

Web-based attacks detection using deep learning techniques: a comprehensive review

Lujain Alghofaili¹, Dina M. Ibrahim^{1,2}

¹Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia

²Department of Computers and Control Engineering, Faculty of Engineering, Tanta University, Tanta, Egypt

Article Info

Article history:

Received Aug 25, 2024

Revised Mar 1, 2025

Accepted Mar 26, 2025

Keywords:

Artificial intelligence

Deep learning

Machine learning

Neural networks

Web applications

Web-based attacks

ABSTRACT

Web applications are utilized extensively by a broad user base, and the services provided by these applications assist enterprises in enhancing the quality of their service operations as well as increasing their revenue or resources. To gain control of web servers, attackers will frequently attempt to modify the web requests that are sent by users from web applications. Attacks that are based on the web can be detected to help avoid the manipulation of web applications. In addition, a variety of research has offered many methods, one of which is artificial intelligence (AI), which is the method that has been utilized the most frequently to identify web-based attacks recently. When it comes to the protection of web applications, anomaly detection techniques used by intrusion prevention systems are preferred. Deep learning, often known as DL, is going to be covered in this paper as anomaly-based web attack detection methods and machine learning (ML) techniques. With the purpose of organizing our selected techniques into a comprehensive framework that encourages future studies, we first explained the most concepts that related to web-based attacks detection, then we moved on to discuss the most prevalent web risks and may provide inherent difficulties for keeping web applications safe. We classify previous studies on detecting web attacks into two categories: DL and ML. Lastly, we go over the features of the previously utilized datasets in summary form.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Lujain Alghofaili

Department of Information Technology, College of Computer, Qassim University

Buraydah 51452, Saudi Arabia

Email: Lujainlg@gmail.com

1. INTRODUCTION

With the development of technology, tasks performance became easier in real life whereas web applications play a major role through the presence of smart devices and permanent internet connection in providing several services depending on various functional aspects. However, web applications can be defined as the program that can run in a web browser, this comprises a wide range of topics as well as numerous examples and applications. Providing capabilities to the user without requiring them to download and install software is one of the most important features of the web applications. In addition, web applications can be updated regularly as introduced by Alzahrani *et al.* [1]. These days, many countries utilize web applications to help their communities in facilitating their daily procedures in different parts of life such as: education, entertainment, medical, governmental, and entertainment. Moreover, web applications provide fast execution of procedures and time utilization. In view of this, web applications act as mediators to deliver responses or send requests.

For example, Saudi Arabia is at the forefront of countries that rely on web applications for online services, some of them are: Absher governmental, Sehhaty healthcare, Madrassati education. The web application is a complicated platform, since it is considered the most popular platform to transmit the information and services over the internet network. That complexity is having an impact in providing security and it became a well-liked and worthwhile target for security threats according to Li and Xue [2]. 1.5 trillion dollars were reportedly generated by cybercrime in 2018. Therefore, no business is safe from a cyber-attack of any size, even large or small companies as mentioned by Maithem and Al-Sultany [3]. The hackers and threats are targeting 42% of web applications of different attacks, which will have an impact on the web information security. That is because of global distribution of the web applications with over-usage and availability was conducted by Tekerek [4]. Therefore, web application security is a big issue that should be considered in order to protect the sensitive information and preserve the web to function as expected. Web application security is a target topic in scientific research and many researchers introduced several solutions or many approaches to protect the web application. Many studies have proposed different technologies and methods to detect and prevent web-based attacks. Nonetheless, detection is considered as an important step to protect web applications and help to prevent attacks from occurring. Therefore, in the attack detection scope the artificial intelligence (AI) science was widely proposed and applied. One of the 21st century's most potently revolutionary technologies is AI. In particular, deep learning (DL) has been used to classify the HTTP requests of web applications between normal or malicious requests and it has been proven to be an effective and significant method. In practice, the neural networks help to detect attack patterns through the training and testing phases according to Maithem and Al-Sultany [3]. This paper has investigated the effects of DL models for detecting web-based attacks. Based on the literature review, earlier studies extensively utilized neural networks such as recurrent neural networks (RNN), convolutional neural networks (CNN), and multilayer perceptron (MLP), achieving notable results in detecting web-based attacks. More recent approaches have focused on models like long short-term memory (LSTM), gated recurrent units (GRU), CNN, and MLP. Likewise, many previous studies used the CSIC2010v2 dataset, whereas the former dataset causes some DL classification algorithms to produce biased findings according to Tekerek [4].

The main contributions in this work are:

- Illustrate the most common risks that target a web application.
- Explain the most concepts that related to web-based attacks detection.
- Discuss the most prevalent web risks and may provide inherent difficulties for keeping web applications safe.
- Classify previous studies on detecting web attacks into two categories: DL and machine learning (ML).
- Explain the main features of the previously utilized datasets in summary form.

2. BACKGROUND

AI as it is one of the most critical aspects that was used to detect the internet attacks specifically the web-based attacks as mentioned by Kumar *et al.* [5]. However, the concepts and definitions section contain the descriptions and clarification of the used concepts in our paper. Next, the most common risks that target a web application.

