected 21 instances for the remaining classes as false positive with a precision rate of 96.87%, recall (99.81%) and f1-score (98.32%).

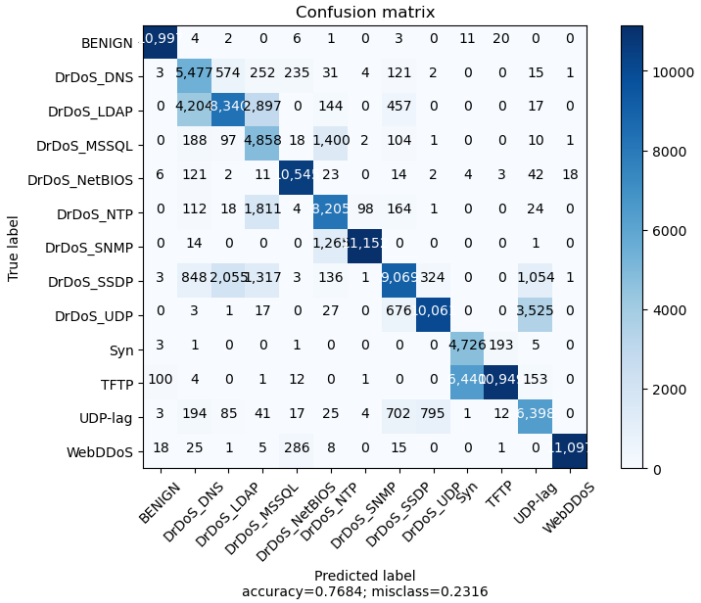


Figure 74: Confusion matrix for Random Forest model on the DB-DDOS

Furthermore, by employing the Adaboost model, we concluded that it was the best classifier in detecting the DrDoS_MSSQL and TFTP classes. The obtained results that were extracted from the confusion matrix shown in figure [75]. Hence, among the 11210 records classified as DrDoS_MSSQL, we obtained 5030 DrDoS_MSSQL and detected 6180 instances for the remaining classes as false positive with a precision rate of 70.23%, recall (44.87%) and f1-score (54.76%). Moreover, among the 11178 records classified as TFTP, we discovered 10963 TFTP and 215 instances for the remaining classes as false positive with a precision rate of 61.91%, recall (98.08%) and f1-score (75.91%).

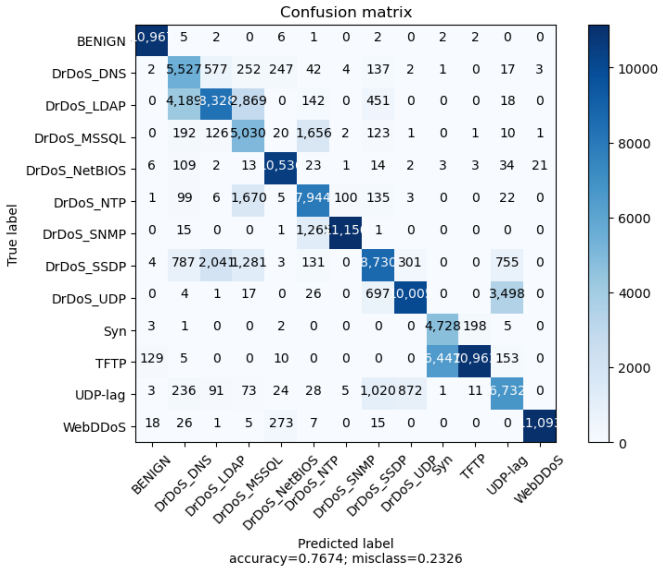


Figure 75: Confusion matrix for Adaboost model on the DB-DDOS

9.2.2.3 Evolving the Ensemble Techniques

In this part, we will present the results ensemble models. These classifiers performed some better results than the previous models. As revealed in the table [35], we found that the voting and the stacking models evolved the results and outputted the best performance measurement during the detection phase. Further, the Stacking model achieved the high performance by a precision of 79.77%, recall (77.07%), F1-score (76.28%) and an accuracy of 77.04%. The voting recorded a precision (79.63%), recall (76.89%), F1-score (76.09%) and an accuracy of 76.86%. However, the bagging resulted the low performance measurements among the proposed ensemble models by a precision of 79.76%, recall (76.89%), F1-score (76.07%) and an accuracy of 76.86%.

Model	Estimator	Precision Weighted Average	Recall Weighted Average	F1-Score Weighted Average	Accuracy ⇩
Stacking	- XGBoost - Random Forest - Adaboost	0.797752	0.77077	0.762804	0.770467
Voting	- XGBoost - Random Forest - Adaboost	0.796387	0.768985	0.760912	0.768688
Bagging	- RandomForest	0.797621	0.768978	0.760718	0.768688

Table 35: Results of the proposed ensemble models tested via the DB-DDOS

In addition, the table [36] displays the detection for each class as well as the coefficient measurements using our proposed ensemble models as follow:

Attack	Nb. Instance	Coefficient	Stacking		Voting		Bagging	
			Rate	Detect	Rate	Detect	Rate	Detect
BENIGN	11133	Precision	99.58%	10992	99.57%	**10997**	99.56%	**10997**
		Recall	98.73%		98.78%		98.78%	
		F1-score	99.16%		99.17%		99.17%	

