DATA MINING FOR WEB APPLICATION ATTACKS ANALYSIS

Doktora Tezi

Mohammed BABIKER

Eskişehir 2019

DATA MINING FOR WEB APPLICATION ATTACKS ANALYSIS

Mohammed BABIKER

Ph.D. Dissertation

Computer Engineering Department
Supervisor: Prof. Dr. Yaşar HOŞCAN
(Co-Supervisor: Assist. Prof. Dr. Enis KARAARSLAN)

Eskişehir
Eskisehir Technical University
Institute of Graduate Programs
August 2019

FINAL APPROVAL FOR THESIS

This thesis titled “Data Mining for Web Application Attacks Analysis” has been prepared and submitted by Mohammed Babiker in partial fulfilment of the requirements in “Eskişehir Technical University Directive on Graduate Education and Examination” for the Degree of Doctor of Philosophy (PhD) in Computer Engineering Department has been examined and approved on 22/08/2019.

Committee Members	**Title, Name and Surname**	**Signature**
Member (Supervisor)	: Prof. Dr. Yaşar HOŞCAN	.....
Member	: Assoc. Prof. Dr. Cihan KALELİ	.....
Member	: Assist. Prof. Dr. Ahmet ARSLAN	.....
Member	: Assist. Prof. Dr. Esra N. YOLAÇAN	.....
Member	: Assist. Prof. Dr. Mehmet KOÇ	.....

.....

Prof. Dr. MURAT TANIŞLI

Director of Institute of Graduate Programs

ABSTRACT

DATA MINING FOR WEB APPLICATION ATTACKS ANALYSIS

Mohammed BABIKER

Department of Computer Engineering
Eskisehir Technical University, Institute of Graduate Programs, August 2019

Supervisor: Prof. Dr. Yasar HOSCAN
Co-Supervisor: Asst. Prof. Dr. Enis KARAARSLAN

The greatest challenge of web application attack investigations is dealing with the huge amount of data, the level of complexity of the web application integration technology and the huge number of sophisticated attacks. There is a need for more intelligent and convenient techniques for better web application attack investigations. Data mining and machine learning might aid to analyze and investigate web application attacks. This study proposes a framework for web application attacks forensic by applying data mining models. To date, there has not been an adequate approach to apply data mining technique to find web attack digital evidence. In this study, a hybrid approach is proposed for the feature selection and a data mining framework is proposed for web application forensics process.

This framework has been validated by experimental measurements on three different web attack datasets. The results show that our proposed framework can find evidence with high recall, high accuracy and low error rates. We believe that the results presented herein may help to improve accuracy and recall of data mining techniques; particularly in the field of web attack investigation. The solution that will utilize this framework may help in digital forensic investigation and provide aid to experts and law enforcement in finding digital evidence significantly more productively and quicker.

Keywords: web application attacks, digital forensic analysis, digital forensic framework, feature selection, digital evidence

ÖZET

WEB UYGULAMA SALDIRILARI ANALIZI IÇIN VERI MADENCILIĞ

Mohammed BABIKER

Bilgisayar Mühendisliği Anabilim
Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ağustos 2019

Danışman: Prof. Dr. Yasar HOŞCAN
Eş-danışman: Dr.Öğr. Üyesi Enis KARAARSLAN

Web uygulama saldırı araştırmalarındaki büyük zorluk ve meydan okuma; büyük miktardaki veri ile uğraşma, web uygulama entegrasyonu teknolojisinin karmaşıklığı ve büyük sayıda sofistike saldırılardır. Daha iyi web uygulama saldırı incelemesi için daha zeki ve kullanışlı tekniklerine ihtiyaç vardır. Veri madenciliği ve makine öğrenme teknolojileri;web uygulama saldırılarını analiz ve soruşturmak için yardımcı olabilir. Bu çalışma web uygulama saldırıları adli analizi için veri madenciliği modelleri uygulayan bir çerçeve önermektedir. Web saldırı dijital kanıtı bulmak için veri madenciliği tekniği uygulayan yeterli bir yaklaşım bugüne kadar bulunmamaktadır. Bu çalışmada, özellik seçimi ve veri madenciliği çerçeve web uygulaması adli işlemi için bir hibrit yaklaşım önerilmektedir.

Bu çerçeve üç farklı web saldırı veri seti üzerinde deneysel ölçümler ile onaylanmıştır. Sonuçlar teklif ettiğimiz çerçevenin; yüksek geri çağırma, yüksek doğruluk ve düşük hata oranları ile delil bulabildiğini göstermektedir. Burada sunulan sonuçların; özellikle, web saldırı soruşturması alanında makine öğrenme tekniklerinin doğruluğunu ve geri çekmesinin geliştirilmemize yardımcı olabileceğine inanıyoruz. Bu metodoloji kullanacak bir çözüm dijital adli soruşturmaya yardımcı olacak; uzmanlar ve kolluk güçlerine dijital delilleri bulmada önemli ölçüde daha verimli ve daha hızlı yardım sağlayabilecektir.

Anahtar Kelimeler: web uygulama saldırıları, dijital adli analiz, dijital adli çerçeve, özellik seçimi, dijital kanıt

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Yasar Hoscan for the support of my Ph.D study and research, I have learned from him many management skills that helped me in this work. It is with immense gratitude that I acknowledge the support and help of him.

Throughout the journey of this thesis I have received a great deal of support and assistance from my co-supervisor, Dr. Enis Karaarslan, all thanks and appreciation for his patience, motivation, enthusiasm, and immense knowledge. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my supervisors, I would like to thank the rest of my thesis committee: Dr. Ahmet Arslan, Dr. Cihan Kaleli, Dr. Esra Yolaçan and Dr. Mehmet KOÇ, for their encouragement, insightful comments, and hard questions, their valuable contributions helped me to complete this study at its best.

This thesis is heartily dedicated to my father “If I could write a story, it would be the greatest ever told. I'd write about my father, for he had a heart of gold.”, and brother for their emotional supports and inspired me to reach my dreams.

I owe my deepest and special gratitude to my angel Zehra.

22/08/2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

.....

Mohammed BABIKER

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABBREVIATION LIST

AIDS	: Application-based Intrusion Detection System
CFS	: Correlation-based Feature Selection
EDADT	: Efficient Data Adapted Decision Tree
HFSN	: Hybrid Feature Selection with Naive Bayes
HFST	: Hybrid Feature Selection with Decision Tree
HTTP	: Hyper Text Transfer Protocol
IDS	: Intrusion Detection System
IP	: Internet Protocol
IPS	: Intrusion Prevention System
IT	: Information Technology
KD	: Knowledge Discovery
MAE	: Mean Absolute Error
MITM	: Man in the middle
OWASP	: Open Web Application Security Project
RMSE	: Root Mean Square Error
SMO	: Sequential Minimal Optimization
SQL	: Structured Query Language
SVM	: Support Vector Machines
TCP	: Transmission Control Protocol
URL	: Uniform Resource Locator
WAF	: Web Application Firewall
XSS	: Cross Site Scripting

1 INTRODUCTION

Web application attacks are one of the most widely ongoing information security issues that have advanced rapidly, to become widely acknowledged. The web application has long been a major gateway of e-services in a wide range of fields and application. Therefore, the web application has become a major target of attacks. There is a large volume of published studies and reports describing the various types and high amount of attacks targeting web application. Symantec's 2019 Internet Security Threat Report [1] reveals that there is around 1 million web attack per day to 40 million attacks per month based on 123 million sensors. These sensors receive attacks every second and they are distributed in 157 countries. A survey from Imperva [2] based upon 1,200 high qualified IT practitioners and decision-makers; found that the major barrier in building effective defense is the enormous amount of data to be analyzed. A consequence of the high amount of web attacks, there is a huge investment in building and buying security tools, 63% of this investment focus on web application firewalls. Data collected over 12 months and across 10,000 targets show 87% of websites suffer from medium-security vulnerabilities while 46% have high-risk vulnerabilities [3].

Digital investigation plays a major role in countering attacks and searching for digital evidence. This serves as a deterrent to attackers and provides information on the nature and new forms of attacks. Regarding digital forensic, the tradition forensic investigation standing helpless in front of the vast amount of data and new high complexity level attacks. This pushes government agencies and companies to use dozens of investigators, wasting resources to get a piece of evidence. Likewise, the vast majority of conventional forensic frameworks are ineffective in finding digital evidence in large-scale data. In another hand, data mining techniques are known to be suitable for analyzing large amounts of data, which could extract interesting, useful and new patterns or knowledge. Data mining may help digital investigators in decision-making if it is applied in web attacks forensic. This research focuses on the development of a novel comprehensive web application forensic framework which combines the data mining process and models for a better process.

1.1 Problem Statement

Web application attacks are one of the most widely used attacks on Internet. Despite the importance and great risks of such attacks, the countermeasure techniques of these attacks have become ineffective due to the size and the complexity of web application data. As a consequence of the ineffective detection and prevention mechanisms, many new sophisticated attacks have taken place. As a result, it is difficult to find digital evidence indicating the attackers or learn the way they carry out their attacks. The security solutions such as Intrusion Detection Systems (IDS) and firewall, provide millions of logs files on a daily basis. It is difficult to find out evidence or clue considering the huge size of those log files. Therefore, traditional digital forensic techniques are insufficient in web application attacks.

1.2 Research Aim

The aim of this research is to gain an understanding of how web application attacks occur by using data mining tasks in the analysis of web application attacks. Applying data-mining algorithms; could lead to finding digital evidence more efficiency. Providing elements to build a novel comprehensive forensic framework can help in combating those attacks, forming best practices and guidelines for the forensic investigation. The following objectives have been identified of paramount importance in helping to achieve the aim above:

- Conducting a series of experiments on the web attacks datasets, to examine the datasets and gain a solid understanding of attacks, to select the appropriate data-mining algorithms.
- Proposing a novel approach for feature constructions and data preprocessing.
- Proposing a novel approach for feature selection to enhance the data mining algorithms accuracy and performance.
- Designing a novel web application forensic framework that combines best web application forensic techniques which are based on data mining to provide a high-level solution to cope with various attacks intelligently and adaptively.

- Applying and testing the web application forensic framework in order to find a hidden and interesting pattern which makes the framework able to gather information, collect evidence, examine evidence in addition to analysis and interpretation of collected information.

1.3 The Research Question:

"How a Framework Based on Data-mining be effectively designed to analyses web application attacks for forensic purposes?"

Supporting questions

- What are the requirements and methodology for developing web attacks forensic framework based on data mining?
- What are the current most dangerous attacks against web application?
- Which data-mining algorithms should be implemented?
- Which steps should be taken with the data-mining model to ensure it's consistent with digital forensics?
- What are the methods for web attack`s data collection to run the experiment?
- How the collected data pre-processed?
- Which considerations should be considered for validating the models?

1.4 Summary of Contributions

Our main contribution lies in the novel comprehensive forensic framework which provides a data mining technique based solution in investigating and tracking web application attacks. To the best of our knowledge, according to the world wide literature review; most of the available digital forensic frameworks have drawbacks. Some of the frameworks are just too much far from the technical aspects. Moreover, most of them depend on traditional methods of digital forensic. The traditional techniques do not offer intelligent or automated approaches. There is also a focus on the legal aspect rather than the analytical aspect of digital

evidence. In contrast, our framework is based on data mining models for decision-making. The proposed framework has the following properties:

- The proposed framework has increased the recall and accuracy of detection attacks.
- The framework was successful in reducing irrelevant features and finding optimal features to investigate web application attacks. This property contributes to building machine learning models with more relevant results, low error rate, high accuracy and better performance.
- The proposed framework fulfilled the potential requirements to be considered as a forensic technique in finding the digital evidence of web attacks.
- This framework contains a novel hybrid approach which could be considered as an effective feature selection approach in building machine learning and data mining models. This framework may be a useful aid for the intelligent automation system of the web attack forensics.

The research may contribute in enhancing the body of research for security and forensic of the web application attacks by:

- Developing and delivering new guidelines, best practices and techniques to enable government agencies, police and investigators to defend, mitigate and secure current and future web, e-government, e-business and infrastructure against cyber-attacks.
- Conduct and support technology transition and studying the latest state of art methods of the attacker. This research could create effective techniques to protect web applications, combats hackers and provides a unique spotlight on forensic of the web application attacks. Using a wide range of data mining techniques for research purposes will lead and coordinate research and development (R&D) among the community, which includes cybersecurity researchers, digital forensic researchers and investigators, government agencies, the private information security sector and international society.

Some papers have been published based on this thesis. Those papers can be listed as follows

- The literature review and background chapter contributed in a conference paper: Web Application Attack Detection and Forensics: A Survey. The conference location was in Antalya, Turkey. And the paper added to IEEEexplore [138].
- The proposed framework chapter contributed in a journal paper: A hybrid feature-selection approach for finding the digital evidence of web application attacks. The paper accepted in Turkish Journal of Electrical Engineering & Computer Sciences (Turk J Elec Eng & Comp Sci.).

1.5 Organization of the Dissertation

This research study is presented in six chapters. Chapter 1 includes an overview of the topic, statement of the problem, purpose and the aim of the study, significance and contribution of the study including the publication list. Chapter 2 presents a background of the study, which includes web application architecture theory, web application attacks, detection techniques of web attacks, machine learning, data mining, digital and web application forensic. Chapter 3 describes the methodology used for this research study. It includes the research method type, data collection, data pre-process and experiment setup. Chapter 4 presents the proposed framework and the novel hybrid approach. Chapter 5 provides the experiments and the results. Chapter 6 presents a discussion of the findings and future works.

2 BACKGROUND

Understanding the basic structure and architecture of the web application with its components and protocols will lead to an understanding of security vulnerabilities. Weaknesses on web application architecture which cause various kind of attacks are investigated. In this chapter, the most important attacks related to web application are reviewed. Thus, it aimed to provide an insight into features of web application attacks that can be used in the investigation. Moreover, learning web application attacks types may lead to clarify the places of a piece of information and that can be a clue of digital evidence. The definition of digital forensic investigation, as well as web forensic is presented. A comprehensive review of data mining and machine learning algorithms, as well as critique and overview of weakness and strengths of the previous studies are provided.

2.1 Web Application Architecture

Web application architecture is a framework describes the interactions between the main components of web application and integration technologies such as client-side, server-side code, middleware systems, application logic, and database back-end hosting. A web application can be loosely described as an application developed by adopting the web paradigm and accessed over the Internet [4]. Web applications are built based on client-server architecture. The client-server architecture consists of two main components. The server listens for requests and responds to the client. The client makes a request to the server in order to receive services and resources.

Most web applications rely on Hypertext Transfer Protocol (HTTP) to exchange information. HTTP is a simple stateless application layer protocol uses request respond model [5]. The HTTP protocol is stateless, the only information available to the server about a request is contained within that request. State optionally may be maintained as session information by server-side applications, however, the reference to this information must exist in the request [6]. Figure 2.1 reveals HTTP Response/Request in client/server model. HTTP request was divided into request headers and optional request body, both are binary data. Likewise, HTTP response was divided into response headers and body. The client sends the

request to the server to retrieve data with fields shown in Table 2.1. Once the request was received by the server, it was necessary for the server to reply by response contains the information requested by the client with fields in Table 2.2.

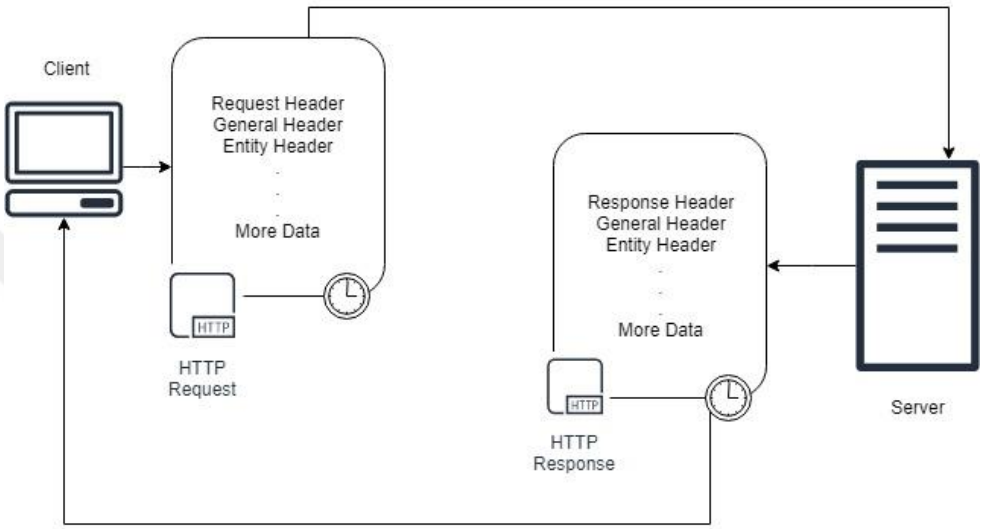

Figure 2.1 *HTTP Request and Response*

HTTP packet plays a critical role in web application attacks because it can be injected by malicious data from attackers. For example, in the GET method, the input data combines with the URL which could be easily exploited by attackers to get sensitive data from log files in cleartext. Attackers could play around the head method to deceive the server. That may, for example, reveal important information about the server such as type, version, and date. Another example of the weakness of TRACE field and User-Agent field and Referee that attackers could steal user`s cookies, hijack the session and launch XSS attack [7].