2.1. Concepts and definitions

Intrusion detection system (IDS) is the operation of continuously monitoring and analyzing events in a computer system or network for indicators of intrusion. However, intrusion detection is usually categorized into three methodologies that are discussed by Hung-Jen *et al.* [6]: signature-based detection (SD), anomaly-based detection (AD) and stateful protocol analysis (SPA). Most previous studies involved the intrusion detection approaches through using the two methodologies: SD and AD as categorize by Kai and Jiankun [7].

Web application is a distributed program that runs on the web platform. It is a crucial component of the modern web ecosystem that allows for the delivery of dynamic information and services. Both server-side and client-side code can be found in a web application according to Li and Xue [2].

As shown in Figure 1, the AI is contained ML as a subset that relies on system training, where ML contains the DL as a subset that applied artificial neural networks which are computing systems inspired by the biological neural networks that constitute brains. AI is a broad topic of computer science related to creating intelligent machines that can accomplish activities that would normally require human intelligence as defined by Neelam [8]. ML is the science of computer algorithms that upgrade themselves over time because of their use as defined by Jordan and Mitchell [9]. ML is the capacity of computer software to learn and reinforce the execution of a collection of tasks gradually as mentioned by Horvitz and Mulligan [10]. DL can be referred to as deep models of neural networks that have deep structures. The core purpose of DL was to tackle broad learning issues in a systematic manner via simulating the human brain systematically as

referred by Pitts and McCulloch [11]. The greatest advantage of DL that plays a role to prove the positive effect of the used DL models is feature learning. Where feature learning means the automatic extraction of features from raw data as mentioned by Lecun *et al.* [12]. Neural network indicates the system that contains neurons whereby it is artificial. In other words, the neural networks are about the chain of algorithms that uses a mechanism that replicates the way the human brain works to detect underlying relationships in a batch of data as introduced by Kamilaris and Prenafeta-Boldú [13].

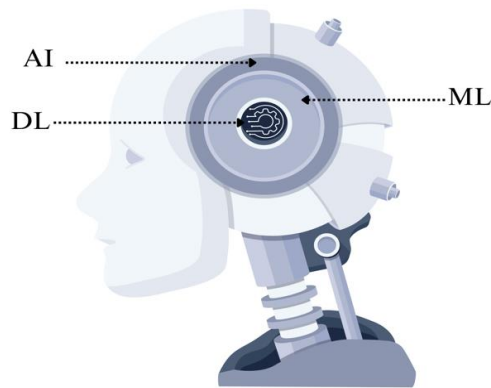


Figure 1. AI, ML, and DL

2.2. Web common risks

The major security threats are considered as the attack that takes advantage of the vulnerability of web applications due to both web application's complexity, and technologies also the spreading use of web applications in nearly all areas. Furthermore, both the lack of security experiences and the new talents of the developers allow the attacks to occur in web applications, common web attacks are listed as follow according to open web application security project (OWASP) [14]:

2.2.1. Broken access control

This attack is ranked as the 1st most common risk of web applications according to OWASP 2021. The broken access control occurs when a malicious user has unauthorized access to data and can perform actions out of their original limits as Figure 2 displays. It resulted in adverse consequences such as failure in the security system that exposes confidential data of the web user and even allows them to modify or destruct the data. However, the broken access control completely relies on breaking the authentication operation i.e., authentication means the user is really who they claim to be as discussed by Hassan *et al.* [15]. Therefore, can be classify the following points the as some reasons of broken access control:

- Uncontrolled redirection of web page.
- Inadequate validation of user credential.
- Sensitive data disclosure misconfiguration.
- Misuse of functions in the code.

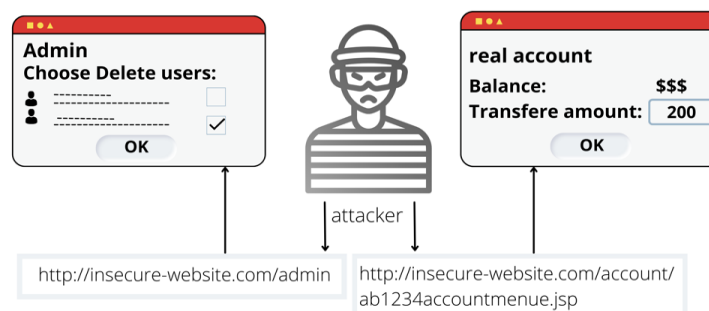


Figure 2. Broken access control [16]

2.2.2. Cryptographic failures

Cryptographic failures are described in Figure 3, previously known as sensitive data exposure, arise whenever a confidential data is exposed or insecurely stored, and it is considered the 2nd most common risks of web applications according to OWASP [14]. The importance of safeguarding data's confidentiality and integrity has grown as the amount of sensitive data has grown. The privacy of users is jeopardized by the disclosure of sensitive data through storage, transmission, and permanent use in a variety of locations as mentioned by Alotaibi *et al.* [17].