DrDoS_DNS	11195	Precision	82.85%	**5588**	81.87%	5500	82.03%	5469
		Recall	49.92%		49.13%		48.85%	
		F1-score	62.30%		61.41%		61.24%	
DrDoS_LDAP	11175	Precision	52.01%	**8431**	52.02%	8362	51.94%	8347
		Recall	75.45%		74.83%		74.69%	
		F1-score	61.57%		61.37%		61.27%	
DrDoS_MSSQL	11210	Precision	73.77%	4847	72.30%	**4927**	72.65%	4888
		Recall	43.24%		43.95%		43.60%	
		F1-score	54.52%		54.67%		54.50%	
DrDoS_NetBIOS	11127	Precision	97.41%	**10622**	97.52%	10569	97.94%	10536
		Recall	95.46%		94.99%		94.69%	
		F1-score	96.43%		96.23%		96.29%	
DrDoS_NTP	11265	Precision	79.35%	**8221**	78.94%	8143	78.68%	8180
		Recall	72.98%		72.29%		72.61%	
		F1-score	76.03%		75.47%		75.52%	
DrDoS_SNMP	11262	Precision	89.73%	**11170**	89.71%	11154	89.64%	11164
		Recall	99.18%		99.04%		99.13%	
		F1-score	94.22%		94.14%		94.15%	
DrDoS_SSDP	11325	Precision	62.65%	8812	61.80%	8994	61.54%	**9042**
		Recall	77.81%		79.42%		79.84%	
		F1-score	69.41%		69.51%		69.51%	
DrDoS_UDP	11186	Precision	70.05%	**10118**	69.97%	10099	70.02%	10095
		Recall	90.45%		90.28%		90.25%	
		F1-score	78.95%		78.84%		78.85%	
Syn	11182	Precision	94.98%	4716	95.82%	**4720**	96.85%	4676
		Recall	42.17%		42.21%		41.82%	
		F1-score	58.41%		58.60%		58.41%	
TFTP	11178	Precision	61.90%	10912	61.99%	10945	61.92%	**11000**
		Recall	97.62%		97.92%		98.41%	
		F1-score	75.76%		75.91%		76.01%	
UDP-lag	11244	Precision	75.56%	**6657**	76.80%	6413	77.43%	6424
		Recall	59.20%		57.03%		57.13%	
		F1-score	66.39%		65.46%		65.75%	
WebDDoS	11118	Precision	97.23%	11094	97.01%	11098	96.72%	**11103**
		Recall	99.78%		99.82%		99.87%	
		F1-score	98.49%		98.40%		98.27%	

Table 36: A summary about the results for detecting each class with the coefficient measurements using our proposed ensemble models that tested via the DB-DDOS

After interpreting the table [36], the stacking shown as the best classifier for detecting 7 classes; the DrDoS_DNS, DrDoS_LDAP, DrDoS_NetBIOS, DrDoS_NTP, DrDoS_SNMP, DrDoS_UDP and UDP-lag. The results were obtained from the confusion matrix shown in figure [76]. Hence, among the 11195 records classified as DrDoS_DNS, we got 5588 DrDoS_DNS and detected 5607 instances for the remaining classes

as false positive with a precision rate of 82.85%, recall (49.92%) and f1-score (62.30%). Further, among the 11175 records classified as DrDoS_LDAP, we detected 8431 DrDoS_LDAP and 2744 instances for the remaining classes as false positive with a precision rate of 52.01%, recall (75.45%) and f1-score (61.57%). Moreover, among the 11127 records classified as DrDoS_NetBIOS, we obtained 10622 DrDoS_NetBIOS and detected 505 instances for the remaining classes as false positive with a precision rate of 97.41%, recall (95.46%) and f1-score (96.43%). Also, among the 11265 records classified as DrDoS_NTP, we identified 8221 DrDoS_NTP and 3044 instances for the remaining classes as false positive with a precision rate of 79.35%, recall (72.98%) and f1-score (76.03%). Finally, among the 11244 records classified as UDP-lag, we obtained 6657 UDP-lag and detected 4587 instances for the remaining classes as false positive with a precision rate of 61.91%, recall (98.08%) and f1-score (75.91%).

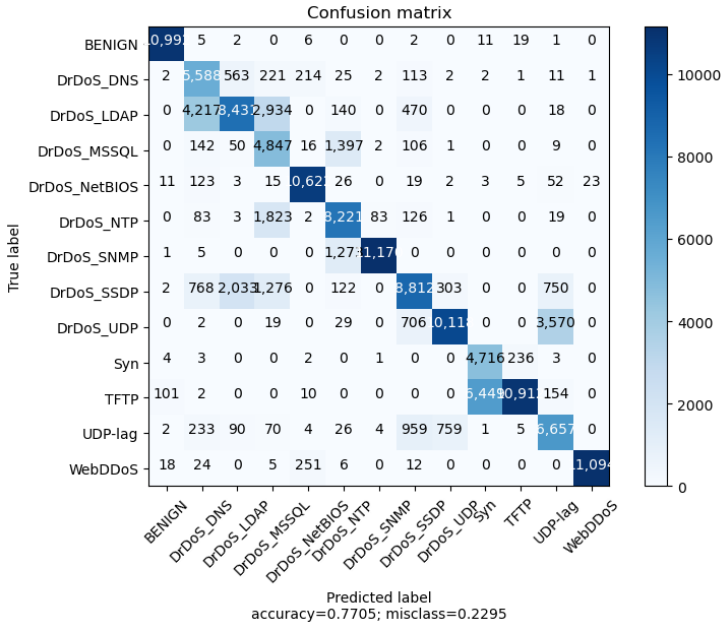


Figure 76: Confusion matrix for Stacking model on the DB-DDOS

In addition, the voting model is classified as the best classifier in detecting the benign, DrDoS_MSSQL and Syn classes. The obtained results were extracted from the confusion matrix shown in figure [77]. Thus, among 11133 records classified as benign, we found 10997 benign and detected 136 instances for the remaining attack classes as false positive with a precision rate of 99.57%, recall (98.78%) and f1-score (99.17%). Moreover, among the 11210 records classified as DrDoS_MSSQL, we detected 4927 DrDoS_MSSQL and 6283 instances for the remaining classes as false positive with a precision rate of 72.30%, recall (43.95%) and f1-score (54.67%). Finally, among the 11182 records classified as Syn, we obtained 4720 Syn and detected 6462 instances for the remaining classes as false positive with a precision rate of 95.82%, recall (42.21%) and f1-score (58.60%).