Table 2.1 *HTTP Request Fields*

Field	Description
Request Line	**Contains Method, Path, Protocol.**
Method	This field indicates the request type most common methods are: Get: retrieves server information. Head: returns header information without body information. Post: sends data to the server. Put: upload representation of URL. Delete: delete resources. OPTIONS: return the methods the server support, it could be used to covert request to TCP/IP tunnel.
Path	Contains the path of the resource and it's a part of the URL and comes directly after the host part.
Protocol	Contains the version of the HTTP protocol.
HTTP headers	**Contains Pairs of Name: Value for each line**
User-Agent	Provides browser version and the Operating System type and configure to the default value by the web browser client.
Accept-Encoding	Provides information to the server about the ability of the client browser to compressed output.
Host	Provides hostname of the website or shared IP.
Cookie	Store values related to the session between server/client.
Referrer	Provides the web page of the link to the current page in redirection or click.

Table 2.2 *HTTP Response Fields*

Field	Description
Status code	Contains 3-digit code.to give information about the operation, common status codes are : 200: for successful operations. 3xx :for redirection 4xx: for error in the client request. 5xx : for error from server-side.
Set-cookie	Could store data to help the server identify the client using random value.
Server	Provides information about the webserver system and version hosting the website.
Content-Length	Indicates the body bytes number in response.

As progressively complex web applications are being utilized today, the traditional method for building web application on a single framework is an account of the past. Multiple tier web application is the most used architecture nowadays, commonly three layers with physical separation between them. The complex structures and the integrated components of the web application architecture are associated with increased vulnerability [8]. What can be clearly seen in Figure 2.2 is the source of threats, place of security controls and weakness on multiple layer web application architecture. HTTP software could contain various kind of errors and security weakness. Moreover, a misconfiguration in the database system could

lead to malicious injection attacks. Likewise, poor coding and weak input validation in the web application are sources of many attacks [9,10]. The design of HTTP protocol when integrates with complex components is another source of security weakness [11]. Current techniques for security have demonstrated to be untrustworthy, moreover an available web application in the front end could be misused by different kinds of web application attacks [11].

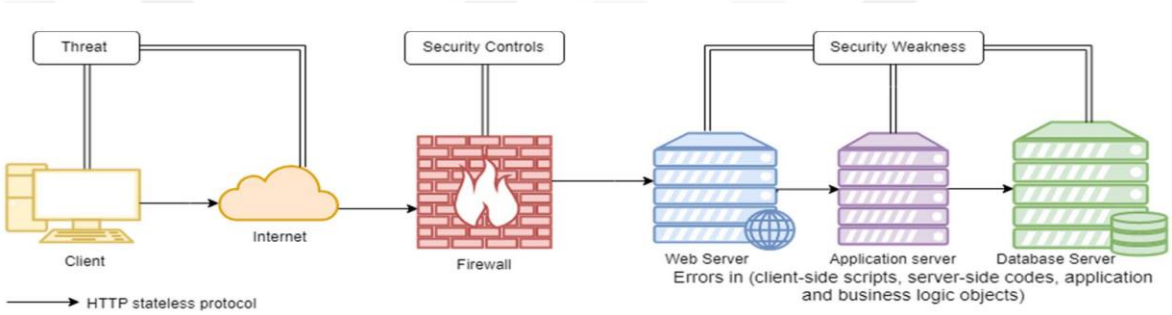

Figure 2.2 *Threats, Security controls and Security weakness in the Multi-Layer Web Application Architecture [138]*

2.2 Web Application Attacks

Web application attack can broadly be defined as an attack where the shortcoming and errors of the web application code are abused and exploited to bargain the security of the back-end system [12]. Subsequently, various taxonomies are utilized to characterize web application attacks, the most widely recognized taxonomies are OWASP Top 10 [13] and Sans Top 25 [14]. Perhaps, more scientific web attacks taxonomy will appear in the future; in this way, a precise framework will contribute and aid in the improvement of detection applications. Table 2.3 lists different types of web application attacks.

Table 2.3 *Top Web application Vulnerabilities OWASP top 10 VS Sans top 25*

OWASP top 10	**Sans top 25**
Injection	Improper Neutralization of Special Elements Used in an OS Command (‘OS Command Injection’) SQL Injection Code Injection Unrestricted Upload of File with Dangerous Type Download of Code Without Integrity Check Inclusion of Functionality from Untrusted Control Sphere
Broken Authentication	Missing Authentication for Critical Function Improper Restriction of Excessive Authentication Attempts Use of Hard-coded Credentials Reliance on Untrusted Inputs in a Security Decision Missing Authorization Incorrect Authorization
Sensitive data exposure	Missing Encryption of Sensitive Data Cleartext Transmission of Sensitive Information
XML External Entities (XXE)	-
Broken Access control	External Control of File Name or Path Improper Authorization
Security misconfiguration	Execution with Unnecessary Privileges Use of Potentially Dangerous Function Incorrect Permission Assignment for Critical Resource
Cross Site Scripting (XSS)	Improper Neutralization of Input During Web Page Generation (‘Cross-Site Scripting’)
Insecure Deserialization	Use of Externally-Controlled Format String
Using components with known vulnerabilities	Integer Overflow or Wraparound Use of a Broken or Risky Cryptographic Algorithm Use of a One-way Hash Without a Salt
Insufficient logging and monitoring	-

2.2.1 Injection attacks

Although the names and types of injection attacks are different according to various type of applications, still the basic concept of these attacks is similar. If clients make a request to private services in web applications, its necessary to fill username and password fields or other fields required by the application. So far, the inserted data is sent as a request for comparison with the stored data in the database, or to retrieve the stored data or to store new data. Foundationally, several languages exist to link the database with web applications such as the widely used SQL. Vulnerabilities in these languages may enable an attacker to exploit the weakness, by input malicious code to modify the request.

Injection attacks manipulate the database to obtain data or perform several actions that threaten confidentiality, integrity, and availability. Additionally, complex attacks may result in other systems breaching. Figure 2.3 presents a general demonstration of how injection attacks took place in the web application architecture. In command injection, the attacker could inject malicious system shell to run command not only on web application but on the target web server system. In Cross-site scripting XSS, the attacker injects script inside the vulnerable web application and store that script, thereby the client browser may be tricked to execute the malicious script on the web application with the client privileges. The injection targeting the search filters meta characters, that could lead to access of unauthorized data in LDAP.

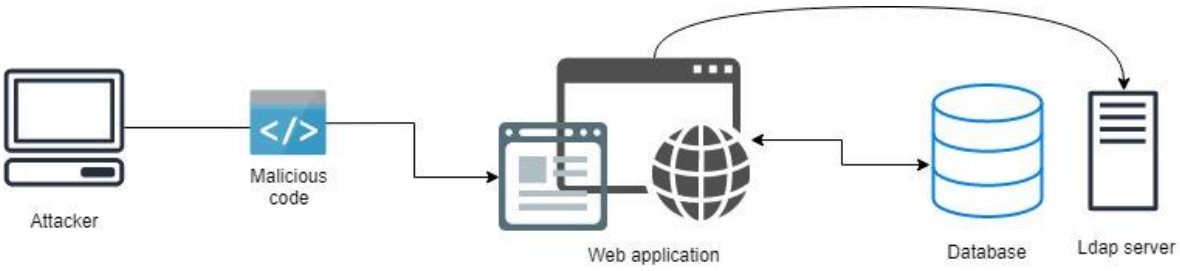

Figure 2.3 *Web application injection attacks*

2.2.2 Misconfiguration attacks

Misconfiguration attacks generating considerable interest in terms of ease and spread. Coupled with vulnerabilities scanning tools for web applications, database systems, and

network services. The misconfiguration attacks as shown in Figure 2.4, become easy to discover and to be exploited by attackers. Although the OWASP excluded some of the attacks from misconfiguration category, the reason behind most of these attacks is due to the lack of good configuration of services. Developers and administrators often use systems with default settings. Besides the deployment of outdated and not patched software and services, even though updates may generate new vulnerabilities if not configures properly. Furthermore, vulnerable components affect the whole security of web applications.

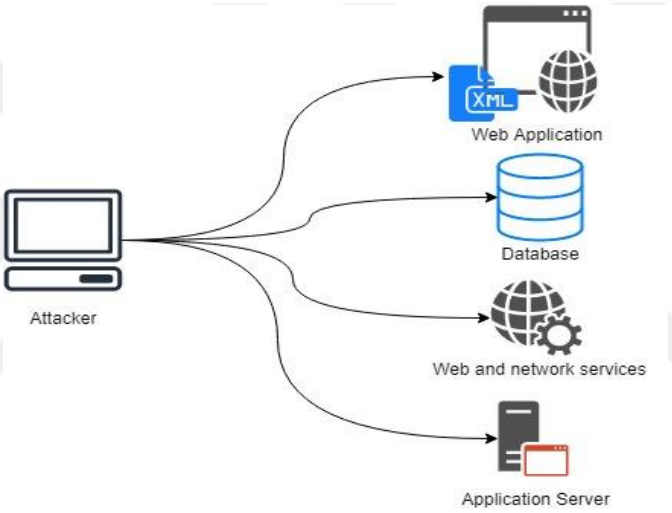

Figure 2.4 *Misconfiguration attack against web application*

In the broken authentication attack the user request access to the web application, then the authentication done by various methods such as password and account name. Further credentials will be compared with the database, leading to the granting of session-id. Now, the user can use the services until the end of the session-time, here the attacker could steal the session-id by many methods such as credential stuffing, brute force random credentials and default username and password. Likewise, misconfiguration in XML parser causes XML external entity attack.

2.3 Detection of Web Application Attacks

Numerous scientists are endeavouring on detection and countermeasure web application attacks. There are two basic detection approaches as of now being embraced in research [15]:

- Anomaly-based: Anomaly-based approached can identify unknown attacks because of the ability to learn. However, anomaly-based facing performance and accuracy problems with high false positive.
- Signature-based: Signature-based techniques depend on the predefined signature of attack, that is the main reason for its ability to get very high accuracy in detection of attacks and less false positives. Many security techniques depend on both approaches in their products.

Web application attacks detection solutions could be divided into :

- Web Application Firewall
- Application Intrusion Detection System
- Web Application Honeypots

2.3.1 Web application firewall

Web application firewall (WAF) is the most widely used technology on which companies spend a lot of money and it is effective in countering attacks. The ability to work on a number of network layers, making it very effective, especially in the application layer. It made it more flexible to control the incoming and outgoing web application traffic, as traditional firewalls failed [16]. Many scholars hold the view that WAF effective in preventing breaches and mitigate attacks [17]. However, it experiences some shortcoming. In recent years, attackers have improved their capabilities to launch attacks, and they are able to launch sophisticated attacks that could avoid and bypass WAF. As a result, much criticism exists against WAF, whether in known products or in the implementation. [18, 19]. For example, predefined rules can`t recognize some high-level application logic. Moreover, WAF suffers from high false negatives and high false positives rates, inability to detect the unknown attack, with high operational cost [20-23]. Recently, researchers applied

automation techniques, such as machine learning and data mining algorithms as in [24, 25] to overcome WAF limitations. The question of performance raised, because web applications are real-time systems dealing with high traffic. Incomprehensibly, applying methods to upgrade the performance of data mining and machine learning algorithms could result in decreasing accuracy and performance [24]. In like manner, solve the performance issue by upgrade WAF hardware equipment will lead to a high cost.

2.3.2 Application intrusion detection system

Application Intrusion Detection System (AIDS) one of the detection techniques for web application attacks. AIDS solved the traditional IDS problems and the drawbacks of the WAF [26, 27]. AIDS could exist as additionally security layer with other detection and prevention techniques. AIDS contains various type of techniques and there is ongoing research to improve it [28-30]. Traditional AIDS has much weakness. For example, sophisticated attacks could bypass it by using encrypted traffic and multi-level, Furthermore, AIDS has weak detection accuracy and the performance not high as in WAF [31-33]. Machine learning and Data mining algorithms could solve some of AIDS weakness. Many publications proved the ability of those algorithms to enhance the detection accuracy and performance of AIDS [34, 35].

2.3.3 Web application honeypot

Honeypot is an alternate methodology aiming to trap the attackers and let him inside rather than keep him outside. Honeypots could be arranged into two sorts; research and production honeypots. The objective of the production honeypot is to straightforwardly secure organizations. While the research honeypot gathers data about attacks and attackers to give indirect security. As well, a honeypot can be implemented by the degree of interaction, and type of services and resources, to high interaction honeypots and low cooperation honeypots [36]. Honeypot solves the issue of false positives, which are knowledgeable problems in WAF, and ID. Research and high interaction honeypots are good

technique to discover new and unknown attacks. Moreover, to study the motivation of the attackers and their way and methods. Table 2.4 compares between different detection techniques.

Table 2.4 *Comparison of the web application attack detection technique features*

Features / Methods	Web Application Attack Detection Techniques			
	Web Application Firewall	**Application Intrusion Detect System**	**Web Application Honeypot**	**Web Application Forensic**
Encrypted traffic inspection	yes	no	yes	yes
Types of attacks	Known web application attacks	Known and unknown network and application layer attacks	Known and unknown web application layer attacks	Known and unknown attacks
Accuracy/False Positive	Medium accuracy/ high false positive	Medium accuracy/high false positive	High accuracy/low false positive	High accuracy/too low false positive
Challenges	Easy to bypass cost, maintenance	High false alarm, encrypted traffic	Great risk if detected	Time, the massive amount of data, legal constraints

2.4 Digital and Web Application Forensic

According to a definition provided by Veena et al. [37] , Digital Forensic is “The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations”. Previous studies mostly defined digital forensic as the way toward recognizing, protecting, analysing the digital evidence and present it to satisfy legal requirements [38].

Digital forensic investigative techniques work together with forensic investigators, assisting them, and doing a lot of tasks which are difficult to done manually. This has made an accelerated and growing interest noticeable in a large number of publications available at various research sites such as ScienceDirect, IEEE Xplore and ACM Digital Library [39] Although the beginning of digital investigation a number of decades ago, it is still considered a young science, which is expected to increase growth [40]. The early beginnings of a digital forensic investigation were limited to simple evidence analysis, but now there is a vast breakthrough and advanced research to cover all aspects of digital evidence investigation [41].

Digital forensic with the widening types of applications and platforms has become branches. The overlap between the cloud forensic and network forensic and the web application forensic exist. They have the same goal and same questions to answer such as: when, how, whom, why questions and the answer probably exist in the traffic or log files [42]. Web application forensic may be defined as the branch of digital forensic which is collected and analysis events, in order to trace back the source of security attacks or other incidents on a web application [43]. Web application forensics detect web application attacks after the incidence, in order to find the evidence to explain the occurrence, causes, and motives of the incident. Traditional web application forensic rely heavily on the expertise and skills of the forensic investigator, also the increasing number of attacks and massive data made evidence analysis hard task even with the help of traditional forensic tools.

One of the main challenges in network forensics and web application forensic is the huge amount of data generated by the network. Investigation and analysis of logs generated by intrusion detection system, web application firewall, network services, and applications are hard and tedious task [44]. Jain and Kalbande [45] proposed digital forensic framework based on research finding of 25 Digital forensic frameworks, which improved existence frameworks by adding two new modules, not taking into account the challenges and specific problems of web application forensics. Likewise, Ussath et al [46] introduced a concept for a security investigation framework. It`s more appropriate for malware analysis for multistage attacks related to it.

There are several network forensic investigation frameworks, suitable to investigate a different type of protocols and networks as general not specified for web application attacks

[47]. Even more, Lazzez and Slimani [48] conducted a good comprehensive survey about web application forensics, presented definitions and did a comparison of tools. According to our best of knowledge from the literature review, there is a lack of comprehensive framework of web application forensics based on real data from attacks. That confirmed by Wazzan and Awadh [49], who suggested in their paper recommendations for building a comprehensive framework of detecting web attack for the web application layer. They considered more vulnerabilities to detect all kind of attacks. The authors give recommendations for building framework to ensure detecting most of the potential web attacks those recommendations are:

- The framework should able to monitor, responding and filtering the traffic.
- The framework should construct records that help in analyzing the payload and headers of the request and response with signature-based detection for the HTTP traffic.
- The framework must report and decrease of false-positive and increase of detection rate.

2.5 Digital evidence

Casey [50] defines digital evidence as "any data stored or transmitted using a computer that support or refute a theory of how an offense occurred or that address critical elements of the offense such as intent or alibi.". As can be seen from Casey`s definition the goal is to reconstruct the past events by investigating digital data. According to a definition provided by the national institute of justice “digital evidence tends to be used to refer to information and data of value to an investigation that is stored on, received, or transmitted by an electronic device” [51]. Digital evidence can be found in many sorts of digital records such as e-mails, access control and event logs, security devices logs, security software logs, Internet activity logs, metadata. There are many challenges in digital evidence. For example, the layer of abstraction [52], Secondly the irrelevant evidence that can be mixed with relevant evidence, Third, digital evidence is volatile which could lead to error, uncertainty, loss and can be manipulated [53].

There is a need to improving digital evidence capabilities, utilization, acquiring, analysis, in order to find digital evidence more effectively and reducing time. Backlogs requires to be analysed with facilitating of chain of custody and authentication of digital evidence. [54]. Because digital evidence is legally used in courts, it must also consider certain legal qualities such as the federal law of evidence [55]. The evidence must have the value of proof or rejection, and its presence plays a role in the probability of the facts. There must be an expert witness who can test the evidence and justify the process of digital forensic. Evidence should be self-authenticating for example by verification of digital evidence integrity through hashing [56].

In web application forensics, log files contain important information about incidents and security violations [57]. For that reason, the log file is the main source of evidence in web application forensics [58, 59]. Figure 2.5 shows the different places of Log files in the web application structure. Application logs provide information related to business logic [60]. Web server logs reveal information related to HTTP traffic. Likewise, the access log contains user activities data [61]. Furthermore, Network traffic and attacks could be found in the web application firewall and application-level intrusion detection system logs [62].