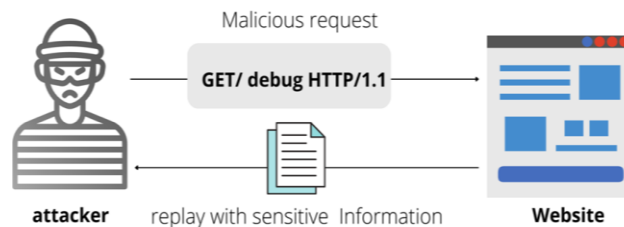


Figure 3. Cryptographic failures [17]

2.2.3. Injection

According to the list of OWASP 2021, the injection is the 3rd most common risk of web applications. In addition, the injection covers the cross-site scripting (XSS) and structured query language (SQL) injection. Figure 4 demonstrates the SQL injection attack which indicates a wide class of attack vectors, it is about inserting malicious code injection technique for executing malicious SQL statements in the database. Moreover, SQL injection is a type of online security flaw that allows an attacker to interfere with a web application's database query. It allows an attacker to disclose data that they wouldn't ordinarily be able to disclose. Mostly, the attackers use the page port accessed by the SQL injection, to be as ordinary web page login. However, SQL injection need to artificial means since the firewall cannot detect this attack according to OWASP [14].

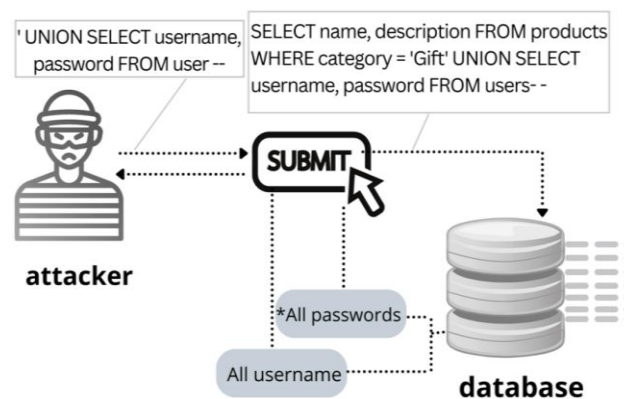


Figure 4. SQL injection representation

XSS is considered one of the web applications security vulnerabilities that happen because of a malicious script injection to the trusted web, Figure 5 explains the process of XSS. This type of vulnerability can create issues for both the server application and the users. Furthermore, the XSS vulnerabilities are categorized into three main types which are stored, reflected and lastly document object model (DOM)-based according to Portswigger [18]. First, the stored XSS vulnerability that happens when the user input is stored in the database and used later in the response page. Next, the reflected XSS vulnerability occurs when the user input is reflected on the immediate response web page lacking the appropriate validation. Lastly, DOM-based XSS vulnerability arises when malicious user input is used by the client-side program that was dynamically obtained from the DOM structure according to OWASP [14].

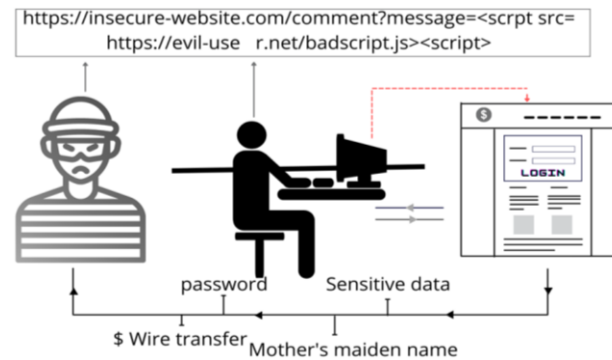


Figure 5. XSS attack representation [19]

2.2.4. Insecure design

Insecure design is a wide term that refers to a variety of flaws, such as “the control design is missing or inefficient”. However, secure design is a strategy and approach that examines threats on a regular basis and guarantees that code is well-designed and tested to avoid known attack methods whereas the insecure design is about a failure or flaw in this process that affects the web application to be vulnerable to an attack. Insecure design is considered as the 4th most common risk according to OWASP [14].

2.2.5. Security misconfiguration

Security misconfigurations are security controls that are inaccurately implemented and defined for a server or a web application which creates a risky environment as Figure 6 displays. It is considered as the 5th most common risk according to OWASP [14]. Further, the security misconfigurations are considered to be an easy target since the misconfigured web servers, cloud instances, and applications are merely detectable, and then turn to be more exploitable which pave the way for attacks. Lastly, this vulnerability occurs whenever a user has illegitimate access to a certain resource as discussed by Loureiro [20].