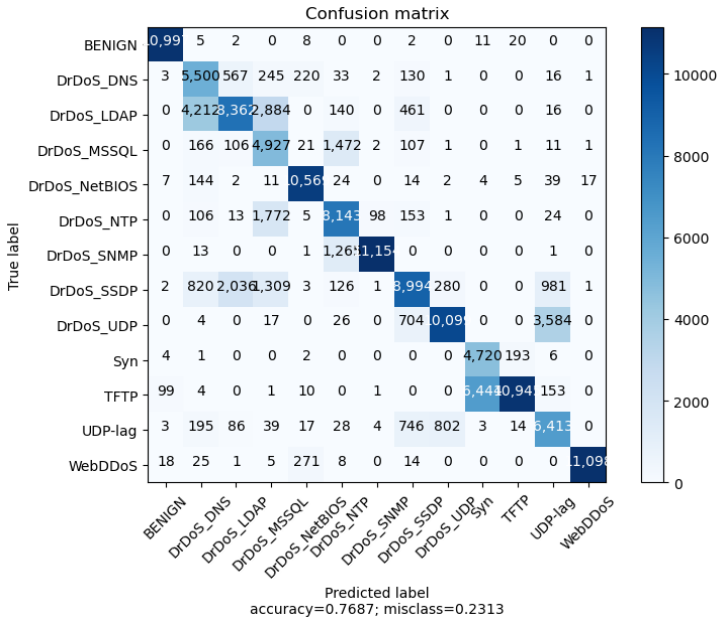


Figure 77: Confusion matrix for Voting model on the DB-DDOS

In addition, the bagging model is considered as the best classifier in detecting the benign, DrDoS_SSDP, TFTP and WebDDoS classes. The attained results were extracted from the confusion matrix shown in figure [78]. Hence, among 11133 records classified as benign, we obtained 10997 benign and detected 136 instances for the remaining attack classes as false positive with a precision rate of 99.56%, recall (98.78%) and f1-score (99.17%). Further, among the 11325 records classified as DrDoS_SSDP, we detected 9042 DrDoS_SSDP and 2283 instances for the remaining classes as false positive with a precision rate of 61.54%, recall (79.84%) and f1-score (69.51%). Moreover, among the 11178 records classified as TFTP, we obtained 11000 TFTP and detected 178 instances for the remaining classes as false positive with a precision rate of 61.92%, recall (98.41%) and f1-score (76.01%). Finally, among the 11118 records classified as WebDDoS, we detected 11103 WebDDoS and 15 instances for the remaining classes as false positive with a precision rate of 96.72%, recall (99.87%) and f1-score (98.27%).

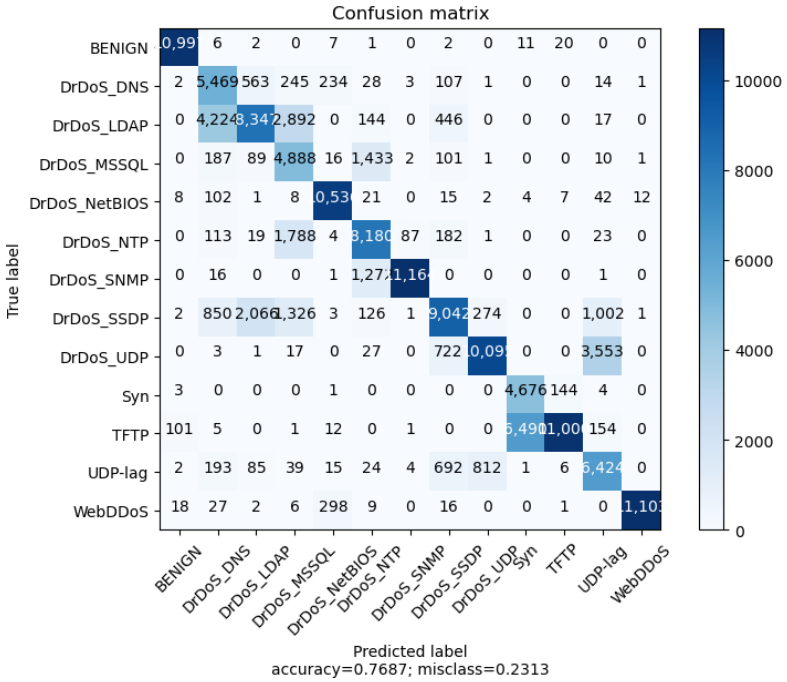


Figure 78: Confusion matrix for Bagging model on the DB-DDOS

9.2.2.4 Unsupervised Learning

In this part we will display the obtained results of the local outlier factor model that was deployed in the proposed hardware for detecting unknown coming traffics (novelty detection). As shown in figure [79] which represents the performance detection in the testing phase using the DB-DDOS, the results reveal that among 40000 records classified as advent attacks (outliers), we detected 38778 as attacks and 1222 as benign (inliers). Further, the model achieved an effective detection rate resulted by 96.94%.

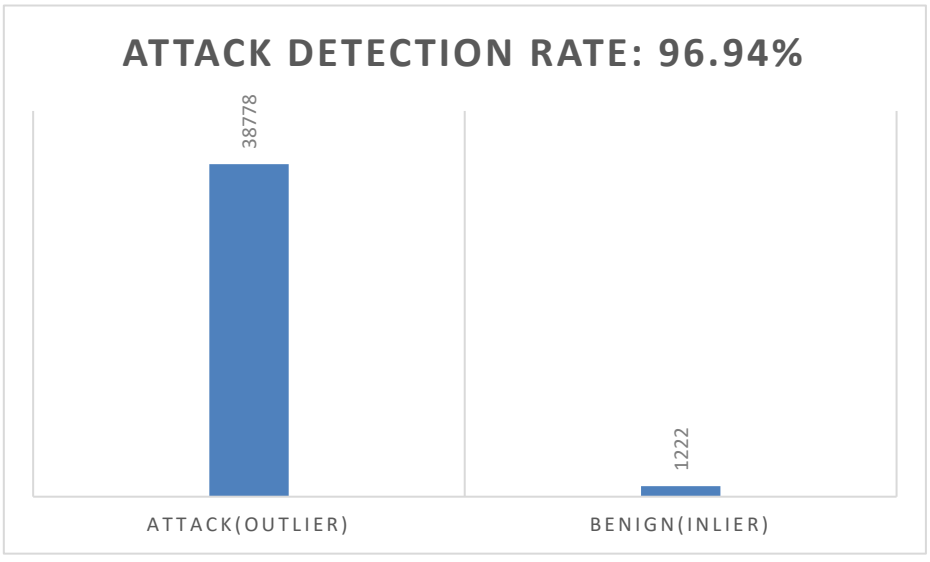


Figure 79: Performance measurement of the LOF model that deployed in the proposed hardware

9.3 General Discussion and Evaluation

In this part, we will discuss the performance measurements of the SIS-ID system using the DB-MALCURL and DB-DDOS. Our proposed security intelligent system shows high detection rates in forecasting both of the malicious URL and DDOS Attacks based on ten machine learning models.