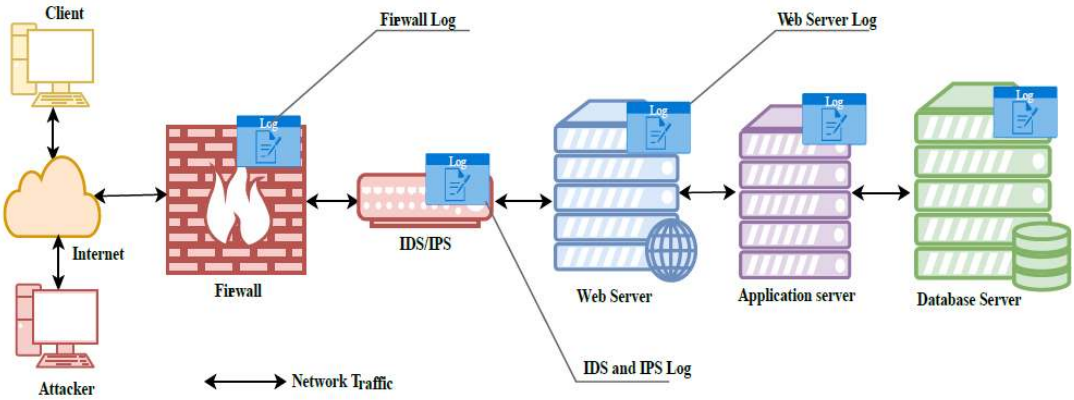

Figure 2.5 Log files in the modern web application structure

2.6 Machine Learning and Data mining in Forensic

Recently, a number of intelligent and unconventional technologies have been introduced in the detection of web attacks, as well as in digital forensic investigation. These technologies are sophisticated and offer better results than normal techniques. There is a significant overlap between the concept of data mining and machine learning, for that reason both will be explained in the next lines.

Data mining refers to scientific methods to extract interesting knowledge from massive data to assist in decision making [63]. Traditional data analysis techniques face problems with large-scale data [65]. On another hand, data mining use statistics, machine learning, database, and artificial intelligence to deal with massive data [66]. The main goal of data mining techniques is to create a descriptive model or a predictive model [67]. The descriptive model often uses statistic techniques to get general properties of datasets [68]. While predictive model usually builds models for prediction or analysis purposes. Predictive data mining has many models and techniques such as classification, association rules, and regression analysis [64]. Data mining is effective in different type of applications.

Machine Learning can be defined as an approach and algorithms using statistic and mathematical models for solving tasks by learning the problems in an automated manner [69, 70]. There are two main approaches to learning in machine learning supervised learning and unsupervised learning. Supervised learning is more suitable in problems with prior knowledge such as classification, support vector machines and neural networks problems. Unsupervised learning includes could be applied in the absence of prior knowledge, for example, clustering, normalization and dimensionality reduction problems [71].

2.6.1 Feature selection

Feature selection aims to identify the relevant features in order to build more effective predictive models. Features could be noisy or redundant that could lead to over-fitting problem and decrease the performance of algorithms [72]. Feature selection could enhance the performance and accuracy of the model. Thus, reduce the complexity of the process [73,

74]. Approaches to obtain an optimal subset of relevant features play an important role in the pre-processing phase in data mining models.

There are 3 main feature selection approaches: filters, wrappers and embedded methods as shown in Figure 2.6 the filter method is the fasted one because it contains statistical measures to evaluate features independently from the prediction model. Therefore, the filter method has a less over-fitting problem and low computational cost [75]. The wrapper method selected the feature subset by evaluating combinations of subsets with respect to the accuracy of the prediction model [76]. As a result, the wrapper method suffers from high computational cost and over-fitting problems, however, it could lead to high accuracy. The embedded method learns the optimal feature subset in creating time. Like the wrapper, the embedded method has over-fitting problems.

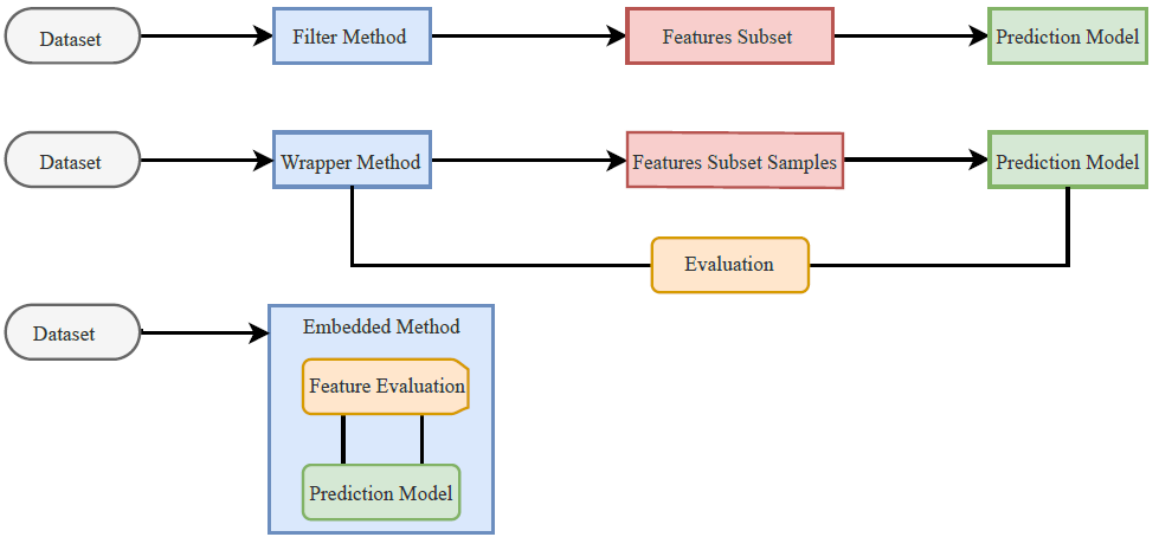

Figure 2.6 *Feature Selection Approaches*

2.6.2 Data mining and machine learning in web application forensic

In order to solve the forensic challenges for massive data, like in networks attacks and web application attacks, researchers resorted to data mining for digital forensic analysis as a branch of computer science focused on pattern extraction from large-scale data, which has

been used to support investigators when they try to solve crimes [77]. That is consistent with digital forensic definition provide by Adebayo et .al [78], who stated Digital forensic is a branch of forensic science deals with careful extraction or mining of digital evidence that has probative value within a predefined scope in such a way it can be admissible in the court of law without doubt or question being raised about its integrity. The goal of forensic investigation after maintenance of digital evidence is a decision-making process and better guidance at each decision lead to more effective investigations and solid reports [79-82] and that could be a major role of data mining where the models and tools can be developed to help investigators, digital forensics professionals, and law enforcement officers.

A growing body of literature has investigated using data mining against some of the web attacks, Bayesian algorithms provided highly accurate predictions on web attacks as in study conducted by Joshi and Geetha [83] on detection of SQL injection attacks. In the same vein, a study by Komiya et .al [84] in detection of malicious web code, the Bayesian classifier obtains 97.6% and SVM obtains 99.16 %. Moreover, clustering algorithms are one of the most useful techniques for crime data analysis [85]. Nampoothiri and Madhavu [86] applied k-means clustering on e-mail malicious. Likewise, Elbasiony et .al [87] employed the k-means clustering algorithm in unsupervised anomaly detection method. Furthermore, Sharma et .al [88] proposed a model used KD algorithm, which is an enhanced version of traditional K-means algorithm.

Furthermore, decision tree classification techniques applied in many studies Hanmanthu et .al [89] proposed model utilized the decision tree classification model to prevent the SQL injection attacks. Additionally, Nadiammai and Hemalatha [90] implemented Efficient Data Adapted Decision Tree on KDD dataset, their proposed EDADT algorithm reduces the actual size of the dataset and helped the administrator to analyse the ongoing attacks efficiently with less false alarm rate respectively. Other data mining techniques such as frequent item-set mining, mining association rules, MapReduce framework, and phylogenetic trees, Data warehousing also investigated in literature [91-95].

Previous studies have explored the anomaly detection of web application attacks, Kruegel et al. [96] conducted analysis on log files to detect web attacks and they came out with main features that help in the detection process namely: the length of a query attribute, character distribution, structural inference parameter, attribute presence or absence, attribute

order, access frequency, inter-request time delay, and invocation order. Based on Markov theory and statistics they built detection models. However, their model has a lack of stability with a high number of alerts which required manual efforts. Robertson et al. [97] deployed heuristics-based technique with an automation method to convert malicious requests to anomaly signatures. However, many similar alerts need manual efforts to handle, which cause detection problems. their study was helpful in extracting good web application attacks features.

Nguyen et al. [98] performed several experiments in web traffic and provided the CSIC dataset that contains 30 features. In their analysis, it was shown that if there were many features linearly correlated to each other, then the correlation-based feature selection (CFS) measure could achieve better results. The CFS success in removing more than 63% of redundant features. Unfortunately, the accuracy of the detection was reduced. Atienza et al. [99] mentioned the problem associated with CSIC dataset which relies on some features with a single value that not helping in the differentiation of attack and normal requests. As a result, Atienza et al. removed single value features from the dataset, consequently, his models didn`t achieve good detection result. In another study [100], researchers conducted a similar experiment with 11 relevant features. The features were extracted by applying the n-grams model which succeed in reducing irrelevant features by 95%. However, the accuracy didn`t increase but was reduced by 6%, that led to unreliable results. In order to solve the problem, they adopted an n-gram model combined with expert knowledge for feature construction [101]. The combination of features extracted first by n-grams and expert knowledge and then used in feature selection performed well when the number of features was important. While merging the expert knowledge features with the selected features from the n-grams more suitable for accuracy. Zhang et al. [102] applied high-order n-gram with Bayesian probability 5-gram works well in HTTP payload when combined with Bayesian probability.

The experiments result showed noticeable decreasing of a false positive and false negative. Therefore, this finding supports those of previous studies. Moreover, Wressnegger et al. [103] conducted experiments by applying high-order n-grams of bytes with a combination of learning schemes. Their methods perform well in the detection of web attacks with a successive identification of 81.5% attacks and 0.01% false alarms. Together, these results provide an important insight into using n-grams for the detection of web attacks, and

this view was supported by Nascimento and Correia [104] found that n-gram modelling is very accurate with a low false-positive rate in the detection of web attacks.

2.7 Summary

In this chapter, the web infrastructure, protocols and their way of communication are discussed. Additionally, the security problems in these technologies are clarified. Those security weaknesses caused different types of attacks. Web application attacks vary in their forms and methods of implementation, attacks can be carried out through the server side or through client side. Indeed, the methods of attacks are constantly evolving. As well as techniques used in countermeasure them.

The most effective and used techniques were reviewed, especially, the advantages and disadvantages. The high number of attack incidents showed that the aforementioned techniques could be evaded and overridden. Their effectiveness is not absolute. Therefore, a digital forensic investigation has been found. Digital forensics and its various models still use manual and traditional methods. In brief, intelligent and automated techniques such as machine learning and data mining are described. Finally, the literature review was conducted for previous studies that used data mining and machine learning algorithm.

3 METHODOLOGY

This research focus on web application attack which is the contemporary phenomenon. The experimental method is considered suitable for this type of phenomena. Two research methods for data collection were adopted to represent two plans; Plan A and alternative Plan B as given in the following sections.

3.1 Data Collection (Plan A)

This approach designed by considering best practices from CISCO in networking and best practices from Barracuda in the web application firewall. The methodology follows information security standards, guidelines, concepts of ISO/IEC 27001:2013 to apply defence-in-depth and multi-layered security, in addition to the best practice of Scientific Working Group on Digital Evidence for evidence collection from logs and database. Specifically, to get results compatible with the digital investigation process.

As general description as in Figure 3.1, the attacks will come from the Internet, which considered as untrusted networks, then face the first security layer the packet-filtering, which will route the traffic to the experiment network. The second security layer is the network firewall, it will protect against many kinds of network attacks. The firewall could give control of traffic by block all other traffic and allow only the web traffic throw ports 80 for normal traffic and 443 for encrypted traffic. The web application firewalls play traffic inspection role. When the attacker launches the attack against vulnerable web application all the traffic will be captured. Using the middle proxy is important in order to connect the web interface with virtual machines. The components are:

- **Routers and Firewalls:** Using a router as packet filtering with network firewall constitute a defence in depth. It will protect against unwanted network attacks and allow only traffic of HTTP/HTTPS by forwarding ports 80/443.
- **Web Application Firewalls:** WAF performs deep packet inspection for HTTP/HTTPS, which can't be performed by network firewalls. Outbound traffic represents a threat, utilizing web application firewall and setting rules to block HTTP outgoing traffic will mitigate the risk of advanced. Thus,

ensure the minimum privileges best practice. WAF could customizing attacks and identify and block the unwanted attack. WAF could prevent information loss and monitor attack. WAF logs contain the authentication context and all GET and POST data.

- **Log files and Integrity Check:** Collecting log files of all firewalls and proxy is the most important step. Since forensic investigators use log files to gather information about the incident. Log files are the main source of digital evidence; indeed, the integrity of the data guarantee the chain of custody and accuracy of the digital evidence. In order to verify the log files, checksum will be used.
- **Fake Websites and MITM Proxy:** Man in the middle proxy or also known as Honey proxy, will help in inspection of encrypted traffic. Moreover, MITIM proxy has the feature of saving HTTP conversations.
- **Virtual Machine:** The snapshots provide the ability to restore the virtual machine to early-stage if the system compromised, thus making it a rich source of maintained evidence for the digital investigation. Importantly, a snapshot for each successful attack will be saved in a secure database to seizure the evidence.
- **Data-warehouse:** The secure database and the logs files will transmit to the data warehouse and mounting frequently. The data-warehouse will be isolated, offline and it will ensure the data integration. Many processes of data mining such as data cleaning, pre-processing could be within the data warehouse.

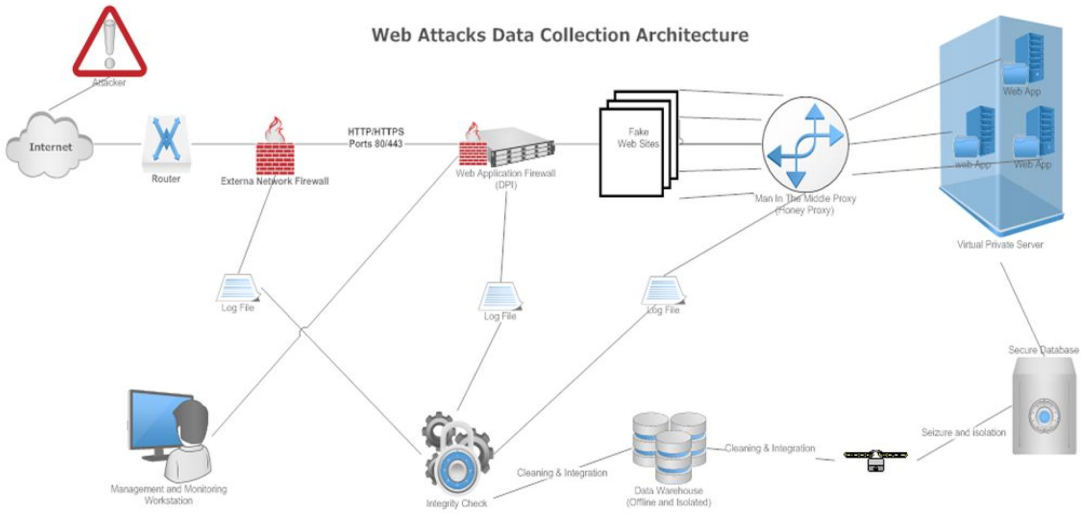

Figure 3.1 *Adopted Methodology for Web Application Attacks collection*

3.2 Data Collection (Plan B)

One of the widest used datasets is DARPA dataset which contains real network traffic. The traffic generated from a system plays the attackers role. KDD CUP'99 dataset processed the 1998 DARPA dataset [105, 106], but the main focus was intrusion detection traffic with few web attacks traffics. Unfortunately, it became outdated dataset ECML/PKDD-2007 [107] datasets and CSIC-2010 [108] both of them contains web application attacks and used in much-related literature. Additionally, AMENSIA dataset [109] contains automated SQL injection web attacks, As KDD and DARPA datasets are outdated and lack of other types of web application attacks. As a result, Hans et al [110] and Canli et al [111] collected sound good datasets, however, those datasets are not available publicly. Moving on now to consider the experimental design and setup, which contain the chosen datasets in our research.

3.3 Experimental Design and Setup

The web application structure has two types of logs, network-level traffic logs, and web application logs. The experimental design simulates the process of digital evidence collection and investigation, by using multiple datasets contain different types of traffic logs. Multiple datasets provide the ability to measure the models and methods of stability. For the web traffic, CSIC 2010 as the single-class dataset was chosen [108], while ECM/PKDD`2007 were selected to represent multiclass web traffic dataset [107]. The two datasets contained labeled HTTP requests and various types of attacks, such as cross-site scripting, LDAP injection, SQL injection, buffer overflow, information gathering, files disclosure, CRLF injection, XPATH injection, SSI attacks, path traversal, and command execution., CICID2017 [112] was selected as network traffic dataset. The CICID2017 contains recent attacks from the intrusion detection system (IDS) traffic and intrusion prevention system (IPS). Those attacks are labeled based on the timestamps, source IP, destination IP, and source and destination ports and protocols. Indeed, the 3 datasets as shown in Table 3.1 can be more useful for identifying the optimal feature selection methods if they are well pre-processed.