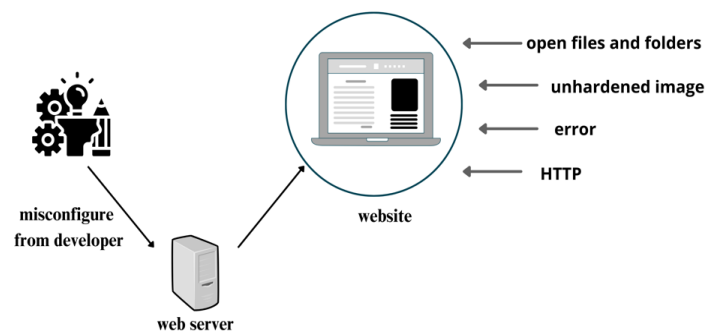


Figure 6. Security misconfiguration [20]

2.2.6. Vulnerable and outdated components

Vulnerable and outdated components are well-known vulnerabilities which are likely to be exploited as in Figure 7. Furthermore, evaluating and testing the risk is quite a struggle as introduced by Pang *et al.* [21]. It is considered as the 6th most common risk according to OWASP [14]. However, applying any prevention techniques designed particularly for detecting vulnerable software components during an initial phase can in fact cut down expenses of software assessing processes which hence could help in creating more solid and reliable software according to Gupta *et al.* [22]. However, some of the reasons behind this risk are:

- Unknown used component versions.
- Vulnerable, unsupported, or out-of-date of OS, web servers, DBMS.
- Not scan the vulnerabilities orderly.
- The underlying platform does not fix or upgrade.
- Updated, upgraded, or patched libraries are not tested for compatibility by program developers.

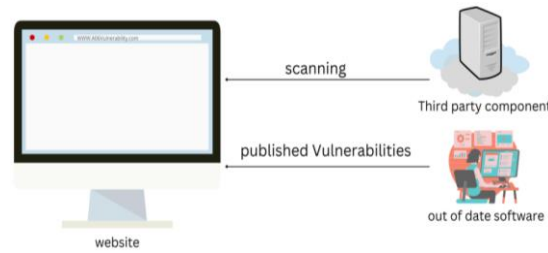


Figure 7. Vulnerable and outdated components [21]

2.2.7. Identification and authentication failures

Identification and authentication failures, previously known as broken authentication, occur whenever there is an illegitimate implementation of user authentication. It is considered as the 7th most common risk according to OWASP [14]. User authentication is critical when a proper web application user cuts their communication while the session is still active, and an intruder uses the exact session to access the system which leads to identity theft and tampering with confidential data. There are many factors that the web application may use which leads to a weak authentication as mentioned by Hassan *et al.* [23]:

- Using automated attacks.
- Permitting weak, basic, default passwords like “Password1”.
- Weak reset password process as in Figure 8.
- Showing the session identifier in the URL.

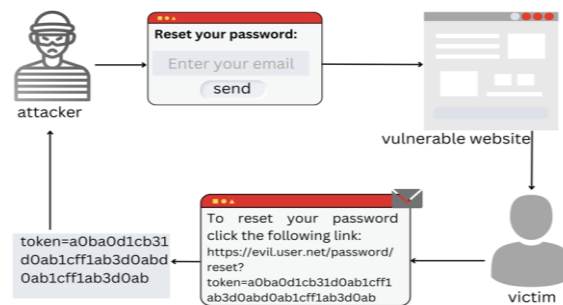


Figure 8. Identification and authentication failures [23]

2.2.8. Software and data integrity failures

Software and data integrity failures happen when a code and infrastructure fail to protect against integrity violations. Further, an insecure CI/CD (CI/CD are continuous integration, continuous delivery, and continuous deployment) pipeline will increase the potential for unauthorized access, malicious code, or even system compromise. The CI/CD is a set of procedures that must be followed to deploy a new software version. It is considered as the 8th most common risk according to OWASP [14]. Finally, there are various applications nowadays that use an auto-update implementation, that updates are downloaded automatically without adequate integrity verification, then implied to reliable application as mentioned by Horvitz and Mulligan [10]. Hence, using auto-update implementation will allow the attackers to upload their update to be distributed and run on all installations as explained in Figure 9.



Figure 9. Software and data integrity failures [10]

2.2.9. Security logging and monitoring failures

The web applications are monitored and logged routinely and in an excellent way result in detecting flaws early in the system. At the same time, any mistakes in detecting or responding to web application attacks result from security logging and monitoring failures. It is considered as the 9th most common risk according to OWASP [14]. The security logging and monitoring failures occur as a response to the following reasons:

- Only local logs are kept.
- As shown in Figure 10, the monitoring process is carried out by a non-skilled employee.
- Insufficient, or ambiguous log messages produced from warnings and errors.
- Events that can be verified are not logged.
- Suspicious activity is not tracked in application and API logs.

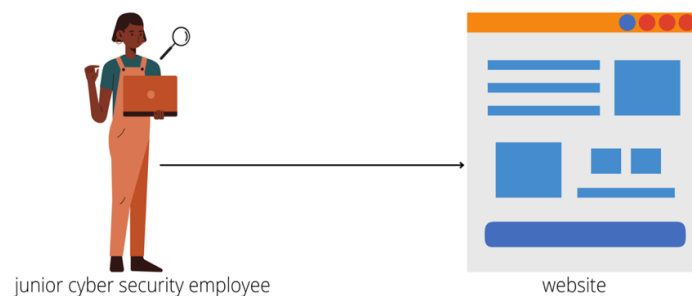


Figure 10. Security logging and monitoring failures

2.2.10. Server-side request forgery

It can be defined as the server-side request forgery (SSRF) as application inducement of the HTTP requests at server-side vulnerability to any arbitrary domain chosen by the attacker as the Figure 11 demonstrates. It is considered as the 10th most common risk. Furthermore, when a web application fetches a remote resource without validating the user-supplied URL, SSRF occurs. The effects of the SSRF risks can be unauthorized procedures, or malicious access to the web application system. In addition, an attacker might use the SSRF vulnerability to execute arbitrary commands according to OWASP [14]. However, there are other multiple web security risks that influenced the web application security does not include on the OWASP top ten web application security risks, as in the sections below.