Indeed, in order to prove our approach and to cross the limitation in detecting the malicious URL attacks, we present the performance of our system in the testing stage on tables [37,38]. These tables display the best models for detecting each attack with the achieved detection rate. Thus, after analyzing these tables, we conclude that the XGBosst model attained the best performance in detecting the defacement attack by a percentage of 99.41%, the voting for benign by 99.13%, KNN for malware by 98.57%, OneVsRest for phishing by 97.16% and finally the stacking for spam with 99.41%.

Attack type	Number of Instances	Voting	Stacking	XG-Boost	One Vs Rest	KNN
Defacement	1547	1535	1536	**1538**	1527	1527
Benign	1506	**1493**	1482	1489	1485	1469
Malware	1334	1312	1311	1313	1302	**1315**
Phishing	1589	1539	1541	1532	**1544**	1421
Spam	1364	1353	**1356**	1353	1350	1344

Table 37: The top models results for detecting each attack using the DB-MALCURL

Attack Type	Model	Detection of Instances	Detection Rate
Defacement	XGBoost	1538	99.41%
Benign	Voting	1493	99.13%
Malware	KNN	1315	98.57%
Phishing	OneVsRest	1544	97.16%
Spam	Stacking	1356	99.41%

Table 38: The detection rates achieved by the top models for each attack using the DB-MALCURL

Furthermore, our system proved an enhancement with worthy results comparing with the decision tree model that recorded the lowest result using the supervised learning as shown in table [39]. Thus, we evolved the performance of our system using the voting as a proposed ensemble model that improved 3% of Precision, 2.86% (Recall), 2.93% (F1-Score) and 2.97% (Accuracy) followed by stacking which was increased by 2.90% (Precision), 2.78% (Recall), 2.85% (F1-Score) and 2.88% (Accuracy).

Model	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Accuracy ⇩
Voting	0.98571	0.98552	0.9856	0.98529
Stacking	0.984885	0.98473	0.9848	0.98447

XGBoost	0.98465	0.98464	0.98463	0.98433
OneVsRest	0.982861	0.982111	0.982456	0.982016
Random Forest	0.982644	0.9818	0.98219	0.98174
Adaboost	0.981808	0.98101	0.98138	0.98093
OneVsOne	0.981371	0.980555	0.980926	0.980518
Bagging	0.979931	0.97903	0.97944	0.97902
KNN	0.964221	0.965574	0.964551	0.964033
Decision Tree	0.955805	0.956841	0.956249	0.955586

Table 39: Results of the different ML Techniques tested via the DB-MALCURL

Besides, after comparing the performance for detecting the Malicious URLs' in our SIS-ID with the results presented by the CIC Laboratory [3] as displayed in table [40], it is clearly obvious that our approach with the KNN model proved a better accuracy and crossed more than 1.4%, increased by 2.42% of precision and (2.55%) for recall while the decision tree model evolved as; 0.55% of accuracy, 1.58% of precision and 1.68% of recall. Finally, the random forest attained a better performance; accuracy by 3.17%, precision by 1.26% and recall by 1.18%.

	CIC Laboratory [3]			**SIS-ID**		
Model	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Random Forest	>0.95	0.97	0.97	0.98174	0.9826	0.9818
Decision Tree	>0.95	0.94	0.94	0.9555	0.9558	0.9568
KNN	>0.95	0.94	0.94	0.9640	0.9642	0.9655

Table 40: Comparative study between the SIS-ID tested via the DB-MALCURL and the CIC Laboratory [3]

On the other hand, we will discuss the performance measurement of our system using the DB-DDOS for detecting DDOS Attack. After analyzing the table [41,42] which presents the top models for the detection of each class, we conclude that the stacking model achieved the best performance in discovering five classes stated respectively; DrDoS_LDAP: 75.45%, DrDoS_NetBIOS: 95.46%, DrDoS_NTP: 72.98%, DrDoS_SNMP: 99.18% and DrDoS_UDP: 90.45%. Thus, we proved our methodology by applying the ensemble models. Moreover, the OneVsRest model attained the best in detecting the BENIGN: 98.79% and DrDoS_SSDP: 80.11% while the KNN model was the superior in detecting DrDoS_DNS: 73.37% and Syn: 46.13%. Further, The bagging model for the TFTP: 98.41%, WebDDoS: 99.87% and Adaboost for DrDoS_MSSQL of 44.87%. Finally, the XG-Boost achieves 59.90% in detecting the UDP-lag attack.

Model	**Stacking**	**Bagging**	**XG-Boost**	**One Vs Rest**	**Adaboost**	**KNN**
BENIGN	10992	10997	10997	10998	10967	10337
DrDoS_DNS	5588	5469	5545	5493	5527	8214
DrDoS_LDAP	8431	8347	8357	8355	8328	4763
DrDoS_MSSQL	4847	4888	5015	4905	5030	5011
DrDoS_NetBIOS	10622	10536	10584	10550	10536	10185
DrDoS_NTP	8221	8180	8032	8076	7944	7745
DrDoS_SNMP	11170	11164	11162	11156	11150	11114
DrDoS_SSDP	8812	9042	8734	9072	8730	6435

DrDoS_UDP	10118	10095	10003	10085	10005	9209
Syn	4716	4676	4729	4714	4728	5158
TFTP	10912	11000	10924	10953	10963	10076
UDP-lag	6657	6424	6735	6400	6732	6255
WebDDoS	11094	11103	11095	11099	11093	11036

Table 41: The top models results for detecting each attack using the DB-DDOS

Attack Type	Model	Detection of instances	Detection Rate
BENIGN	OneVSRest	10998	98.79%
DrDoS_DNS	KNN	8214	73.37%
DrDoS_LDAP	Stacking	8431	75.45%
DrDoS_MSSQL	Adaboost	5030	44.87%
DrDoS_NetBIOS	Stacking	10622	95.46%
DrDoS_NTP	Stacking	8221	72.98%
DrDoS_SNMP	Stacking	11170	99.18%
DrDoS_SSDP	OneVSRest	9072	80.11%
DrDoS_UDP	Stacking	10118	90.45%
Syn	KNN	5158	46.13%
TFTP	Bagging	11000	98.41%
UDP-lag	XG-Boost	6735	59.90%
WebDDoS	Bagging	11103	99.87%

Table 42: The detection rates achieved by the top models per each attack using the DB-DDOS

Additionally, our system proved an enhancement of the performance measurements with worthy results comparing to the KNN that recorded the lowest result using the supervised learning as shown in the table [43]. Therefore, we evolved our system using the stacking model which was improved as follow; 6.26% (precision), 4.55% (Recall), 4.29% (F1-Score) and 4.56% (accuracy).