Table 3.1 *The CSIC, ECM, CICID attacks datasets*

Dataset	Data	Class	Features	Attacks type	Size (instances)
ECM/PKDD`2007	Web traffic	Multiple class	14	SQL injection XSS, OS commanding, LDAP Injection, PathInjection attacks, SSI, Pathtraverser attacks	68269
CSIC 2010	HTTP requests	Single class	12	SQL injection, buffer overflow, information gathering, files disclosure. CRLF injection, XSS, server-side include, parameter tampering attacks	223,585
CICID2017	Network traffic	Multiple class	84	Brute force attack, XSS, SQL Injection attacks	170366

The experiments were run using Weka experimenter version to evaluate feature selection methods and to compare between different classification models. Weka is an open-source software contains a collection of data mining and machine learning algorithms for performing tasks related to discover and extract knowledge from data.

The experiment was performed on an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz, 2801 MHz, 4 Core(s), 8 Logical Processor(s) and 16 GB of RAM with 64-bit Windows 10 Pro Operating System. Significance levels were set at the 0.05, the experiments were carried out with cross-validation 10 folds. In order to identify the best classification algorithm and feature selection methods.

The trees algorithms were used the first one J48 algorithm or C4.5 algorithm was used to create a decision tree based on the concept of information entropy the second one, The random forests algorithm which is referred to a statistical machine learning method constructing a multitude of decision trees each tree casts a unit vote for the most popular class at each input. To compare the trees algorithms Naïve Bayes classification was selected, can be defined as an algorithm based on applying simplified Bayesian analysis, given a dataset instance to be classified is assumed to belong to exactly one class Naïve Bayes classifier computes a posterior value to attach a class label to the given test instance.

Wrapper feature selection methods were applied using 2 approaches a forward search strategy with Naïve Bayes, and a backward strategy with the decision tree. Following that, the best filter feature selection method was chosen to be combined with the wrapper methods. The basis for comparing the feature selection methods was based on 2 machine learning algorithms, the C4.5 decision tree algorithm was selected for its reliability and validity on web attacks. It outperforms other tree-based machine learning algorithms namely Random Forest, CART, Random Tree on CSIC 2010 as in [98].

Naïve Bayes algorithm was selected to represent a different approach in machine learning; as it implements the Bayes theorem for the classification problems. Naive Bayes has the best performance in CICIDS2017 comparing to other 6 machine learning algorithms as specified in [113]. The same two machine learning models have been used for comparison of feature selection methods as in [114]. One advantage of the C4.5 and Naïve Bayes is that both could work in a binary classification problem and multi-class problem. The experiments were run using cross-validation.

3.4 Experiment Measurement

3.4.1 Cross-validation

There are several techniques available for evaluation prediction of the data mining and machine learning models. The proper technique should maximize the accuracy of the model and minimize the training time and the execution time. One of the most common approaches is cross-validation. Applying cross-validation to the training set will result in splitting the dataset into training and validation subsets [115]. Cross-validation divides the dataset into a number of folds train/test subset. Aiming to get the equal size of the test set and training set which produces a number of evaluation results. then the average performance of the k-folds produced. Cross-validation could estimate the prediction accuracy and training and prediction errors. For large-size datasets, 10-fold cross-validation is broadly used because it is computational friendly and outperformed other estimators [116].

3.4.2 Accuracy measurement

The model accuracy measure calculated by the percentage of correct prediction to all prediction of attacks by a given model. Accuracy measure depends on the confusion matrix with help precision, recall and following parameters:

TP: true positive of prediction for positive values that are actually positive.

FP: false positive of prediction for positive values that are actually negative.

TN: true negative of prediction for negative values that are actually positive.

FN: false negative of prediction for negative values that are actually negative.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3-1}$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \tag{3-2}$$

$$Precision\ (Specifity) = \frac{TP}{TP + FP} \tag{3-3}$$

3.4.3 Error measurement

One of the most well-known error measurements for prediction model is the root-mean-squared error (RMSE) and the mean absolute error (MAE). The RMSE measurement is more practical because it could discover large errors. Root Mean Square Error (RMSE) is the standard deviation of the prediction errors, it calculates the square root of value and figures the differences between predicted values and actual values as shown in Equation (3-4) where n is the sample size.

$$Root\ Mean\ Square\ Error\ (RMSE) = \sqrt{\frac{\sum_{i=1}^{n}(predicted\ value - actual\ value)^2}{n}} \quad (3\text{-}4)$$

3.4.4 Feature selection and basis

In order to validate the experiment results, the most well-known algorithms and feature selection method that applied before in literature in web application attacks analysis were chosen. The following filter feature selection methods were selected as a basis of comparison:

- Information gain: this feature selection measures the information gain of the attribute with respect to the class. Where H(Class) is Shannon's Entropy for the class.

$$Infogain\ (Class, Attribute) = H(class) - H(Class|Attribute) \quad (3\text{-}5)$$

- Chi-squared: computing the value of the chi-squared statistic with respect to the class. Where O=the observed frequencies, E=the expected frequencies.

$$\chi 2 = \sum \frac{(O-E)^2}{E} \quad (3\text{-}6)$$

- Correlation-based feature selection (CFS): this method considers the prediction and redundancy between the features. Next, the highly correlated features with class and less between each other were selected.

For basis; C4.5 decision tree algorithm and Naïve Bayes were selected in hybrid feature selection approach. The two selected algorithms were particularly useful in web application attacks and their reliability and validity on detection of web application attacks confirmed in previous studies as mentioned on background Chapter 2.

The methods for selecting the algorithms were as follows:

- literature review about machine learning algorithms that proven to have high accuracy and high performance on the same problem which is web attack forensics and on the same three datasets was conducted.
- Since the selected datasets with binary class problem and the other with multiclass problems, the algorithms which could solve the both problems together were chosen. Not all machine learning algorithms work on both types.
- Initial experiments were conducted, in order to see the algorithms which has less model building time. The wrapper method in in the hybrid approach depends on the performance of the algorithm. Therefore, the time for building the C4.5 was calculated and resulted in 0.97 seconds while reached 15.08 seconds for the CART model and 7.06 seconds for the Random Forest model for the CSIC2010 dataset. For the ECML/PKDD 2007 dataset; the time taken for building the C4.5 model was 0.3 seconds, Random forest 7.67 seconds and 927.49 seconds for the CART model.
- Moreover, the C4.5 and Naïve Bayes have been used in similar experiments to compare feature selection method as in [114].
- Furthermore, the two algorithms represent a different approach in data mining; Naïve Bayes implements Bayes theorem for the classification problems. Naïve Bayes has outperformed six data mining algorithms in term of performance in CICIDS2017 dataset [113].

In literature review chapter many algorithms applied in web application attacks were discussed, like Naïve Bayes, Random forest and Bayes net. However, to experiment further

data mining algorithms to be applied in the proposed framework following algorithms were selected:

- SMO Algorithm was used before in web application attack detection, particularly in injection attacks as in [130] and for DDOS attack as in [131].
- the PART algorithm which builds a decision tree in iteration in order to covert best tree leaf to at rule [132].
- The Jrip algorithm is a rule classification which repeats the incremental pruning to reduce error in classification algorithm [133], also used in web application attacks and resulted in good performance as in [134].
- OneR is rule-based algorithm which produces a decision tree with 1 level its simple algorithm and applied in many previous studies in detection of web application and network attacks [135, 136].
- Adaboost a boosting algorithm, it boosts nominal class classifiers by select subset of a weak classifier to generate strong classifier, utilized before in detecting of web-based attacks as in [137].

3.5 Summary

This chapter has described the methodology used in the research. Starting with data collection, a novel way to collect real attacks contracted. Besides a review of most available web application attacks datasets was conducted and three datasets were chosen. Later experimental environment was set up to simulate real cases scenarios. Where web application attacks investigation involves various log files. The measurements used to test the effectiveness of experiments were mentioned as those used to measure accuracy and error rate. Next The selected data mining algorithms to validate the model were provided. In the chapter that follows, the proposed framework for web application attacks forensics will be presented.

4 THE PROPOSED FRAMEWORK FOR WEB APPLICATION ATTACKS FORENSICS

As was mentioned in the first chapters, there is increasing concern on applying data mining and machine learning techniques in web application attacks forensic. The structure and functions of the proposed framework will be explained in the following sections. The proposed framework combines the basic process used in digital forensic investigation with the data mining process. Extensive literature search has shown that there are many digital forensic models. More than twenty Digital forensic models exist [117]. The maximum number of the processes exists in the digital forensic frameworks is 13 processes while the minimum number of processes is 3 process [118] namely preparation, investigation and presentation. Those processes were considered as essential processes in digital forensic.

The framework designed to ensure the preservation of digital evidence and data analysis with the highest accuracy and reliability. The framework designed to handle digital evidence in large datasets. Together, by integrated data mining phases with essential digital forensic phases. Figure 4.1 shows the basic structure of the proposed framework.

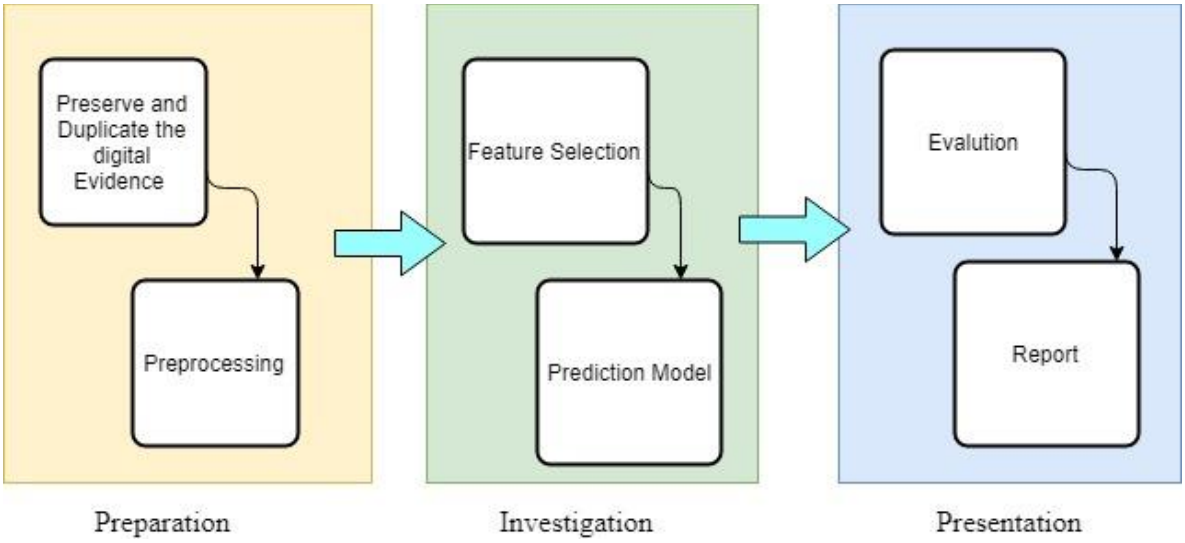

Figure 4.1 *The Proposed Framework for web application attacks forensics*

4.1 Preparation Phase

This phase has several different names in other frameworks. However, it is the most important phase, as it contains the preservation of digital evidence. In order to complete this phase properly, it requires that the log service be activated for all types of hardware and software and configured properly.

4.1.1 Preserve and duplicate

The malicious users use many techniques to manipulate log files, therefore backup is required. The log files are collected from the web server, application server and related security systems as shown in Figure 4.2. The secure database ensures the copies of log files saved and their hashes/checksum are calculated. Additionally, the secure database plays the digital evidence container role, the log files should not be manipulated or tampered. Chain of custody should be applied in any transaction or transmission of data which the legal requirements are met.

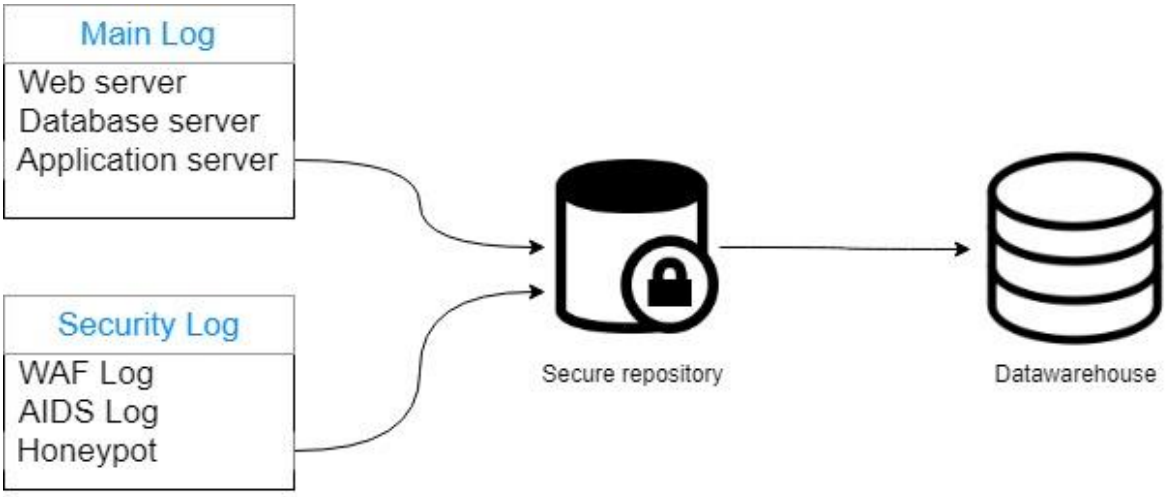

Figure 4.2 *Log file Duplication and Preserving*

Log files have large size and diversity because they came from different sources and collected over time. As a result, there is a need for a technology that provide integration, non-

volatile and time variant features. Data warehouse facilitates the process of storing while ensuring that they are stored for a period of time and do not change that ensures all the previous requirements are met. Moreover, the framework is flexible, more modern techniques such as blockchain can be adopted.

The key process of preserve and duplicate can be listed as follows:

- Collect Log files.
- Backup and Secure Log files.
- Calculate hashes/checksum of log files.
- Transfer Log files to Data warehouse.
- Document every action to maintain the chain of custody.

4.1.2 Preprocessing

The pre-processing and feature selection combined in a novel hybrid approach as shown in Figure 4.3. The pre-processing in data mining also known as the preparation, and it's the most important step for data mining, because the pre-processing addresses the problems with dirty data such as incomplete, noisy and inconsistent data by transformation, integration and cleaning data. Most of the web traffic and log contain the same structure as explained in chapter 2 web application architecture section, therefore algorithm1 was proposed for data pre-processing and to construct new features. The algorithm extracts the static characteristics of the string attributes, such as the length of attributes, and the total number of digits, letters, and special characters. Web attack datasets contain many string attributes that are problematic for data-mining algorithms, therefore as a set of numeric features representing character occurrences was constructed from string attributes. Multi-interval discretization [129] was adopted to pre-process the numerical features to partition the numerical features into subranges.

The key process of preprocessing can be listed as follows:

- Log files integration.
- Log files cleaning
- Features construction as in Algorithm1.

- Discretization features.

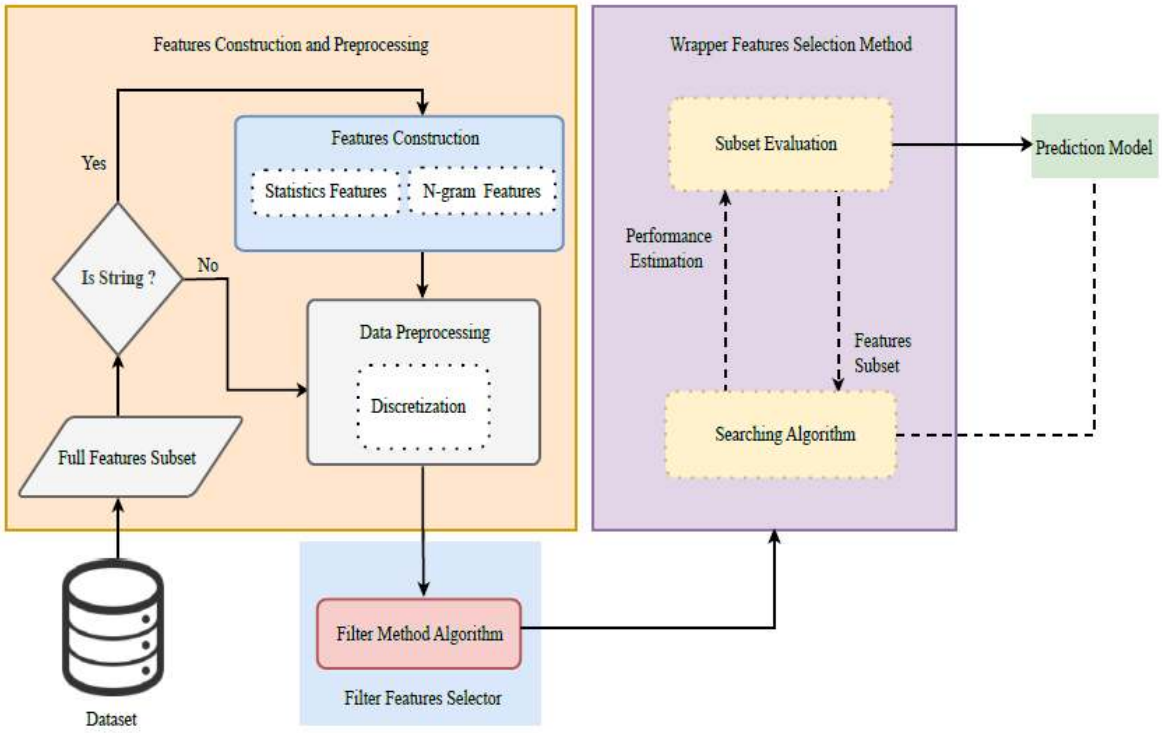

Figure 4.3 *The hybrid approach for feature selection*

Input: The string features F= $\langle f_1, f_2, \ldots, f_n \rangle$
Output: New constructed features FDigits[], FLetters[], FChar[],FLength[], 1-gram sequence FUnigram[]

$i \leftarrow 1$
while $i \leq n$ **do**
  Length, number of digits, number of letters, number of special characters calculated and constructed as new numeric features.
  $FDigits[i] \leftarrow$CountDigit(fi);
  $FLetters[i] \leftarrow$CountLetters(fi);
  $FChar[i] \leftarrow$CountSpecialChar(fi);
  $FLength[i] \leftarrow$CountLength(fi);
  Splits a string feature into unigram with max 1 gram.
  FiUnigram = CharacterNGramTokenizer(Fi,min=1,max=1)
  $i \leftarrow i + 1$
end
return *FDigits[], FLetters[], FChar[],FLength[], FUnigram[]*

Algorithm 1: *Feature construction*

4.2 Investigation

The investigation phase is the stage at which digital evidence is handled and analysed. After creating new features in the previous stage. The optimal features will be selected, then the mining model will be used to search for attack information and extract the interesting results. The prediction model indicates the attack and accordingly indicates the digital evidence.