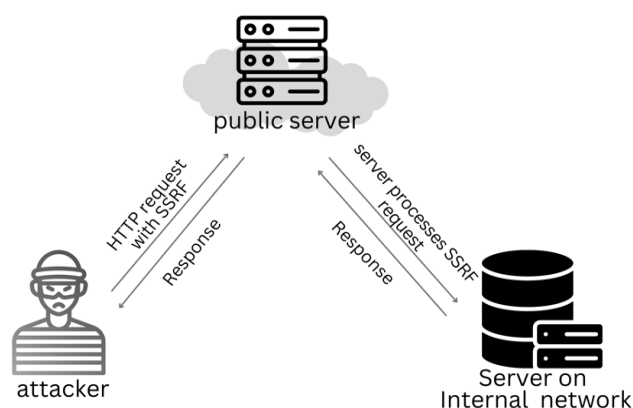


Figure 11. Server-side request forgery [24]

A. Buffer overflow

The buffer overflow is about an irregularity in a program because of a misshapen input; it happens when a malicious program or process tries to write additional data in the fixed-length block of memory as the Figure 12 shown. Shellcode is a sequence of instructions that run a command in software to gain control of or

exploit a compromised machine. However, the buffer is locations set aside in memory to hold data, normally happening through moving between program sections or programs. Moreover, the stack smashing is in the simplest form of buffer overflow that depends on “overwrites a buffer” on the stack to supplant the return address according to OWASP [25]. Therefore, the buffer overflow is used by the attackers to ruin the implementation stack of a web application through transmitting carefully crafted input.

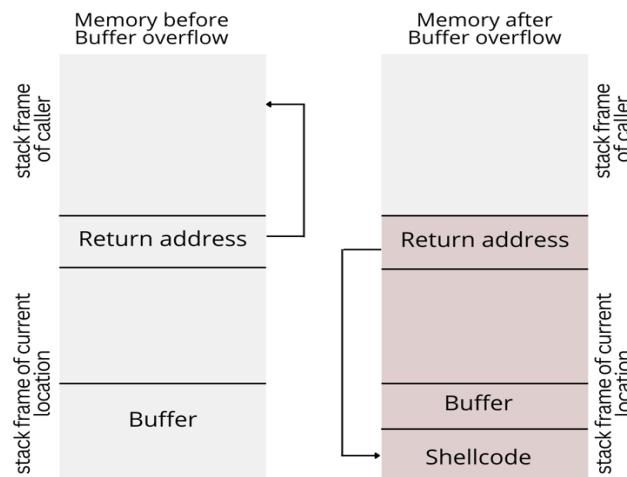


Figure 12. Buffer overflow [26]

B. CRLF injection

The CRLF means carriage return and line feed that consist of a series of the special character input, which demonstrate the end of line marks: carriage return (ASCII 13, \r) line feed (ASCII 10, \n) employed to close the HTTP header information. However, the attack CRLF injection is a code flaw in a software program that happens when the user runs to submit the CRLF inside the application. It is usually occurring via the HTTP parameter or URL according to TeachTarget [27]. There are four main HTTP attacks for the CRLF vulnerability according to OWASP [28]: HTTP request splitting, HTTP response splitting, HTTP request smuggling, and HTTP response smuggling.

C. Parameter tampering

Parameter tampering means when the parameter manipulation is exchanged between the server and client it aims to change the web application data like product price, product quantity, user credentials and permissions. This attack can be carried out by a malevolent user who wants to take advantage of the program for their personal gain, or by an attacker who wants to employ a man-in-the-middle. This data is typically stored in hidden form fields, cookies, or URL query strings and is used to enhance application control and functionality according to OWASP [29].

3. RELATED STUDIES

The proposed study presented by Tekerek [4] aimed to suggest a web-based cyber-attacks detection technique of the web application via CNN, their approach applied on CSIC2010v2 datasets, were used to illustrate the suitability and viability of the CNN architecture. The methodology focuses on detecting anomalous HTTP requests, where the web application is at risk from malicious internal or external anomalous behaviors. To summarize, the data preprocessing was organizing the HTTP requests from the dataset in text form. The goal of data preprocessing is to convert each character-based record into a numerical feature vector that is suitable for binary classification. This study uses two steps to preprocessing. The first stage is to create dictionaries, and the second step is to create matrices. The dictionary was created via the bag-of-words (BoW) technique which is a ML modeling method for extracting features from text. It is noteworthy to mention that CNN has reached an adequate result in detecting web attacks with a high detection rate.