Model	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Accuracy ⇩
Stacking	0.79775	0.77077	0.7628	0.77047
Voting	0.79639	0.76899	0.76091	0.76869
Bagging	0.79762	0.76898	0.76072	0.76869
XGBoost	0.79455	0.76894	0.76135	0.76863
Random Forest	0.79649	0.76865	0.76051	0.76837
OneVsRest	0.79609	0.76853	0.7604	0.76824
OneVsOne	0.79436	0.76819	0.76027	0.76788
Adaboost	0.79343	0.76771	0.7603	0.7674
Decision Tree	0.79298	0.76749	0.75975	0.76719
KNN	0.73518	0.72524	0.71987	0.72484

Table 43: Results of the different ML Techniques tested via the DB-DDOS

Furthermore, by comparing our approach in detecting the DDOS attack with the IDS performance related to the CIC Laboratory [3] seen in table [44], it is clearly shown that by employing the random forest model our system proved better measurements; precision up to 2.64%, recall of 11.87% and f1-score (7.05%). The improvement using the decision tree model can be shown respectively; precision by 1.30%, recall (11.75%) and F1-Score (6.97%).

	CIC Laboratory [3]			SIS-ID		
Model	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Random Forest	0.77	0.65	0.69	0.796485	0.768653	0.760505
Decision Tree	0.78	0.65	0.69	0.792984	0.76749	0.759745

Table 44: Comparative study between the SIS-ID tested via the DB-DDOS and the CIC Laboratory [2]

9.4 Hardware-Based Real-Time Simulation

In this part, we will discuss the results of the validation phase for our SIS-ID system that was examined using the DB-DDOS based on configured hardware on a real time stage in the faculty of technology. Due to the problematic issue of intrusion prevention system related to the difficulty of dealing with unknown upcoming traffics that may hold cyber-attacks over the network, we applied in this experiment with a target of implementing a cyber-security mechanism to avoid any attack that can threat the server as shown in figure [80].

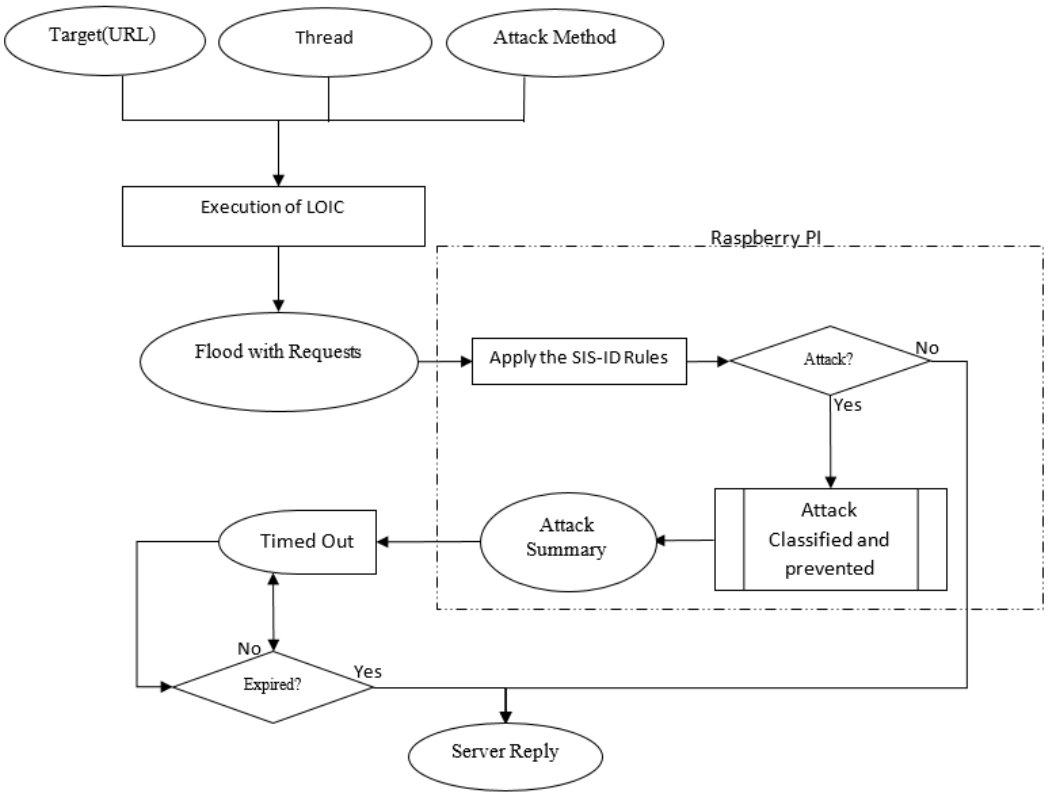


Figure 80: The general architecture of our SIS-ID hardware-based real-time simulation

9.4.1 Validation

In this section, we will display the attack simulation and the validation process of our hardware to avoid the denial of service attack on a real time stage. As presented in figure [80], the LOIC software were suggested to perform a denial of service attack (DOS) and then to be validated based on a raspberry pi as an intelligent security hardware using the local outlier factor model. Consequently, we selected the victim which is the domain name of the faculty of technology. The HTTP request is a method of attack, there were five threads to simulate that attack. Afterward, we executed a small DOS attack due to the university's restriction in terms of the applicable laws regarding the cybercrimes. On the other hand, the hardware was configured to capture the coming packets using the CICFLOWMETER in order to extract the corresponding features related to the flows. Thus, these packets have been produced and matched with our SIS-ID learning system according to its rules. Furthermore, as shown in figure [81], the hardware proved the efficiency in detecting the advent attack as well as preventing it. The observed attack lasted for 60 seconds from 12:40:08 pm to 12:41:08 pm and flooded the victim with several abnormal requests.