4.2.1 Feature selection

The proposed hybrid approach integrated filter and wrapper methods, to select the optimal features. Figure 4.3 shows the hybrid feature selection approach which combined the filter and wrapper methods. As shown in Figure 4.3.; the feature selector uses selection criteria to evaluate each feature based on information measures, distance measures, and dependence measures. The filtered feature subset was evaluated using the wrapper method. The evaluation was based on prediction models, Bayesian classifiers and the decision tree. The prediction model could be applied with the forwarding searching strategy, where features were added in each iteration until adding a new feature did not enhance the performance of the model. Applying the prediction model with a backward strategy started with all of the subset features, and then eliminated the feature that had less effect on the performance of the model in each iteration. The searching strategy was chosen according to computation expensive in the evaluation of subsets. Finally, the feature that performed the best was chosen as one of the optimal feature's subsets.

4.2.2 Prediction model

Recent developments in the field of digital forensic have led to an interest in a large number of data mining algorithms. Data mining algorithms which are suitable for web application attack have been tested and implemented recently. However, to select an appropriate algorithm for digital forensic investigation, a number of criteria must be

considered to ensures the digital evidence requirements. Moreover, those algorithms should be capable to provide the highest accuracy and reliability in digital evidence analysis. To this reason, algorithms that have proven effective in detecting web attacks tested in the experiment section. The algorithms that get the best results can be applied and used as a prediction model in the proposed framework.

4.2.3 Presentation

At this phase the result of the investigation and evaluation presented. For data mining visualization techniques could be used, however visualization sometimes hard to grasp and still not formally recognized on law and policies. On the other, a written report will be more effective and easier to understand by law enforcement, lawyers, digital forensic investigators. The presentation should be clear and explain the phases in details and maintain chain of custody.

Taken together, this framework was prepared by adapting and mapping the most important process in traditional digital forensic to basic data mining and knowledge discovery process. Figure 4.4 shows the process model of the proposed framework. The model was divided into three main phases namely preparation, investigation and presentation. In order to make a forensic image of log files, the log files were collected. To ensure the integrity and security of the log files, the database was configured to calculate the checksum and enforce access control policy. In an attempt to prepare the digital evidence for the investigation phase, the log files were transferred to Datawarehouse. In the Datawarehouse the different log files integrated and stored. To construct new features for web application attacks, a feature construction algorithm was used. In order to select optimal features, a hybrid feature selection approach was performed. These features were then used for the data mining model to which the web attacks had been detected. The results were evaluated using recall and accuracy. Finally, forensic report and chain of custody document were generated.

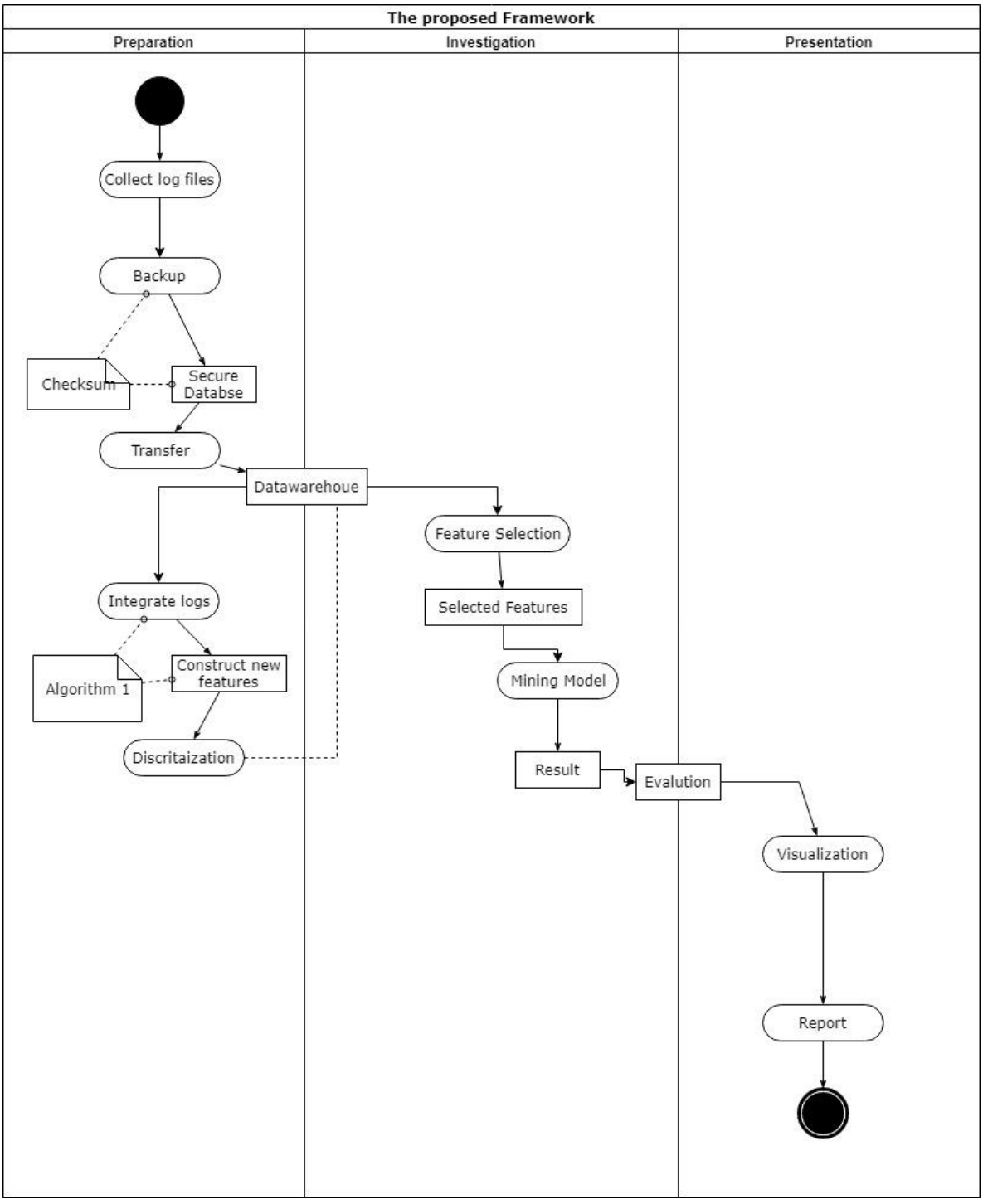

Figure 4.4 *The process model for the proposed framework*

4.3 Summary

Thus far, the proposed framework was designed according to the essential phases of the digital forensic investigation. These those essential phases were integrated with the basic processes of data mining and extracting knowledge from large amount of data. The phases and activities on each were described. In particular, data mining algorithms need different preparations for the digital evidence. The adopted pre-processing algorithm were presented, which is part of a proposed hybrid feature selection approach. The next chapter describes the experiments and results of algorithms, beside discussion.

5 EXPERIMENTS AND RESULTS

This chapter presents the experiments conducted as prepared in the previous chapter. The experiments were conducted on three databases in order to ensure the reliability of the results. The first sets of results on each dataset focus on the experiments of the proposed hybrid features selection approach. The comparison of the hybrid approach result is based on 3 feature selection methods and the original features as a baseline. The second sets of results present the findings of experiments on the proposed framework where nine data mining algorithms used to validate the framework. Finally, these findings will be discussed in detail in the discussion section.

5.1 CICID2017 Dataset Experiments

On observation of the CICID2017 dataset, most of the original features are numeric, only 4 features are nominal namely Flow ID, Source IP, Destination IP and Timestamp the discretization conducted according to the experiment setup. Moreover, the timestamp feature removed. As a result, the features number decreased by one. Table 5.1 shows the number of instances and features before and after pre-processing.

Table 5.1 *The Instances and Features of CICID2017*

Dataset	**Original**		**Preprocessed**	
CICID2017	Instances	Features	Instances	Features
	170366	84	170366	83

5.1.1 The feature selection results

The experiments on the CICID2017 dataset as shown in Figure 5.1 shows the effectiveness and accuracy of the CFS and information gain methods in term of features reduction. CFS reduced features to 3 features with 99.5401%. average accuracy. The proposed hybrid feature selection with naive Bayes (HFSN) achieve accuracy of 99.5759% and selected only 2 features which is better than the CFS method. On the contrary, the

proposed hybrid feature selection with decision tree (HFST) approach did not achieve a significant enhancement in accuracy compared to the CFS filter method.

The recall reaches the best value in the HFSN method. There are no significant differences in recall value in the proposed hybrid approach. The root means squared error (RMSE) declined in all feature selection methods, but the lowest RMSE was in the HFSN approach. Notably, HFSN and HFST approach decreased the CPU time.

5.1.2 Classification model results

The framework applied with various type of data mining algorithms as a prediction model. It can be seen from the data in Table 5.2 the Naïve Bayes algorithm reported a significantly different in model building time with good accuracy and RMSE. Moreover, a sharp increase in accuracy and a slight decrease of RMSE error were detected in J48. Interestingly, most of the algorithms have the same Precision, Recall, and F-measure great value. Conversely, OneR, Adaboost have lower recall comparing to other algorithms. SMO algorithm has the lowest performance in term of building time.

Table 5.2 *Comparison of classification models on CICID2017 Dataset*

Classification Algorithm	Accuracy %	RMSE	Building time (Seconds)	Precision	Recall	F-measure
Bayes Net	99.578	0.0392	0.08	0.999	1.0	1.0
Naïve Bayes	99.5733	0.0393	0.05	0.999	1.0	1.0
J48	**99.5786**	**0.0384**	0.2	0.999	1.0	1.0
Random Forest	99.578	0.0384	6.16	0.999	1.0	1.0
SMO	99.5774	0.3124	39.59	0.999	1.0	1.0
PART	99.5697	0.0389	0.22	0.999	1.0	1.0
JRip	99.5263	0.0422	12.97	0.999	1.0	.999
OneR	99.2469	0.0614	0.02	0.999	0.997	0.998
Adaboost	99.1319	0.0645	**0.02**	0.998	0.997	0.997

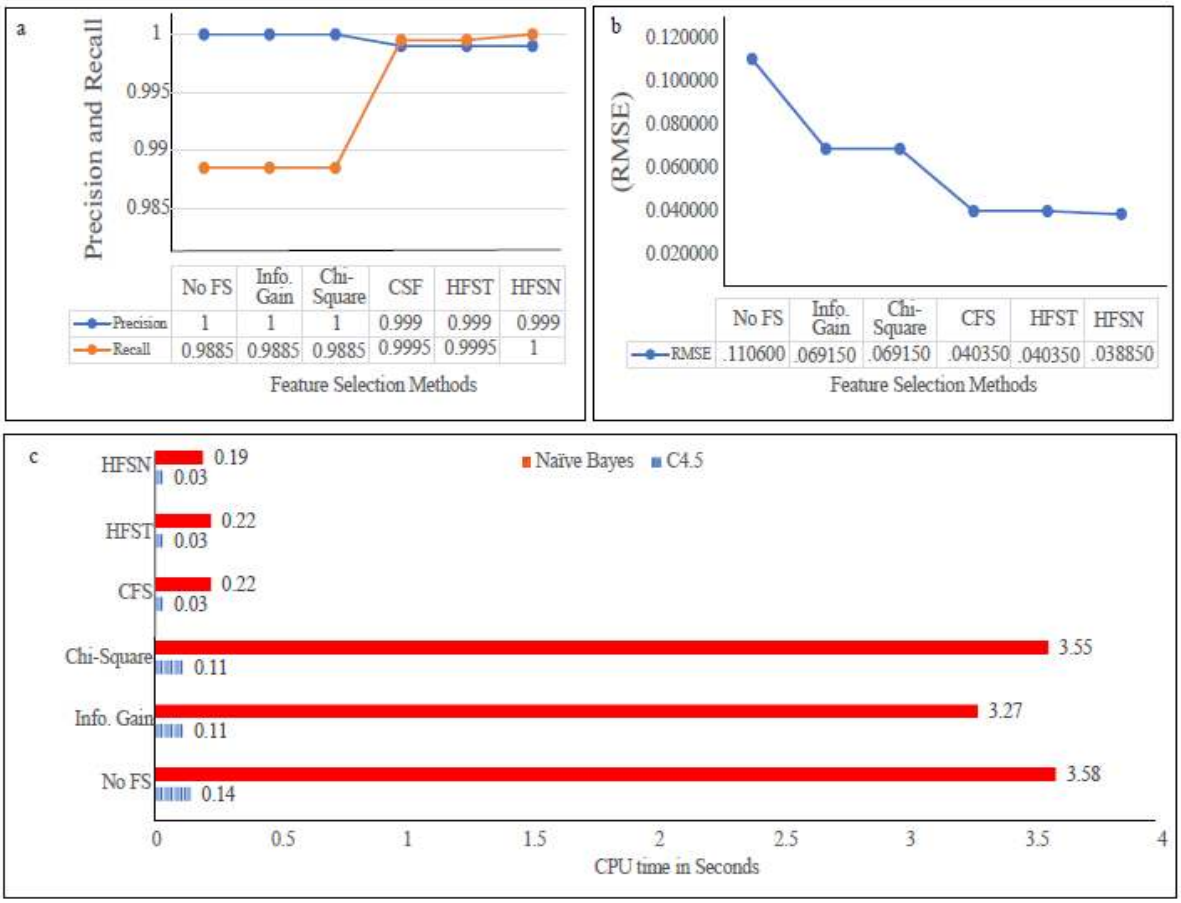

Figure 5.1 *(a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the CICID2017 dataset.*

5.2 ECML/PKDD 2007 Dataset Experiments

ECML/PKDD2007 contains 13 string attributes only the ID attribute has numeric type. The dataset pre-processed by constructing two sets of features. The first set based on statistics of the attribute by calculating length, number of digits, number of characters, number of special characters for each string attribute. The second set of features constructed based on 1-gram, then the two sets of features discretized. Therefore, 96 new features constructed as shown in Table 5.3.

Table 5.3 *The number of instances and features for the ECML/PKDD2007 dataset*

Dataset	Original		Preprocessed	
ECML/PKDD2007	Instances	Features	Instances	Features
	68269	14	50107	96

5.2.1 The feature selection results

The experimental results on the ECML/PKDD 2007 dataset as shown in Figure 5.2 shows the CFS method and the hybrid methods successfully decreased the precision to reach 0.92 for each. The hybrid approach and CFS increased the recall to reach the best value which is 0.98. The accuracy improved with low error with the hybrid and CFS approach. Increase of performance was noticed in the hybrid approached by reducing CPU time. Additionally,

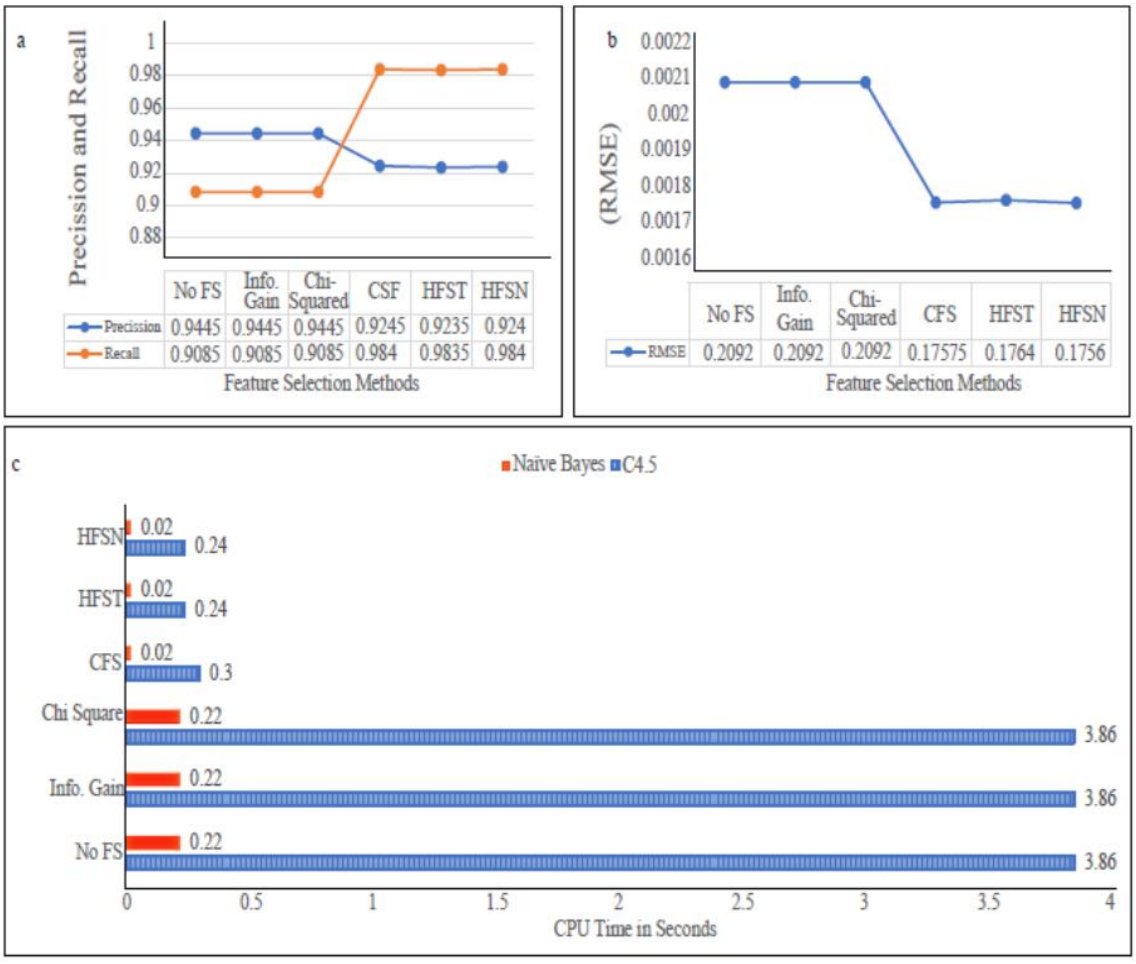

Figure 5.2 *(a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the ECML/PKDD 2007 dataset.*

the hybrid approach with the C4.5 algorithm decreased the CPU time to reach 0.24 seconds while by Naïve Bayes algorithm CPU reached 0.02 seconds.