Liang *et al.* [30] offer the RNN with LSTM unit or GRU as a novel anomaly detection approach for the HTTP GET requests. The process starts with tokenizing the URLs, the URL path structure and the query parameter structure are tokenized separately. The proposed approach uses two RNNs, that RNN trained to

“familiarize” with legitimate request patterns. The output of the RNNs is then used by a trained neural network classifier to determine if given requests are unusual. The experiment is worked on two datasets: CSIC2010, URLs from WAF log files dataset. Between the GRU-RNN and the LSTM-RNN: on the CSIC dataset, the LSTM-RNN defeats the GRU-RNN. However, the GRU-RNN outperformed the LSTM-RNN on the WAF logs dataset. Accordingly, by Kuang *et al.* [31] aim to construct a web-based attack detection model, the proposal discusses the two DL models CNN and LSTM systematically. Through combining the CNN and LSTM authors attempt to achieve a novel and functional web application firewall WAF as DeepWAF. The request is first analyzed in the parser to the HTTP header and body, then the HTTP request preprocesses and produces the URL sequence in the preprocessor. The methodology is applied on four models CNN, LSTM, CNN-LSTM, LSTM-CNN. However, through evaluating the four models on the dataset HTTP CSIC2010 the CNN achieved the best percentage in accuracy, F1, and recall. Whereas the LSTM has the minimum false alarm and maximum precision.

Rao *et al.* [32] introduced sparse autoencoder (SAE) with smoothed $l1$ regularization and deep neural network (DNN). The proposed hybrid methodology uses DNN to categorize the multi-class attacks in IDS. However, DNN is a variant of MLP which is a kind of feed forward neural network (FFN) that has more than two layers and single input, one output layer, and two or more invisible layers. Moreover, the study used multiple datasets KDDCup99, NSL-KDD, and UNSW-NB15 respectively. The study's findings demonstrated that SAE-DNN can not only identify known and unidentified attacks but also have a high detection rate. Xiao *et al.* [33] suggested a multi-heads self-attention with CNN which is an approach that precisely detects phishing websites. Hence the authors began to utilize generative adversarial networks (GAN) to set up real phishing URLs and create the balanced training dataset along with actual real URLs. This design can consider URLs as input then determine them as either positive or negative output. These URLs are almost identical to the actual phishing URLs that help to train an unbiased classifier. The authors claimed that the combination of CNN and multi-head self-attention to form a new classifier could enhance the results and considered this study as one of the earliest papers that discussed phishing websites detection.

Jin *et al.* [34] proposed a model for detecting the payload-based web attack detection through using two different types of DL, neural networks that are RNN and autoencoder. Obviously, the HTTP packets contain text as payload. For this reason, the authors suggested using DNN as the technique of extracting the key features through text processing to detect web attacks. On the positive side, this study is better in extracting features than studies which are working manually to extract features. However, the model is capable of detecting the anomaly URLs with ambiguous properties. By using a specially created CNN, Zhang *et al.* [35] present a DL method to identify web threats. Their approach is based on examining HTTP request packets; only a small amount of preprocessing is required, and the laborious feature extraction is carried out by the CNN itself. Their experimental findings using the CSIC 2010 dataset demonstrate that the planned CNN performs well, and that the method detects web attacks to a decent level.

The proposed model presented by Gong *et al.* [36] is using the uncertainty model to field a secure web with the CNN. Similarly, the proposed model consists of two steps for producing the model uncertainty via web logs: a CNN extractor and the Bayesian classifier. The CNN step extracts latent features through raw web logs, whereas the Bayesian classifier detects assaults using the latent features and offers model uncertainty in the form of the variance. However, the authors implemented the experiments on two datasets to evaluate the model effectiveness: Apache-2006 and Apache-2017 datasets. The result of the experiment that was performed on the Apache-2006 dataset achieved higher accuracy than Apache-2017 dataset. Tang *et al.* [37] discusses a straightforward and effective method for SQL injection detection acquired from an artificial neural network that is subcategorized further into three main parts: data preprocessing, feature extraction, and model training. Basically, this method begins to extract 8 types of related features from the analyzed SQL injection data. Then, a huge amount of actual data is employed to train the neural network model. They conduct the experiment with actual data provided by the ISP, so that the model learned may be used in the real environment. The last step is to compare these model training results. MLP is a feedforward artificial neural network that uses an artificial feature extraction method to extract the identical features from the URL as the model input. However, the experiment ends with a high detection rate.