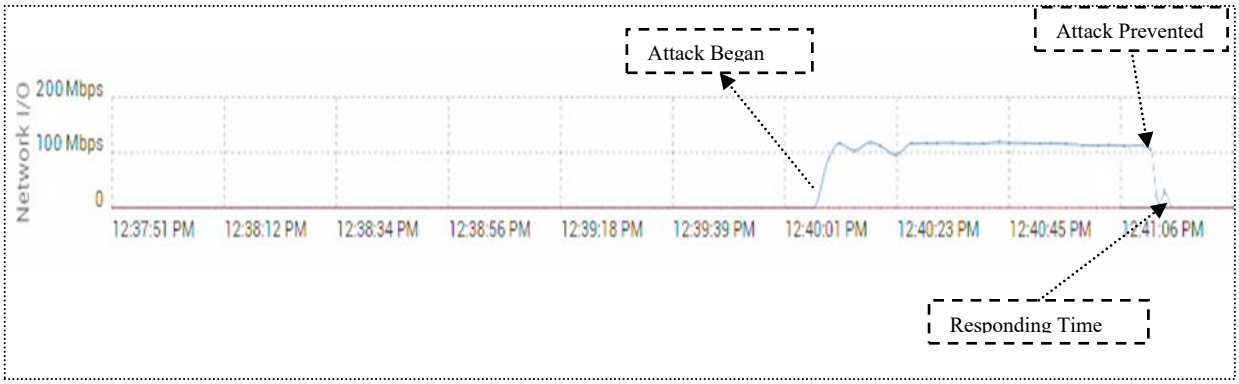


Figure 81: The efficiency in detecting the advent attack on real time stage

Afterwards, as displayed in figure [82], we will present some of the results of our raspberry pi during that period. Further, it achieved the prevention by employing the configured firewall within our hardware to avoid the detected attacks. Thus, our deployed system captured all the flows that contains several anomaly packets with the ability to identify the IP addresses related to the threat source. Thus, in this experiment, our security intelligent hardware prevented five consequence flows that were coming from the IP address "10.3.141.106" to the faculty web server which proved our dynamic rule protection. Ultimately, the attack has been identified and blocked as well as we delay it for 5 seconds as a suggested request timed out to indicate that the sever did not received any abnormal request within the designated period of time.

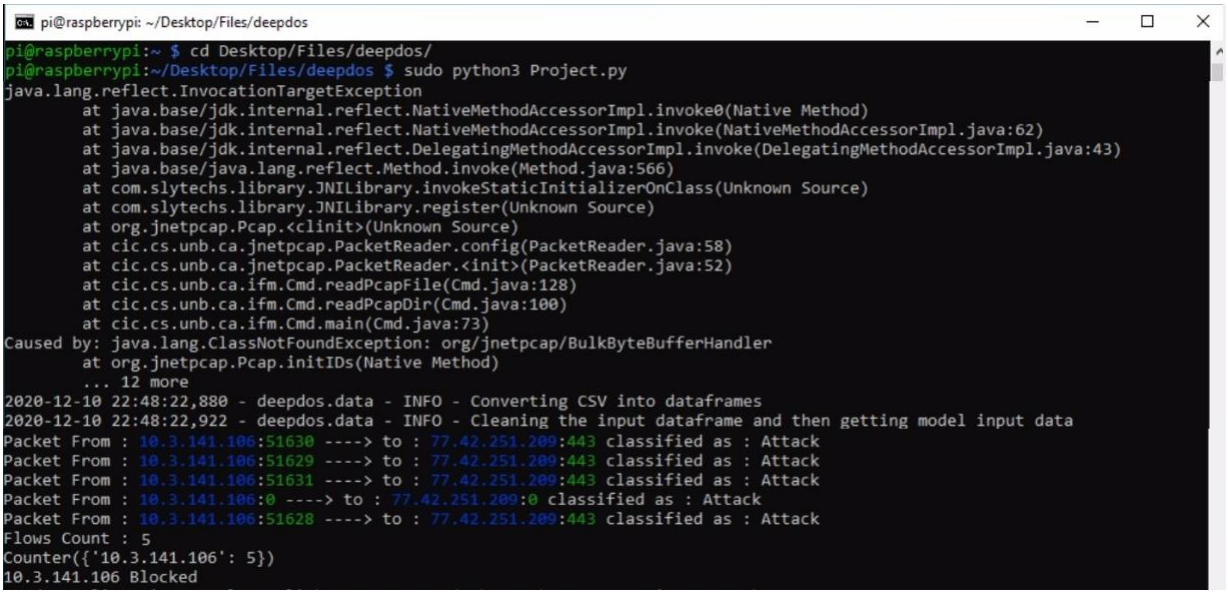


Figure 82: The result of our hardware for avoiding coming DOS attack

Conclusion Part III

In this part, we introduced the development of the general architecture of the "SIS-ID" using the machine learning techniques. We suggested all the materials used for detecting both of malicious URL and DDOS attacks. Therefore, we configured our datasets that were gathered from the Canadian Institute for Cyber-security. The DB-MALCURL used the ISCX-URL-2016 and DB-DDOS based on the DDOS2019 datasets. Moreover, we proposed our data engineering and features' method that was employed during the learning stage. We utilized the preprocessing stage with the data cleaning, under-sampling and the data transformation technique. In addition, the feature method was selected using the "recursive feature elimination" technique. Consequently, we applied eleven machine learning techniques; the supervised learning, unsupervised learning and ensemble techniques. Afterwards, we suggested a learning optimization method that uses the Hyperparameter and the GridSearchCV techniques. It achieved particularly an evolving of the performance measurements.