5.2.2 Classification model results

The framework also tested on ECML/PKDD2007 dataset by using data mining algorithms were chosen for their effectiveness in the web attack detection according to the literature review. As shown in Table 5.4 it is noticeable that the Naïve Bayes has the fastest building time model followed by OneR, Bayes Net, Adaboost and J48 algorithm. SMO achieved better recall than J48 and outperformed the Bayes algorithms with a significant difference in the accuracy. The SMO outperformed all algorithms with 85% of accuracy.

Table 5.4 *Comparison of classification models on ECML/PKDD 2007 Dataset*

Classification Algorithm	Accuracy (%)	RMSE	Building time (Seconds)	Precision	Recall	F-measure
\Bayes Net	83.657	0.1713	0.12	0.932	0.982	0.956
Naïve Bayes	83.661	**0.1713**	**0.05**	0.932	0.982	**0.956**
J48	82.4915	0.1799	0.35	0.916	0.986	0.949
Random Forest	-	**-**	12.12	-	-	-
SMO	**85.0001**	0.2952	1265.8	0.921	**0.992**	0.955
PART	82.4236	0.1818	26.65	0.922	0.978	0.949
JRip	77.3345	0.2194	209.41	0.794	0.992	0.882
OneR	75.72	0.2464	0.04	0.861	0.998	0.924
Adaboost	69.8445	0.2425	0.19	0.698	1.000	0.822

5.3 CSIC 2010 Dataset Experiments

As the ECML/PKDD2007 dataset the CSIC201 dataset contains many string attributes. The pre-processing conducted by applying 1-gram and statistic calculation to construct new features, then discretization applied for newly constructed features. Table 5.5 shows the resulted number of instances and features.

Table 5.5 *The number of instances and features on CSIC2010*

Dataset	Original		Preprocessed	
CSIC2010	Instances	Features	Instances	Features
	223585	17	37539	81

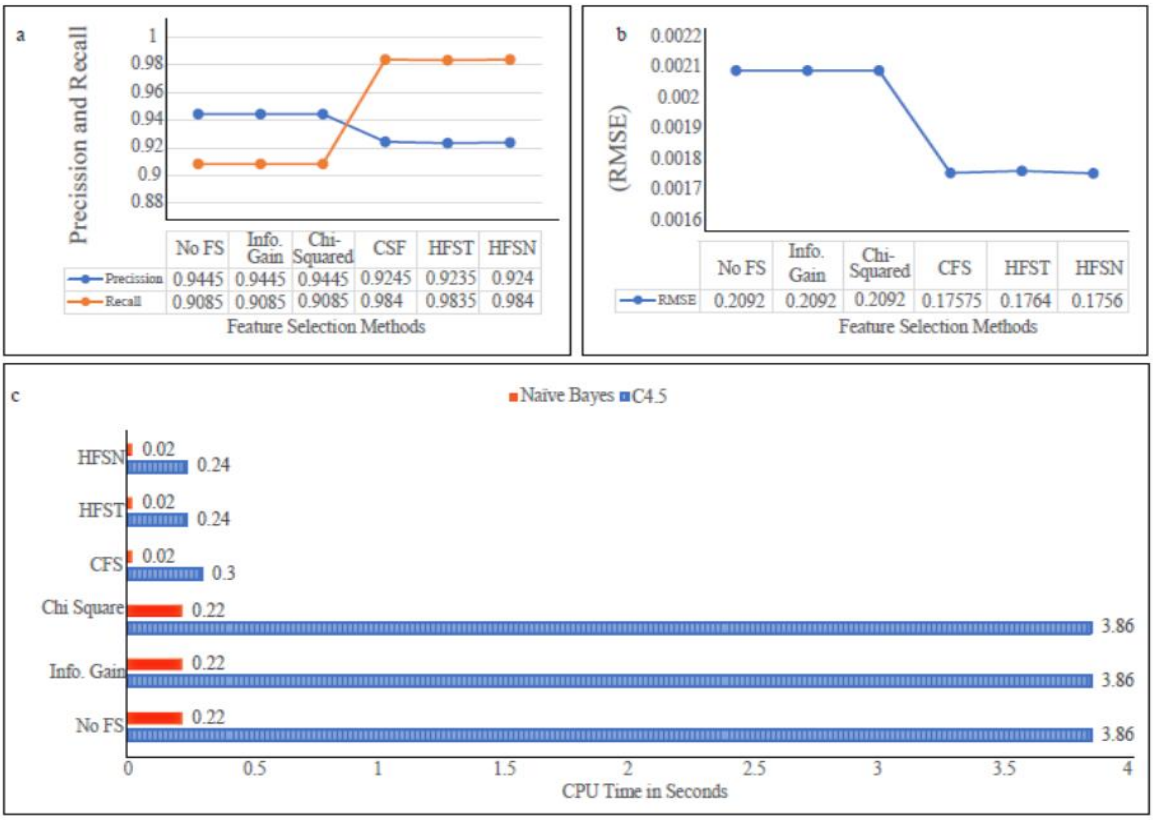

Figure 5.3 *(a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the ECML/PKDD*

5.3.1 The feature selection results

In CSIC 2010 dataset there is no enhancement of accuracy comparing to ECML/PKDD 2007 and CICID2017 datasets. However, there is a significant increase in recall in the hybrid approach compared to the filter methods, with a slight drop in precision. CFS enhances the recall and declines the precision compared to information gain. the accuracy and RMSE of the HFSN achieve better than HFST in both accuracy and RMSE were significantly higher than that of the HFST. No decline in the RMSE was detected in the HFST. The CPU time

decreased sharply by 90% with the hybrid approach in Naïve Bayes. Likewise, the CPU time decreased by 98.8% with HFST decrease of time. What can be clearly seen in Figure 5.3 is the dramatic decline in the number of features with the hybrid approach. The Figure shows that there has been a marked increase in the average accuracy with the hybrid approach.

5.3.2 Classification model results

Table 5.6 compares the results obtained from experiments conducted on CSIC 2010 Datasets by applying the hybrid feature selection approach. As can be seen from Table 5.6 no significant differences in accuracy were found between Naïve Bayes and Bayes Net, likewise in RMSE and building time. The PART algorithm was shown to have the significant highest accuracy and lowest RMSE compared to other algorithms. However, OneR has the fastest building time.

Table 5.6 *Comparison of classification models on CSIC2010 dataset*

Classification Algorithm	Accuracy (%)	RMSE	Building time (Seconds)	Precision	Recall	F-measure
Bayes Net	65.3507	0.4688	0.06	0.760	0.322	0.452
Naïve Bayes	65.3507	0.4688	0.03	0.760	0.322	0.452
J48	65.4652	0.4646	0.2	0.749	0.335	0.463
Random Forest	65.5052	0.4632	3.32	0.751	0.335	0.335
SMO	65.3454	0.5887	100.55	0.744	0.336	0.463
PART	**65.9847**	**0.4617**	0.42	0.772	0.333	0.465
JRip	65.7902	0.4654	2.96	0.758	**0.338**	0.468
OneR	61.8104	0.618	**0.02**	**0.974**	0.144	0.252
Adaboost	61.8104	0.4753	0.49	0.974	0.144	0.252

Overall, these results indicate that the hybrid approach on the 3 datasets achieved a significant increase in accuracy when the average accuracy calculated. Additionally, the number of features declined by high percentage. Figure 5.4 shows the average classification accuracy against the number of features. The data mining algorithms were selected in order to know which algorithms are suitable for web application forensics

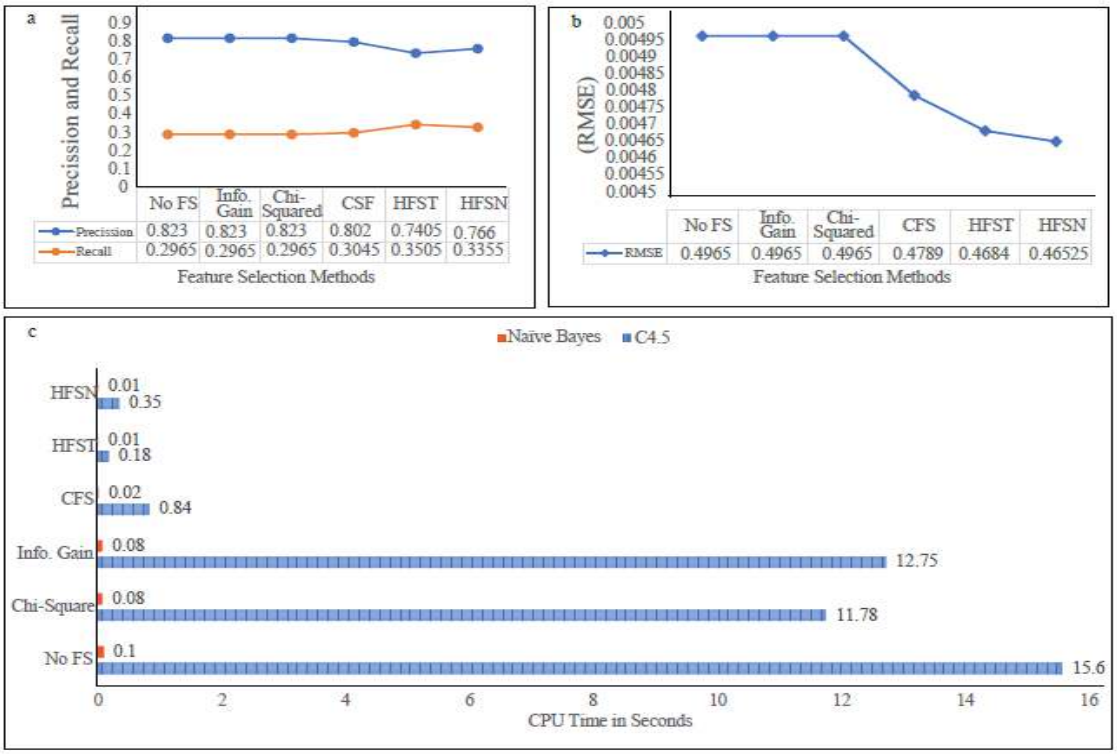

Figure 5.4 *(a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the CSIC 2010 dataset.*

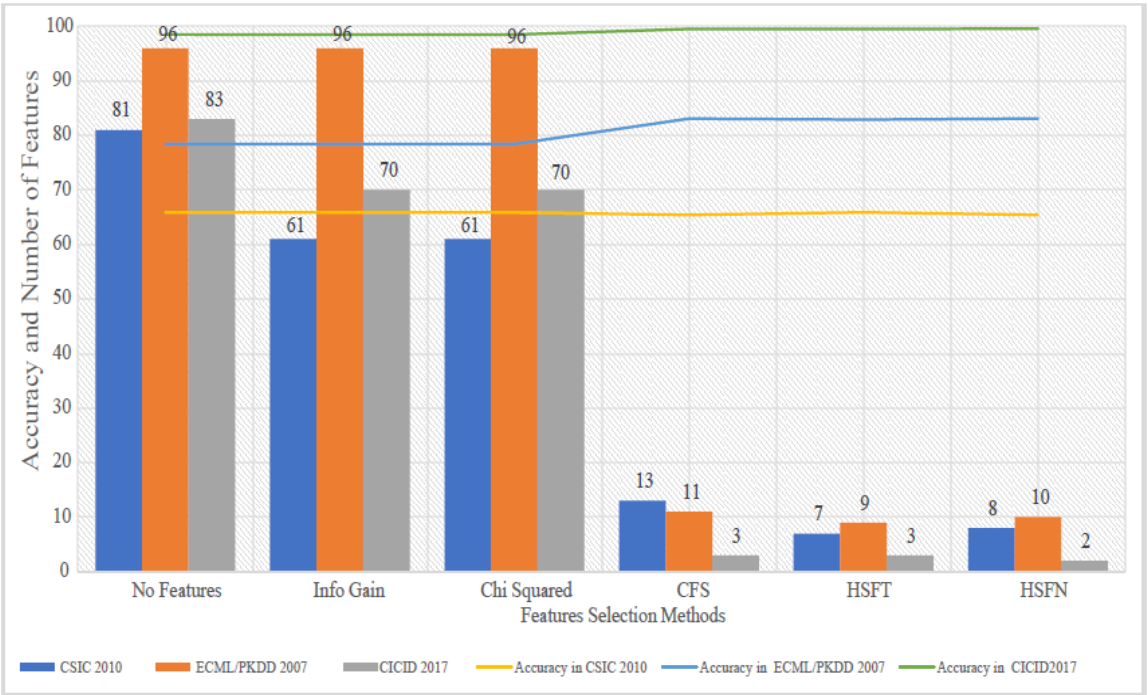

Figure 5.5. *The average classification accuracy against the number of features*

5.4 Discussion

The process of digital forensic investigation depends on examining all possible evidence without missing evidence, and this is what the aim of the digital investigator and what is required by the courts. The algorithms must aim for the same goal, so measuring the effectiveness of the algorithm depends on four important criteria. The most important one is the recall. Getting a high recall means the false negative is less, in another word all actual evidence predicted as evidence. This means that the algorithm has not missed any digital evidence. There is an inverse relationship between the recall and the precession. The precession related to false-positive where the data is not evidence but predicted as evidence performance. By far the greatest demand is for high recall. The experiment for the hybrid approach enhanced the recall by approximately 8% for ECML/PKDD2007 and 20% for CISC2010 until it reached the highest value in CICID2017. Clearly, the increasing of recall was associated with the decreasing of precision. According to the average of F-measure, we can infer that the performance of the hybrid approach is satisfactory.

Table 5.7 *Comparison of accuracy and features selection between the proposed study and related studies*

Dataset	**Our Approach HFSN**		**Nguen et al [24] mRMR**		**Ngyen et al [24] CFS**		**Torrano et al [100] n-gram+CFS**	
	Accuracy +/-	FR	Accuracy +/-	FR	Accuracy +/-	FR	Accuracy +/-	FR
CICID2017	+1%	97%	-	-	-	-	-	-
ECML/PKDD2007	+5%	90%	-4%	80%	-10%	93%	-	-
CSIC2010	+0.15%	90%	-19%	53%	-0.12%	63%	-6%	90%

Table 5.7 compares the hybrid approach with the related literature which utilized the web attack datasets in terms of feature reduction and simple accuracy. Comparison of the findings with those of Nguyen et al [98] confirms the outperformance of our approach in term of accuracy. As shown in Table 5.4 the minimal-Redundancy-Maximal-Relevance (mRMR) in ECML/PKDD 2007 dataset, removed 80% of irrelevant features with a 4.2% decrease in the accuracy. On the other hand, our hybrid approach succeeded in reducing almost 90% of the irrelevant features. Moreover, the accuracy increased 5% in the

ECML/PKDD 2007 dataset. Additionally, our hybrid approach reduced 90% of the features in the CSIC 2010 a dataset with an increase in accuracy by 0.15%. This finding is consistent with that of Nguyen et al. [24], their CFS feature selection 0.4% of C4.5 accuracy.

In contrast to earlier findings in [119, 120], the hybrid approach accuracy of the CSIC 2010 lower by 30%. These results are likely to be related to the pre-processing methods adopted were some of the original features removed from the dataset. Those removed features are the reason for overfitting confirmed by Atienza et al. [99]. Another possible explanation for this is that our pre-processing method increased the features by 79%, that resulted in decreasing of accuracy.

Remarkably, in CICID2017 dataset the irrelevant featured reduced by 97% with a 1% increase in the accuracy. Surprisingly, only 2 features, namely FWD IAT MIN and Init Win bytes backward selected by the hybrid approach. A possible explanation for these results is the importance of forwarding packet timing and forward pack size in the detection of malicious traffic. It can thus be suggested that the time difference between those 2 packets and the total number of bytes that sent to the initial window, maybe an important issue in detection of web attack including the encrypted traffic. This finding is consistent with that of by Iman et al. [113] who run an experiment on the same dataset and considered 4 features related to web attacks, including the two features FWD IAT MIN and Init Win bytes backward. The hybrid approach performed very similar to the CFS method in the ECML/PKDD 2007 and CICID2017 datasets; however, in CSIC 2010 the hybrid approach outperformed the CFS method. Taken together, the experimental results showed that the hybrid approach achieved better recall and accuracy rate and satisfaction performance. The hybrid approach improved the recall with a slight decrease in RMSE compared to the other methods.