Nan and Sheng [38] proposed a CNN as the calculation of the web information feature extraction structure scheme. The methodology is performed by setting the web network nodes as the neuron's elements and the many connected neurons to the model matrix are selected as a simulation for the effect versus web-based attacks. However, this experiment assessed web attack test simulation using the CSIC2010 dataset. The experiment achieved high performance in detecting the following attacks: remote to local, denial-of-service, user to root, and probe. Similarly, as by Gong *et al.* [39], examined the web attack detection model which is CECOR-Net. This detection model is composed of both CNN and LSTM techniques. Further, this model mainly depends on a character-level input of the HTTP requests which greatly facilitate the data preprocessing. This application scenario involves training the CECOR-Net on a labeled dataset with normal and

abnormal HTTP requests, which then could be used in the front-end of the web server. It should be mentioned that the CECor-Net has the features of both misuse detection and anomaly detection. Moreover, this study has analyzed the extracted features of CECor-Net, that have not been fully discovered yet. Finally, this paper introduced two datasets in the experiment: (A) CSIC2010 (B) collected to test the unknown attack detection. CECor-Net revealed constant performance on detecting unknown attacks.

The proposed approach by Yulianto *et al.* [40] aims to improve the AdaBoost classification performance through use synthetic minority oversampling technique (SMOTE), principal component analysis (PCA), and ensemble feature selection (EFS) on the CICIDS2017 dataset. Therefore, they used the SMOTE to make the layout more sensitive to minority classes, the PCA and EFS used to choose significant attributes from a new dataset as a feature selection and used the AdaBoost in training phase classification. However, after the experiment proven that their proposed method achieves a good result. Uncertainty model was used by Gong *et al.* [41] to evaluate the prediction that is caused by the DL model of the web-based attacks. The aim of this model is to fix data annotation errors for the DL model of the web-based attacks. Moreover, the uncertainty model is an analysis of the whole piece of web log. The following components make up the proposed methodology of the model: 1) a layer that embeds characters to create 2D tensors from raw web logs. 2) A CNN model to take the generated tensors' latent characteristics. 3) A dropout NN layer to present the classification outcome as well as the output variance, which acts as the model uncertainty. Using two datasets that contain real web log: apache2006 and CISC2010 and public benchmark dataset. However, by the aid of GPU acceleration the uncertainty model was able to prove its efficiency.

Van *et al.* [42] performed the anomaly-based network IDS via using techniques of DL. The proposed builds depending on two techniques: autoencoder, restricted Boltzmann machines (RBM), RBMs and autoencoders in DL provide a combination of production model demonstration capabilities and high classification accuracy. The proposed system succeeds in detecting attacks and classifying the attacks in five categories: normal, DoS, Probe, U2R, R2L with high accuracy on the KDDCup99 dataset. Atienza *et al.* [43] introduces the neural projection architectures as a technology to detect the web-based attacks through visual analysis of the http request. They use the unsupervised connectionist projection techniques to identify unknown or never-before-seen attacks. their experiment on three neural visualization approaches: PCA, cooperative maximum likelihood Hebbian learning (CMLHL), self-organizing map (SOM) and with the CISC2010 datasets. None of the models used have been able to distinguish between normal and abnormal traffic. The reason behind the low performance that applied to the CSIC 2010 of the neural visualization is because of the preprocessing process that removes certain information which could help distinguish between different types of traffic.

ML was used by Hoang [44] as a model to detect web attacks through using web logs. However, the model is built on the decision tree algorithm, and it can detect four types of web-based attacks: path traversal, XSS, SQL injection, CMDi. The labeled dataset that is HTTP Param dataset, and real web logs were used in the experiment. After performing the experiment, they noticed the CMDi attacks have low detection rate unlike other attacks, and they mentioned the amount of this attack in the dataset is not sufficient is the reason behind the low rate. But generally, the result confirmed high detection rate in the proposed model. Liu *et al.* [45], a data-driven method for web phishing detection using multilayer perceptrons is presented. Their model is about three steps: data acquisition, data preprocessing, and classification. Moreover, Kaggle was used to collect their own dataset. Model has high results in both test and training accuracy. Garcia *et al.* [46] introduce the ID3 decision tree as web-based attacks detection, commonly with false alarm rates and low missing alarm. Nonetheless, the ID3 is a well-known classifier that uses a predetermined set of instances to create a decision tree. In addition, this study had a hypothesis which is that IDS may generalize on specifying a variety of web attacks if it is presented with a sizable enough set of attack instances. Their experiment was on web application queries dataset, and it achieved a successful result. Vartouni *et al.* [47] introduces the web application firewall WAF depends on anomaly detection. The scheme of the WAF is applied autoencoder as attributes extracted, employing the DL algorithm with the stacked autoencoder (SAE) to create a hierarchical learning process. The isolated forest model is used in SAE's highest level output representation to find abnormalities. The experiments are conducted on the CSIC 2010 dataset. Results demonstrate that deep models perform differently with various SAE structures, and deep models perform better overall.

Pillai and Sharma [48] offers a hybrid unsupervised detection model that uses anomaly-based DL to detect web attacks. The de-noising autoencoder (DAE) and SAE encoded outputs, on the other hand, are combined and sent into the GAN as input to enhance the feature representation's capacity to identify web attacks. Consequently, a unique deep Boltzmann machine (DBM)-bidirectional long short-term memory (Bi-LSTM)-based classification model has been created for categorizing the different types of attacks. which uses Bi-LSTM for multi-class classification and DBM for binary classification to categorize the various attacks. Their result achieves a high accuracy rate.