Consequently, we explained the obtained results and the performance of our proposed SIS-ID for detecting the latest malicious and DDOS attacks. By examining the DB-MALCURL, the voting model achieved a high performance among all the models with a precision of 98.57%, recall (98.55%), F1-score (98.56%) and an accuracy of (98.52%). In addition, this model proved an enhancement with worth results comparing with our lowest achieved result (Decision Tree) that improved 3 % of precision, 2.86% (Recall), 2.93% (F1-Score) and 2.97% (Accuracy). Furthermore, by testing the DB-DDOS, the stacking model achieved the highest performance among all the models by a precision of 79.77%, recall (77.07%), F1-score (76.28%) and an accuracy of 77.04%. Hence, this model proved an enhancement in the results comparing to our lowest achieved result (KNN) which improved as shown respectively; 6.26% (precision), 4.55% (Recall), 4.29% (F1-Score) and 4.56% (Accuracy). Moreover, several measurements were discussed concerning the attained detection rate for each model in terms of every examined attack using SIS-ID. Afterwards, our proposed SIS-ID has been validated as an intrusion prevention hardware with the efficiency of avoiding a denial of service attack (DOS) on a real time stage.

References Part III

[1] MSI. Mamun, MA Rathore, AH. Lashkari, N. Stakhanova, AA. Ghorbani. (2016). "Detecting Malicious URLs Using Lexical Analysis", Network and System Security. doi:10.1007/978-3-319-46298-1_30.

[2] Sharafaldin, AH. Lashkari, S. Hakak, AA. Ghorbani. (2019). "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", 2019 International Carnahan Conference on Security Technology (ICCST). doi:10.1109/ccst.2019.8888419.

[3] AH. Lashkari, G. Draper-Gil, MSI Mamun, AA. Ghorbani. (2017). "Characterization of Tor Traffic Using Time Based Features", In the proceeding of the 3rd International Conference on Information System Security and Privacy, SCITEPRESS, Porto, Portugal.

[4] Pedregosa. (2011). "Scikit-learn: Machine Learning in Python", JMLR 12, pp. 2825-2830.

[5] R. Mohammed, J. Rawashdeh, M. Abdullah. (2020). "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results", 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 243-248, doi: 10.1109/ICICS49469.2020.239556.

[6] S. Mani, A. Ozdas, C. Aliferis, HA.Varol, Q. Chen, R. Carnevale. (2014). "Medical Decision Support Using Machine Learning for Early Detection of Late-Onset Neonatal Sepsis", J Am Med Informatics Assoc [Internet]. 21(2):326–36.

General Conclusion and Future Work

The security of computer networks and systems are considered as a critical challenge. A major of cyber scientist's collaboration had occurred to overcome this approach and to offer the best efficient security standards. Unfortunately, with the evolution of new threats, the existing tools may be not adequate to detect and avoid them.

The essential problems for existent security assessment and intrusion detection systems using machine learning can be stated as follows:

- The difficulty in applying the security evaluation techniques in terms of data privacy imposed by companies.
- The deficiency in the dataset for IDSs.
- The limitation of IDS ability for detecting the latest and most common cyber-attacks.
- The validation of the IDSs performance.
- The IDS deployment on real time attack detection.

Therefore, the main objective of our thesis was to enhance the applied security mechanism and to propose an intelligent security system against the most important cyber-attacks that was accomplished summarized in the below steps:

We proved a cyber-security assessment for the applied mechanism in the Lebanese university that achieved the essential safety standards. We applied the penetration testing technique according to some of the 10 OWASP security standards. Afterward, we discovered several vulnerabilities in many web pages concerning the SQL injection (SQLi), cross-site scripting (XSS) and sensitive data exposure occurred in the CCNE system in the Faculty of Technology. Therefore, we presented the most significant security solutions against those kinds of attacks in order to guarantee the protection, as well as to provide the ability of fixing the detected vulnerabilities.

Furthermore, we presented a motion to detect the visitor's behavior based on the web usage mining technique. Hence, the deep log analyzer and the security analysis tool were employed to analyze the web server access log. Then, we identified several approaches relevant to the visitor's activities and the irregular actions. Accordingly, the tools achieved the identification of the visitor's behavior in terms of the activity that showed related summaries for the occurred events, accessed resources, visited links, and hits' error etc... . On the other hand, since the developed security analysis tool is a rule based detection technique, we built a data base that includes the most important attacks' pattern. The tool discovered 15 malicious URLs related to cyber-attacks were they were identified as; 10 of SQLi, 3 for XSS and 2 related to the path traversal. Ultimately, our experimental results will be effective and helpful for the web analysts and security administrators to improve the usability and the security stability of the web server as well as their resources.

By going forward to the intelligent detection, we developed a host based intrusion detection system (HIDS) using the text mining technique. Hence, to overcome the problem of the lacked reliable dataset, we constructed our dataset that includes 5997 textual records of malicious URLs related to the HTTP GET request with the ability of matching it with the university web server log file. Afterword, we faced a complicated hypothesis related to the features representation methods that were came from this kind of data. To overcome these problems, we proposed the Doc2Vec model as a feature representation method that were used to learn embedded word (URL structure) to produce the vector of features in the learning stage. Eventually, four different machine learning techniques (KNN, SVM, Decision Tree and MLP) were tested to fulfill the best accurate classification model to detect the web server cyber-attacks. Our HIDS proved the effectiveness in detecting the SQLi, XSS and directory traversal attacks. Subsequently, the experimental results showed that the SVM classifier accomplished the highest rate in detecting the Path-Traversal attack by 95.41% while the MLP model achieved the highest detection rate for both SQLI and XSS stated respectively by 94.69%, 84%. Furthermore, the MLP recorded the best accuracy of 90.67% Subsequently, the KNN attained the second accuracy score with 88.17 % followed by the decision tree with 86.08 %. Finally, the SVM retrieved the lowest accuracy rate with 82.67%.

From another perspective, due to the noticed problem in the HIDS related to the limitation in detecting networks' attack, we suggested to develop a security intelligent system called "SIS-ID" using the machine learning techniques. The system had the potential to detect both of malicious URLs and DDOS attacks. Therefore, we configured the DB-MALCURL and DB-DDOS datasets that were extracted from the Canadian Institute for Cyber-Security (CIC).