Turning now to the experimental evidence on classification models. The experiments carried out by the hybrid approach show its superiority over the rest of the feature selection methods. While the feature selection plays the most important role in enhancing the accuracy and performance. There is no need to repeat the comparison of the classification models in previous studies. Instead, the average of the experiments on the 3 datasets was calculated as shown in Table 5.8.

Table 5.8 *The average evolution of the chosen data mining algorithms*

Classification Algorithm	Accuracy (%)	RMSE	Building time (Seconds)	Precision	Recall	F-measure
Bayes Net	82.8619	**0.226433**	0.086667	0.897	0.768	0.802667
Naïve Bayes	82.86167	0.226467	0.043333	0.897	0.768	0.802667
J48	82.51177	0.227633	0.25	0.888	0.773667	0.804
Random Forest	82.5416	0.2508	7.2	0.875	0.6675	0.6675
SMO	**83.30763**	0.398767	468.6467	0.888	0.776	**0.806**
PART	82.65933	0.227467	9.096667	0.897667	0.770333	0.804667
JRip	80.88367	0.242333	75.11333	0.850333	0.776667	0.783
OneR	78.92577	0.3086	**0.026667**	**0.944667**	0.713	0.724667
Adaboost	76.92893	0.260767	0.233333	0.89	0.713667	0.690333

The most striking result to emerge from the data is that SMO algorithm achieved the highest accuracy, however, the average building time has sharp increasing. Despite the high accuracy of the SMO algorithm, the implementation may bring many problems related to computation and performance. Likewise, PART, JRip and Random forest. Bayes net and naïve Bayes provided reasonable and balanced results, between accuracy and time. The Bayes based algorithms outperformed the J48 algorithm with significant differences in accuracy and time. Overall, the average results are consistent with the literature and there is abundant room for further progress in building algorithms. The proposed framework is theoretical. It has the potential to accept future technical development. If the proposed framework applied in a software, or hardware solution. The results indicate the advantage of using Bayes or tree-based algorithms for good performance and satisfactory results.

As mentioned in the literature review, there are a number of digital investigation frameworks. The phases used in our investigation framework are similar to that used by other frameworks as shown in Table 5.9. However, the proposed framework differs from prior frameworks in a number of important ways as follow:

- One study by Pilli et al [121] integrated into their framework the previous forensic models where have been used in digital forensic investigations. However, in their framework, the investigation process allocated only to networks digital evidence. There are some recommendations in their study to suggest statistical techniques and data mining techniques to search and to identify patterns of attacks. Their framework would have been more interesting if the author had included web attacks. Another limitation their framework can be applied in the case of the presence of security devices only. By contrast, our proposed framework could be applied in both network and web application attacks, additionally, it could work in case of absence of security devices because it depends on various log files.
- Quick and Choo [122] adopted the data mining technology in their framework to reduce the size of data. However, data mining was not used in the analysis and investigation. Moreover, their framework did not replace the traditional investigation and analysis process. Unlike our framework which uses data mining and aims to complete replacing of the digital forensic investigation and analysis phases.
- Kohn et al [123] extracted the basic components from other forensic models, then merged those essential components in order to reduce their number. In a similar vein, the essential components in traditional frameworks used in our framework, furthermore the basic data mining process combined to make the framework capable to make an automated and intelligent analysis of digital evidence which has not done in their framework.
- Carthusian [127] compared and integrated known digital forensics models, then he added new elements that focused more on the flow of information than on digital evidence handlings, such as hypothesis and proof/defense activities. Nevertheless, building hypotheses, then rejecting them or accepting them is related more to the legal aspects than the technical aspects. Similarly, Ieong [125] proposed a framework which is primarily intended to further clarify the technical models for legal professionals. The legal layer was added

in order to answer important questions from a legal perspective. Moreover, to explain the technical layer. In our proposed framework the legal consideration was not taken importantly, however, the proposed framework focuses more on the technical process and digital evidence analysis.

- Baryamureeba and Tushabe [124] improved Carrier and Spafford model [128], by dividing the investigation into two layers primary and a secondary layer. In particular, the traditional linear method of investigation replaced into iterative handling. Likewise, their framework replaced the deployment phase which existed in Carrier and Spafford framework with two phases one for a criminal investigation of physical evidence and another for the criminal investigation of digital evidence. Therefore, their framework focused more on crime scene. In contrary, the proposed framework doesn`t give same attention to the crime scene, because the attacks on the web architecture are clear and the place of pieces of evidence is obvious.
- Beebe and Clark [126] divided the framework into two tiers, a simple tier which is like the rest of the framework and another detailed tier. However, the framework used the traditional techniques related only to the digital investigation as a hard disk is the main evidence.

Table 5.9 *Comparison of the proposed framework with other digital forensic frameworks*

Proposed Framework	Pilli et al [121]	Quick and Choo [122]	Kohn et al [123]	Baryamureeba and Tushabe [124]	Beebe and Clark [126]	Ciardhuáin [127]
Preparation -Preserving -Duplication -Pre-processing	-Preparation and Authorization -Detection of Incident Crime -Incident Response -Collection of Network Traces -Protection and Preservation	-Commence (Scope) -Preparation -Identification and collection -Preservation -Reduce and Store	-Readiness phases -Deployment phases	-Readiness phases	-Preparation Phase -Data Collection Phase	-Awareness -Authorization -Planning -Notification -Search/identification -Collection -Transport Storage
Investigation -Feature Selection -Prediction Model	-Examination -Analysis -Investigation and Attribution	-Evidence analysis -Review & data mining -Open & closed source data	-Physical Crime Scene Investigation phases -Digital Crime Scene Investigation phases	-Deployment phases -Traceback phases -Dynamite phases	-Data Analysis Phase	-Examination -Hypothesis
Presentation -Visualization -Reporting	-Presentation and Review	-Presentation -Completion	-Review phase	-Review phase	-Presentation of Findings Phase -Incident Closure Phase	-Presentation -Proof/defence -Dissemination

A case-study approach was used to ensure that the model has the same result in the real-world scenarios. The scenario was based on the common methods of web applications hacking. In order to conduct the scenario these steps were followed:

- A virtual environment was prepared by using virtual box software. The virtual machine was configured to run Kali Linux and Windows operating systems.
- The vulnerable web application was prepared based on DVWA system which use PHP/MYSQL. The DVWA contains many web application vulnerabilities.

- The DVWA system's security level is set to the lowest level, and PHPID is turned off.
- The vulnerable system was attacked by XSS exploits. The attacks were launched from the Kali Linux operating system.
- Data were collected from the access-log file, then the collected log pre-processed by adapting the procedure in our hybrid approach.
- The pre-processed data are not classified as attacks or normal data. The prediction task was left for the model. After, the model was configured to predict the class type. Table 5.10 shows the high level of recall which reached 98% that confirms the low level of false negative 2%.

Table 5.10 *The evolution of web application attack case study*

TP	FP	FN	TN	Accuracy	Recall
78%	%82	2%	98%	67%	98%

The case study shows the experimental data on web application attacks from real world scenario. The results, as shown in Table 5.10, indicate that the model is able to analysis the digital evidence with high efficiency. The recall reached a high percentage. Interestingly, the correlation between recall and false negative is related to few absences of digital evidences which is important requirement in digital forensic investigation.

5.5 Summary

This chapter began by providing the experiment result for each dataset that conducted on the hybrid feature selection method, beside the experiments that validated the proposed digital forensic framework. It went on to compare the result of the experiments with previous results in literature in case of the hybrid feature selection approach. Additionally, comparison between different classification algorithm was conducted. The chapter has reviewed the key aspects of the proposed framework comparing to existence of digital forensic frameworks.

6 CONCLUSIONS AND FUTURE WORK

The purpose of this research was to propose a data mining based novel comprehensive web application forensic framework which can be used to help in digital evidence analysis. The second aim was to conduct a series of experiments on web application attacks datasets, to select the appropriate data-mining algorithms for web application attacks investigation. A framework was built to achieve these aims and to answer the research questions:

How a Framework Based on Data mining be effectively designed to analyses web application attacks for forensic purposes?

The main findings of the experiments are chapter-specific and were summarized within the respective experiments and results chapter: (EXPERIMENTS AND RESULTS). This section will synthesize the findings to answer the study's research questions:

1. What are the requirements and methodology for developing web attacks forensic framework based on data mining?

The requirements are based on the essential's phases in digital forensic investigation and data mining. These requirements have been met in the proposed framework, in terms of documentation and chain of custody as well as seizing the digital evidence and ensure that it is not tampered by an unauthorized person. Another requirement is the use of scientific techniques in each process of digital forensic investigation, which lead to recognition in accordance with the laws concerned. The proposed framework includes the collection and preparation of the data, moreover it includes the extraction of the relevant features. Then, with optimal features, the algorithms applied to produce satisfactory results.

2. What are the methods for web attack`s data collection to run the experiment?

These methods specified in the RESEARCH METHODOLOGY chapter. In the chapter, all methods, plans and alternative plans that can be used for data collection are described. Data has been collected from three different datasets, including the traffic that can be generated in the web architecture environment. Gathering data containing the HTTP traffic is important. Besides collecting data from different network devices. We found that the main place of the digital evidence is the Logs, it is a convenient place to collect relevant web application attacks data.

3. How the collected data pre-processed?

In this research, a new algorithm for dealing with data collected from web traffic was constructed. Furthermore, the algorithm handles all text data and extracts data that can be analyzed by different algorithms. Equally important, the proposed algorithm extracts statistical and other characteristics based on n-gram. Certainly, each step of this algorithm has been built by integration and mapping previous studies steps that have proven effectiveness in the pre-processing.

4. Which data mining algorithms should be implemented?

A series of controlled experiments were conducted on different datasets. These experiments confirmed the ability of the proposed framework in the detection of digital evidence with high accuracy, recall, and performance. This was done by using a sufficient number of algorithms. which have already proved high accuracy in detecting attacks and have now proved effective in creating digital evidence. As a result, two types of algorithms can be implemented namely, decision tree-based algorithms, and naïve theorem based algorithms. The high recall algorithms are suitable for high accuracy while the high precision algorithms suitable for high performance.

5. Which consideration should be considered for the validation of the framework?

Prior to this study, it was difficult to make predictions about how a data mining based framework be effective in analyzing the web application attacks for forensic purposes. This study has gone some way towards enhancing the traditional digital forensic framework and has confirmed that data mining models with a proposed framework are effective in terms of stability, accuracy, and recall. This framework has applied most of the forensic requirements for finding digital evidence of web attacks. Therefore, the insights gained from this research could be assistance in a future automation system for web attacks forensics.

6.1 Recommendation for future research

Future research and direction may be towards automation and reduction of humanitarian intervention in digital forensic. One of the problems that still exist is the traditional way of securing and documenting digital evidence. So far, the chain of custody is often done by paper and is often prone to human error A technique like hashing contributes

to the integration process. However, it does not offer more services such as authentication, auditing and access control. Using techniques such as blockchain will not only contribute to create more effective and reliable digital records but also could connect digital units in the automated digital forensic framework. Few efforts have been initiated in this direction, and we are planning to contribute to it in future studies. More broadly research is also needed to determine the ability to apply deceptive data model in digital forensic. This would also be a fruitful area for further work.

6.2 Limitation of the study

This study was limited by the absence of real web application attacks and live system. We have had two plans as described in the research method to collect web application attacks traffic. Many security, privacy, and administration reasons prevented us from the implementation of the first plan within the campus network of the university. Therefore, we resorted to an alternative plan by using existing datasets. It was not possible to assess all data mining algorithms; therefore, it is unknown if there are different models which could perform better. The selected algorithm has proven to have high accuracy and high performance on the same web attacks problem. The datasets contain the binary class problem and the other with multiclass problems. Therefore, only algorithms which could solve both problems together were selected. Not all machine learning algorithms could work on both types.

REFERENCE

[1] Symantec (2019). *Symantec's 2019 Internet Security Threat Report*. Retrieved from Symantec website: https://www.symantec.com/en/sg/security-center/threat-report.

[2] Cyberedge Group (2019). *2019 Cyberthreat Defense Report*. Retrieved from Cyberege group website: https://cyber-edge.com/cdr/.

[3] Acunetix (20`19). *Web Application Vulnerability Report 2019.* Retrieved from Acunetix website: https://www.acunetix.com/acunetix-web-application-vulnerability-report/.

[4] Monga, M., Paleari, R., & Passerini, E. (2009, May). A hybrid analysis framework for detecting web application vulnerabilities. In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems* (pp. 25-32). IEEE Computer Society.

[5] Santos, O., & Taylor, R. (2019). *CompTIA PenTest+ Cert Guide*. Pearson IT Certification.

[6] Rosen, L. S. R., & Shklar, L. (2009). Web Application Architecture: Principles, Protocol and Practices.

[7] Najera-Gutierrez, G., & Ansari, J. A. (2018). *Web Penetration Testing with Kali Linux: Explore the methods and tools of ethical hacking with Kali Linux*. Packt Publishing Ltd.

[8] Bugliesi, M., Calzavara, S., & Focardi, R. (2017). Formal methods for web security. *Journal of Logical and Algebraic Methods in Programming*, *87*, 110-126.

[9] Gupta, M. K., Govil, M. C., & Singh, G. (2014, May). Static analysis approaches to detect SQL injection and cross site scripting vulnerabilities in web applications: A survey. In *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)* (pp. 1-5). IEEE.

[10] Antunes, N., & Vieira, M. (2009, September). Detecting SQL injection vulnerabilities in web services. In *2009 Fourth Latin-American Symposium on Dependable Computing* (pp. 17-24). IEEE.

[11] Ghanbari, Z., Rahmani, Y., Ghaffarian, H., & Ahmadzadegan, M. H. (2015, November). Comparative approach to web application firewalls. In *2015 2nd*

International Conference on Knowledge-Based Engineering and Innovation (KBEI) (pp. 808-812). IEEE.

[12] Justin Crist (2007). *Web Based Attacks, SANS Institute*. Sans 2007 Retrieved from Sans website: http://www. sans.org/reading_room/whitepapers/application/web-basedattacks_2053.

[13] Phil, P. R. (2014). OWASP Top 10: The Top 10 Most Critical Web Application Security Threats Enhanced with Text Analytics and Content by PageKicker Robot Phil 73.

[14] CWE, C. T., & Top, S. A. N. S. (25). Most Dangerous Software Errors, 2011.T.

[15] Kokkonen, T. (2016). Anomaly-based online intrusion detection system as a sensor for cyber security situational awareness system. *Jyväskylä studies in computing*, (251).

[16] Husny, H. R. M., Abdullah, D., Seid, N., & Solaiman, M. A. (2017). Web Application Firewall With Telegram Bot Integration. *Journal of Computing Technologies and Creative Content (JTeC)*, *2*(1), 46-55.

[17] Olayemi, O., Antti, V., Keijo, H., & Pekka, T. (2017). Security issues in smart homes and mobile health system: threat analysis, possible countermeasures and lessons learned.

[18] Clarke-Salt, J. (2009). *SQL injection attacks and defense*. Elsevier.

[19] Byrne, P. (2006). Application firewalls in a defence-in-depth design. *Network Security*, *2006*(9), 9-11.

[20] Jia, W. C., Hu, R. G., & Shi, F. (2016, July). Feature Design and Selection Based on Web Application-Oriented Active Threat Awareness Model. In *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)* (pp. 597-600). IEEE.

[21] Prandl, S., Lazarescu, M., & Pham, D. S. (2015, December). A study of web application firewall solutions. In *International Conference on Information Systems Security* (pp. 501-510). Springer, Cham.

[22] Shugrue, D. (2017). Fighting application threats with cloud-based WAFs. *Network Security*, *2017*(6), 5-8.

[23] Appelt, D., Panichella, A., & Briand, L. (2017, October). Automatically repairing web application firewalls based on successful SQL injection attacks. In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 339-350). IEEE.

[24] Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., & Franke, K. (2011). Application of the generic feature selection measure in detection of web attacks. In *Computational Intelligence in Security for Information Systems*(pp. 25-32). Springer, Berlin, Heidelberg.

[25] Fan, W. K. G. (2012, July). An adaptive anomaly detection of WEB-based attacks. In *2012 7th International Conference on Computer Science & Education (ICCSE)* (pp. 690-694). IEEE.

[26] Park, Y., & Park, J. (2008, November). Web application intrusion detection system for input validation attack. In *2008 Third International Conference on Convergence and Hybrid Information Technology* (Vol. 2, pp. 498-504). IEEE.

[27] Vigna, G., Robertson, W., Kher, V., & Kemmerer, R. A. (2003, December). A stateful intrusion detection system for world-wide web servers. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.* (pp. 34-43). IEEE.

[28] Adeva, J. J. G., & Atxa, J. M. P. (2007). Intrusion detection in web applications using text mining. *Engineering Applications of Artificial Intelligence*, *20*(4), 555-566.

[29] Niksefat, S., Ahaniha, M. M., Sadeghiyan, B., & Shajari, M. (2010, September). Toward specification-based intrusion detection for web applications. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 510-511). Springer, Berlin, Heidelberg.

[30] Park, Y., & Park, J. (2008, November). Web application intrusion detection system for input validation attack. In *2008 Third International Conference on Convergence and Hybrid Information Technology* (Vol. 2, pp. 498-504). IEEE.

[31] Yang, C. H., & Shen, C. H. (2009). Implement web attack detection engine with snort by using modsecurity core rules. In *Graduate Institute of Information and Computer Education*. National Kaohsiung Normal University Kaohsiung.

[32] Andrekanic, A., & Gamble, R. (2012, June). Architecting web service attack detection handlers. In *2012 IEEE 19th International Conference on Web Services* (pp. 130-137). IEEE.

[33] Leu, F. Y., & Yang, T. Y. (2003, October). A host-based real-time intrusion detection system with data mining and forensic techniques. In *IEEE 37th Annual 2003 International Carnahan Conference onSecurity Technology, 2003. Proceedings.* (pp. 580-586). IEEE.