Hence, we employed the preprocessing stage for those data bases, it helped to improve the quality of data. Moreover, it promotes the extraction of meaningful knowledge related to the attacks using several techniques; the data cleaning to remove and handle missing values, under-sampling to produce balanced classes and the data transformation technique to encode the categorical features for making them in the same scale of magnitude. In addition, the feature method was proposed using the "recursive feature elimination" technique. Consequently, we applied several machine learning techniques; the supervised learning (SL) and ensemble techniques (ET). Hence, we applied our proposed optimization method during the learning stage; the hyperparameter was utilized to control the learning stage as well as the cross validation and grid search technique "GridSearchCV" were conducted to fit the model in order to attain the powerful affected parameters.

Thereafter, after applying our classification models, the experimental results of our proposed SIS-ID for detecting the malicious URLs and DDOS attacks showed an effective enhancement for the attacks' detection rate and accurate performance.

By examining the DB-MALCURL, we conclude that the XGBosst model achieves the best performance in detecting the defacement attack by a percentage of 99.41%, the voting for Benign by 99.13%, KNN for

Malware by 98.57%, OneVsRest for Phishing by 97.16% and the stacking for spam by 99.41%. In addition, our system proved an enhancement in the accuracy rate comparing with the lowest result attained by the decision tree model. Thus, according to our suggested ensemble model and after applying our learning optimization method, the voting model was improved by 2.97% followed by the stacking model that was increased by 2.88%. Moreover, the voting model was classified as the best classifier with a precision of 98.57%, recall (98.55%), F1-score (98.56%) and an accuracy (98.52%).

Afterwards, by testing the DB-MALCURL, the SIS-ID using the stacking model, we achieved the best performance in detecting five classes stated respectively; DrDoS_LDAP: 75.45%, DrDoS_NetBIOS: 95.46%, DrDoS_NTP: 72.98%, DrDoS_SNMP: 99.18% and DrDoS_UDP: 90.45%. Moreover, the OneVsRest model attained the top model in detecting the BENIGN: 98.79% and DrDoS_SSDP: 80.11%, while the KNN model was the superior in detecting DrDoS_DNS: 73.37% and Syn: 46.13%. Furthermore, the Bagging model for the TFTP is 98.41% and WebDDoS was 99.87%. After that, the Adaboost model for DrDoS_MSSQL was 44.87% and finally the XG-Boost achieved 59.90% in detecting the UDP-lag attack. Additionally, we demonstrated an improvement in the accuracy rate comparing with lowest result attained by the KNN model. The stacking model increased by 4.56% followed by the voting model that was improved by 4.38%. In addition, the tacking model achieved the highest performance results among all the models by a precision of 79.77%, recall (77.07%), F1-score (76.28%) and an accuracy of 77.04%.

On other hand by comparing our SIS-ID with the performance measurement resulted from the CIC laboratory and by using the DB-MALCURL, the SIS-ID based on the KNN model proved a better accuracy and crossed more than 1.4%, increased by 2.42% of precision and 2.55% for recall. The decision tree model attained an increasing of results as shown respectively; accuracy: 0.55%, precision: 1.58% and recall: 1.68%. Finally, the random forest reached the higher risen by 3.17% for accuracy, 1.26% for precision and 1.18 for recall. However, by examining the DB-DDOS, with the use of the random forest model, our system proved favored results; precision up to 2.64%, 11.87% of recall and 7.05% for F1-Score while the improvement using the decision tree model can be shown respectively; precision: 1.30%, Recall: 11.75% and F1-Score: 6.97%.

Eventually, our proposed SIS-ID was validated as an intelligent hardware using the LocalOutlierFactor as unsupervised model. We deployed our system in a raspberry pi where the CICFLOWMETER was configured to capture the coming packets in order to extract the corresponding features of the flows. Thus, the coming packets were reproduced to be matched with our SIS-ID learning system. Afterward, we simulated a denial of service attack using the LOIC software. In this experiment, we selected the victim which is the domain name of the Faculty of Technology to validate the prevention on a real time stage. Thus, in this experiment, our security intelligent hardware prevented 5 consequence flows coming from the IP address "10.3.141.106" to the faculty web server. Ultimately, our intelligent raspberry pi showed an efficiency in avoiding the advent attack as well as providing a dynamic rule for the protection.

In addition, the development of a powerful intrusion detection system (IDS) is thought to be an endless field of research. In our thesis, we have offered the contributions to propose an intelligent IDS based on the machine learning techniques for identifying and preventing the contemporary cyber-attacks. Our motivation in this research field forced us to extend and improve our suggested defense level. Thus, we will present a potential perspective for this research work:

Concerning the host based intrusion detection system (HIDS), we would like to raise our recommended dataset by adding more recorded attacks to enhance the learning phase. Further, we will upgrade it to be adopted with other types of logs. On the other hand, we will improve the SIS-ID system to overcome the issue related to the false alarms rates, memory consumption and the processing time. Then, we would like to expand the learning stage of the system using specific databases for new attacks. Regarding the intelligent hardware perspectives, we recommend to add some statistical features to analyze the attacker's behavior. Moreover, we will work on evolving the SIS-ID to identify and analyze the coming attack type as well as validate it in other environment like the diverse IOT cyber-attacks.

Publications

i. AK. Kassem, SA. ARKOUB, B. DAYA, P. CHAUVET. (2019). "A Survey of Methods for the Construction of an Intrusion Detection System", Artificial Intelligence and Applied Mathematics in Engineering Problems. ICAIAME 2019. Lecture Notes on Data Engineering and Communications Technologies. Springer, Cham, pp. 211-225, Volume 43.

ii. AK. Kassem, M. El¬Sayed, B. Daya, P. Chauvet, M. Saadeldine. (2019). "A New Feature Representation Method for Intrusion Detection System", 4th International Conference on Computational Mathematics and Engineering Sciences.

iii. AK. KASSEM, B. DAYA, P. CHAUVET. (2018). "A Proposed Methodology on Predicting Visitor's Behavior Based on Web Mining Technique", (IJACSA) International Journal of Advanced Computer Science and Applications, No. 12, Volume 9.

iv. AK. Kassem, A. AL HAJJAR, B. Daya and P. Chauvet. (2018). "A Proposed Methodology for Cyber Security Mechanism According to the Most Popular Detected Attacks for University Web Application", 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 215-219, doi: 10.1109/WorldS4.2018.8611626.