[34] Khairkar, A. D., Kshirsagar, D. D., & Kumar, S. (2013, April). Ontology for detection of web attacks. In *2013 International Conference on Communication Systems and Network Technologies* (pp. 612-615). IEEE.

[35] Corona, I., & Giacinto, G. (2010, September). Detection of server-side web attacks. In *Proceedings of the First Workshop on Applications of Pattern Analysis* (pp. 160-166).

[36] Gupta, M. K., Govil, M. C., & Singh, G. (2014, July). An approach to minimize false positive in SQLI vulnerabilities detection techniques through data mining. In *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)* (pp. 407-410). IEEE.

[37] Bhat, V. H., Rao, P. G., Abhilash, R. V., Shenoy, P. D., Venugopal, K. R., & Patnaik, L. M. (2010, February). A novel data generation approach for digital forensic application in data mining. In *2010 Second International Conference on Machine Learning and Computing* (pp. 86-90). IEEE.

[38] Ko, R., & Choo, R. (2015). *The cloud security ecosystem: technical, legal, business and management issues*. Syngress.

[39] Muda, A. K., Choo, Y. H., Abraham, A., & Srihari, S. N. (Eds.). (2014). *Computational intelligence in digital forensics: forensic investigation and applications*. Springer International Publishing.

[40] Tallón-Ballesteros, A. J., & Riquelme, J. C. (2014). Data mining methods applied to a digital forensics task for supervised machine learning. In *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications* (pp. 413-428). Springer, Cham.

[41] Soltani, S., & Seno, S. A. H. (2017, October). A survey on digital evidence collection and analysis. In *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)* (pp. 247-253). IEEE.

[42] Nelson, B., Phillips, A., & Steuart, C. (2014). *Guide to computer forensics and investigations*. Cengage learning.

[43] Kyaw, A. K., Sioquim, F., & Joseph, J. (2015, November). Dictionary attack on Wordpress: Security and forensic analysis. In *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)* (pp. 158-164). IEEE.

[44] Khobragade, P. K., & Malik, L. G. (2014, April). Data generation and analysis for digital forensic application using data mining. In *2014 Fourth International Conference on Communication Systems and Network Technologies* (pp. 458-462). IEEE.

[45] Jain, N., & Kalbande, D. R. (2015, January). Digital forensic framework using feedback and case history keeper. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)* (pp. 1-6). IEEE.

[46] Ussath, M., Cheng, F., & Meinel, C. (2015, July). Concept for a security investigation framework. In *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-5). IEEE.

[47] Manesh, T., El-atty, S. M. A., Sha, M. M., Brijith, B., & Vivekanandan, K. (2015, November). Forensic investigation framework for VoIP protocol. In *2015 First International Conference on Anti-Cybercrime (ICACC)* (pp. 1-7). IEEE.

[48] Lazzez, A., & Slimani, T. (2015). Forensics investigation of web application security attacks. *Int. J. Comput. Netw. Inf. Secur*, *7*(3), 10-17.

[49] Wazzan, M. A., & Awadh, M. H. (2015, August). Towards Improving Web Attack Detection: Highlighting the Significant Factors. In *2015 5th International Conference on IT Convergence and Security (ICITCS)* (pp. 1-5). IEEE.

[50] Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the internet*. Academic press.

[51] Ashcroft, J. (2001). A Guide for First Responders. *United States Dep. Justice Off. Justice*, 93.

[52] Årnes, A. (Ed.). (2017). *Digital forensics*. John Wiley & Sons.

[53] Brenner, S. W. (2006). DIGITAL EVIDENCE AND COMPUTER CRIME: FORENSIC SCIENCE, COMPUTERS AND THE INTERNET.

[54] Goodison, S. E., Davis, R. C., & Jackson, B. A. (2015). Digital evidence and the US criminal justice system. *Identifying Technology and Other Needs to More Effectively Acquire and Utilize Digital Evidence. Priority Criminal Justice Needs Initiative. Rand Corporation*.

[55] Jianhui, L. I. N. (2008, October). A Web Forensic System Based On Semantic Checking. In *2008 International Symposium on Computational Intelligence and Design* (Vol. 1, pp. 99-102). IEEE.

[56] Kerr, O. S. (2005). Digital evidence and the new criminal procedure. *Colum. L. Rev.*, *105*, 279.

[57] Pichan, A., Lazarescu, M., & Soh, S. T. (2018). Towards a practical cloud forensics logging framework. *Journal of information security and applications*, *42*, 18-28.

[58] Suteva, N., Mileva, A., & Loleski, M. (2015). Finding forensic evidence for several web attacks. *International Journal of Internet Technology and Secured Transactions*, *6*(1), 64-78.

[59] Hraiz, S. (2017, May). Challenges of digital forensic investigation in cloud computing. In *2017 8th International Conference on Information Technology (ICIT)* (pp. 568-571). IEEE.

[60] Kyaw, A. K., Sioquim, F., & Joseph, J. (2015, November). Dictionary attack on Wordpress: Security and forensic analysis. In *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)* (pp. 158-164). IEEE.

[61] Khobragade, P. K., & Malik, L. G. (2014, April). Data generation and analysis for digital forensic application using data mining. In *2014 Fourth International Conference on Communication Systems and Network Technologies* (pp. 458-462). IEEE.

[62] Meyer, R., & Cid, C. (2008). Detecting attacks on web applications from log files. *Sans Institute*, *26*.

[63] Alzahrani, A., Alqazzaz, A., Zhu, Y., Fu, H., & Almashfi, N. (2017, May). Web application security tools analysis. In *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*(pp. 237-242). IEEE.

[64] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[65] Roiger, R. J. (2017). *Data mining: a tutorial-based primer*. Chapman and Hall/CRC.

[66] Mirza, N., Patil, B., Mirza, T., & Auti, R. (2017, June). Evaluating efficiency of classifier for email spam detector using hybrid feature selection approaches. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 735-740). IEEE..

[67] Deepashri, K., & Ashwini, K. (2017). Survey on Techniques of Data Mining and its Applications. *International Journal of Emerging Research in Management &Technology*, *6*, 198-201.

[68] Jain, N., & Srivastava, V. (2013). Data mining techniques: a survey paper. *IJRET: International Journal of Research in Engineering and Technology*, *2*(11), 2319-1163.

[69] Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

[70] Chellapilla, K., & Simard, P. Y. (2005). Using machine learning to break visual human interaction proofs (HIPs). In *Advances in neural information processing systems* (pp. 265-272).

[71] Hotho, A., Nürnberger, A., & Paaß, G. (2005, May). A brief survey of text mining. In *Ldv Forum* (Vol. 20, No. 1, pp. 19-62).

[72] Lu, Q., Li, X., & Dong, Y. (2018). Structure preserving unsupervised feature selection. *Neurocomputing*, *301*, 36-45.

[73] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

[74] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70-79.

[75] Hu, L., Gao, W., Zhao, K., Zhang, P., & Wang, F. (2018). Feature selection considering two types of feature relevancy and feature interdependency. *Expert Systems with Applications*, *93*, 423-434.

[76] Hidayah, N. M., Faizal, M. A., Selamat, S. R., Fadhlee, R. M., & Ramzi, W. A. W. (2017). Revealing the feature influence in HTTP botnet detection. *International Journal of Communication Networks and Information Security*, *9*(2), 274.

[77] Quintana, M., Uribe, S., Sánchez, F., & Álvarez, F. (2015). Recommendation techniques in forensic data analysis: a new approach.

[78] Adebayo, O. J., ASuleiman, I., Ade, A. Y., Ganiyu, S. O., & Alabi, I. O. (2015, November). Digital Forensic analysis for enhancing information security. In *2015 International Conference on Cyberspace (CYBER-Abuja)* (pp. 38-44). IEEE.

[79] Mouhtaropoulos, A., Dimotikalis, P., & Li, C. T. (2013, November). Applying a Digital forensic readiness framework: Three case studies. In *2013 IEEE International Conference on Technologies for Homeland Security (HST)* (pp. 217-223). IEEE.

[80] Olivier, M. S. (2015, August). Towards a digital forensic science. In *2015 Information Security for South Africa (ISSA)*(pp. 1-8). IEEE.

[81] Ab Rahman, N. H., Glisson, W. B., Yang, Y., & Choo, K. K. R. (2016). Forensic-by-design framework for cyber-physical cloud systems. *IEEE Cloud Computing*, *3*(1), 50-59.

[82] Akarawita, I. U., Perera, A. B., & Atukorale, A. (2015, August). ANDROPHSY-forensic framework for Android. In *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 250-258). IEEE.

[83] Joshi, A., & Geetha, V. (2014, July). SQL Injection detection using machine learning. In *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)* (pp. 1111-1115). IEEE.

[84] Komiya, R., Paik, I., & Hisada, M. (2011, September). Classification of malicious web code by machine learning. In *2011 3rd International Conference on Awareness Science and Technology (iCAST)* (pp. 406-411). IEEE.

[85] Bharathi, A. S., & Shilpa, R. (2014). A survey on crime data analysis of data mining using clustering techniques. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(8), 9-13.

[86] Nampoothiri, A. P., & Madhavu, M. L. (2015, August). Email forensic analysis based on k-means clustering. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 814-817). IEEE.

[87] Elbasiony, R. M., Sallam, E. A., Eltobely, T. E., & Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, *4*(4), 753-762.

[88] Sharma, S., Kumar, M., & Dubey, K. A Data Mining Based Approach for Mitigating Attacks on Web Services.

[89] Hanmanthu, B., Ram, B. R., & Niranjan, P. (2015, January). SQL Injection Attack prevention based on decision tree classification. In *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)* (pp. 1-5). IEEE.

[90] Nadiammai, G. V., & Hemalatha, M. (2014). Effective approach toward Intrusion Detection System using data mining techniques. *Egyptian Informatics Journal*, *15*(1), 37-50.

[91] Promrit, N., & Mingkhwan, A. (2015, March). Traffic flow classification and visualization for network forensic analysis. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications* (pp. 358-364). IEEE.

[92] Ohrui, M., Kikuchi, H., & Terada, M. (2010, September). Mining association rules consisting of download servers from distributed honeypot observation. In *2010 13th International Conference on Network-Based Information Systems* (pp. 541-545). IEEE.

[93] Tahir, S., & Iqbal, W. (2015, November). Big Data—an evolving concern for forensic investigators. In *2015 First International Conference on Anti-Cybercrime (ICACC)* (pp. 1-6). IEEE.

[94] Halboob, W., Mahmod, R., Abulaish, M., Abbas, H., & Saleem, K. (2015, April). Data warehousing based computer forensics investigation framework. In *2015 12th*

International Conference on Information Technology-New Generations (pp. 163-168). IEEE.

[95] Gupta, M. K., Govil, M. C., & Singh, G. (2014, July). An approach to minimize false positive in SQLI vulnerabilities detection techniques through data mining. In *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)* (pp. 407-410). IEEE.

[96] Kruegel, C., Vigna, G., & Robertson, W. (2005). A multi-model approach to the detection of web-based attacks. *Computer Networks*, *48*(5), 717-738.

[97] Robertson, W., Vigna, G., Kruegel, C., & Kemmerer, R. A. (2006, February). Using generalization and characterization techniques in the anomaly-based detection of web attacks. In *NDSS*.

[98] Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., & Franke, K. (2011). Application of the generic feature selection measure in detection of web attacks. In *Computational Intelligence in Security for Information Systems* (pp. 25-32). Springer, Berlin, Heidelberg.

[99] Atienza, D., Herrero, Á., & Corchado, E. (2015, June). Neural analysis of http traffic for web attack detection. In *Computational Intelligence in Security for Information Systems Conference* (pp. 201-212). Springer, Cham.

[100] Torrano-Gimenez, C., Nguyen, H. T., Alvarez, G., Petrović, S., & Franke, K. (2011, May). Applying feature selection to payload-based web application firewalls. In *2011 Third International Workshop on Security and Communication Networks (IWSCN)*(pp. 75-81). IEEE.

[101] Torrano-Gimenez, C., Nguyen, H. T., Alvarez, G., & Franke, K. (2015). Combining expert knowledge with automatic feature extraction for reliable web attack detection. *Security and Communication Networks*, *8*(16), 2750-2767.

[102] Zhang, Z., George, R., & Shujaee, K. (2016, March). Efficient detection of anomolous HTTP payloads in networks. In *SoutheastCon 2016* (pp. 1-3). IEEE.

[103] Wressnegger, C., Schwenk, G., Arp, D., & Rieck, K. (2013, November). A close look on n-grams in intrusion detection: anomaly detection vs. classification. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security* (pp. 67-76). ACM.

[104] Nascimento, G., & Correia, M. (2011, June). Anomaly-based intrusion detection in software as a service. In *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 19-24). IEEE.

[105] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., ... & Zissman, M. A. (2000, January). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*(Vol. 2, pp. 12-26). IEEE.

[106] Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer networks*, *34*(4), 579-595.

[107] Raıssi, C., Brissaud, J., Dray, G., Poncelet, P., Roche, M., & Teisseire, M. (2007, September). Web analyzing traffic challenge: description and results. In *Proceedings of the ECML/PKDD* (pp. 47-52).

[108] Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., & Franke, K. (2011). Application of the generic feature selection measure in detection of web attacks. In *Computational Intelligence in Security for Information Systems* (pp. 25-32). Springer, Berlin, Heidelberg.

[109] Halfond, W. G., & Orso, A. (2005, November). AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering* (pp. 174-183). ACM.

[110] Han, E. E. (2015, August). Detection of web application attacks with request length module and regex pattern analysis. In *International Conference on Genetic and Evolutionary Computing* (pp. 157-165). Springer, Cham.

[111] Canali, D., & Balzarotti, D. (2013, February). Behind the scenes of online attacks: an analysis of exploitation behaviors on the web.

[112] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017, January). Characterization of Tor Traffic using Time based Features. In *ICISSP* (pp. 253-262).

[113] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018, January). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *ICISSP* (pp. 108-116).

[114] Hall, M. A. (1999). Correlation-based feature selection for machine learning.

[115] Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, *21*(2), 137-146.

[116] Allix, K., Bissyandé, T. F., Jérome, Q., Klein, J., & Le Traon, Y. (2014, March). Large-scale machine learning-based malware detection: confronting the 10-fold cross validation scheme with reality. In *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy* (pp. 163-166). ACM.

[117] Sachowski, J. (2019). *Implementing digital forensic readiness: From reactive to proactive process*. CRC Press.

[118] Yusoff, Y., Ismail, R., & Hassan, Z. (2011). Common phases of computer forensics investigation models. *International Journal of Computer Science & Information Technology*, *3*(3), 17-31.

[119] Kozik, R., Choraś, M., Renk, R., & Hołubowicz, W. (2015, September). A Proposal of algorithm for web applications cyber attack detection. In *IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 680-687). Springer, Berlin, Heidelberg.

[120] Althubiti, S., Yuan, X., & Esterline, A. (2017). Analyzing HTTP requests for web intrusion detection.

[121] Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). A generic framework for network forensics. *International Journal of Computer Applications*, *1*(11), 1-6.

[122] Quick, D., & Choo, K. K. R. (2014). Data reduction and data mining framework for digital forensic evidence: storage, intelligence, review and archive. *Trends & Issues in Crime and Criminal Justice*, *480*, 1-11.

[123] Kohn, M. D., Eloff, M. M., & Eloff, J. H. (2013). Integrated digital forensic process model. *Computers & Security*, *38*, 103-115.

[124] Baryamureeba, V., & Tushabe, F. (2004, August). The enhanced digital investigation process model. In *Proceedings of the Fourth Digital Forensic Research Workshop* (pp. 1-9).

[125] Ieong, R. S. (2006). FORZA–Digital forensics investigation framework that incorporate legal issues. digital investigation, 3, 29-36

[126] Beebe, N. L., & Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, *2*(2), 147-167.

[127] Ciardhuáin, S. Ó. (2004). An extended model of cybercrime investigations. *International Journal of Digital Evidence*, *3*(1), 1-22.

[128] Carrier, B., & Spafford, E. H. (2003). Getting Physical with the Investigative Process International Journal of Digital Evidence. Fall 2003.

[129] Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

[130] Alnabulsi, H., Islam, R., & Mamun, Q. (2017, June). A novel algorithm to protect code injection attacks. In International Conference on Applications and Techniques in Cyber Security and Intelligence (pp. 281-292). Edizioni della Normale, Cham.

[131] Daneshgadeh, S., & Baykal, N. (2017, December). DDoS attack modeling and detection using smo. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 432-436). IEEE.

[132] Frank, E., & Witten, I. H. (1998). Generating accurate rule sets without global optimization.

[133] Cohen, W. W., & Singer, Y. (1999). A simple, fast, and effective rule learner. *AAAI/IAAI*, *99*, 335-342.

[134] Dong, Y., & Zhang, Y. (2017). Adaptively detecting malicious queries in web attacks. *arXiv preprint arXiv:1701.07774*.

[135] Nguyen, H. A., & Choi, D. (2008, October). Application of data mining to network intrusion detection: classifier selection model. In *Asia-Pacific Network Operations and Management Symposium* (pp. 399-408). Springer, Berlin, Heidelberg.

[136] Karar, A. (2013). Preventing SQL-BasedAttacks Using Intrusion Detection System. *International Journal of Science and Engineering Applications*, *2*(6), 145-150.

[137] Cao, L. C. (2006, August). Detecting web-based attacks by machine learning. In *2006 International Conference on Machine Learning and Cybernetics* (pp. 2737-2742). IEEE.

[138] Babiker, M., Karaarslan, E., & Hoscan, Y. (2018, March). Web application attack detection and forensics: A survey. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1-6). IEEE.