However, to discuss the critical information of the previous approaches that were used as IDS of web-based attacks. Table 1 summarize the studies that were performed as experimental methods to verify the

approach results. In addition, all approaches are designed for the same reason which have improved the detection techniques via using neural networks. Nevertheless, the table outlines each study's year, used approach, the scope of attack detection, used dataset, and classifies if the study uses ML or DL approach. Apart from several above-mentioned merits, the results are the most important part in each previous approach. Whereas some studies implemented more than an experiment and present multiple results according to applied more than model, use more than dataset, or use different parameters. Therefore, the Table 2 displays the highest result of each study.

Table 1. Summarization of previous techniques for web traffic anomaly detection

Ref	Year	Approach	Attack scope	Dataset	ML/DL
[4]	2021	- CNN	HTTP requests	- CSIC2010v2	DL
[30]	2021	- Without RNN - RNN-LSTM - RNN-GRU	HTTP GET request	- CSIC2010 - Collected dataset.	DL
[31]	2019	CNN LSTM CNN-LSTM LSTM-CNN	HTTP requests	- CSIC 2010	DL
[32]	2021	-SAE-DNN	Multi-class attacks in IDS	- KDDCup99 - NSL-KDD - UNSW-NB15	DL
[33]	2021	-Self-attention CNN	HTTP	- Collected datasets.	DL
[34]	2018	-RNN and autoencoder	HTTP	- Collected dataset.	DL
[35]	2017	-CNN	HTTP requests	- CSIC 2010	DL
[36]	2019	- CNN-Bayesian classifier	Web logs	- Apache-2006 - Apache-2017	DL
[37]	2020	- MLP	HTTP	- Collected dataset.	DL
[38]	2019	-CNN	DoS, Probe, U2R, R2L.	- CSIC2010	DL
[39]	2019	-CECoR-Net	HTTP requests	- CSIC2010 dataset - Collected dataset	DL
[40]	2019	-SMOTE, PCA, EFS	HTTP requests	- CICIDS2017	ML
[41]	2020	-CNN	Web logs HTTP requests	- Apache-2006 - CSIC2010 - public benchmark dataset	DL
[42]	2017	-Autoencoder, RBM	DoS, Probe, U2R, R2L.	- KDDCup99	DL
[43]	2015	-PCA, CMLHL, SOM	HTTP requests	- CSIC2010	DL
[44]	2020	-Decision tree algorithm	Web logs	- Collected dataset. - HTTP Param.	ML
[45]	2020	-MLP	Not mentioned	- Collected dataset.	DL
[46]	2006	-ID3	Web queries	- Collected dataset.	ML
[47]	2018	-SAE	HTTP requests	- CSIC2010	DL
[48]	2023	-DAE-SAE	HTTP requests	- CSIC2010v2	DL

Table 2. Summarization of previous studies machine and DL models and their results

Ref.	Approach	ACC*	PRE	REC	F1	FN	FP	TN	TP	SN	SP	FA
[4]	CNN	0.971	0.974	0.976	0.975	0.024	0.036	0.963	0.975	---	---	---
[30]	With GRU	0.986	---	---	---	---	---	---	---	0.988	0.983	---
[31]	With CNN	0.973	0.977	0.954	0.966	278	138	---	---	---	---	0.015
[32]	With SAE-MLP	1.000	1.000	1.000	1.000	---	---	---	---	---	---	0.017
[33]	Self-CNN	0.921	0.994	0.846	0.914	---	---	---	---	---	---	0.470
[35]	CNN	0.965	---	---	---	---	---	---	---	---	---	---
[36]	CNN- Bayesian	0.984	0.998	0.948	0.972	---	---	---	---	---	---	---
[37]	MLP	0.997	1.000	0.994	---	---	0.000	---	---	---	---	---
[38]	CNN	0.965	---	---	---	---	---	---	---	---	---	0.036
[39]	CECoR-net	0.978	0.985	0.960	0.987	---	---	---	---	---	---	---
[40]	With AdaBoost+ EFS + SMOTE	0.818	0.818	1.000	0.900	0	12250	3	55170	---	---	---
[41]	Uncertainty	0.999	1.000	0.998	0.997	0.112	0.194	0.004	0.057	---	---	---
[44]	SVM+ decision tree+ random forest	0.997	---	---	---	---	---	---	---	---	---	---
[45]	MLP	0.930	---	---	---	---	---	---	---	---	---	---
[47]	SAE	0.883	0.803	0.883	0.841	---	---	---	---	---	0.883	---
[48]	DAE-SAE	0.976	0.988	0.988	0.988	---	---	---	---	---	---	---

*ACC: accuracy, PRE: precision, REC: recall, F1: F1-score, FN: false negative, FP: false positive, TN: true negative, TP: true positive, SN: sensitivity, SP: specificity, FA: false alarm

According to the study by Tekerek [4] the insufficient and diversified datasets that are usually used in web attacks detection approaches. However, the Table 3 contains datasets used in the previous studies to detect web-based attacks that will involve the scope of what each dataset contains, studies that worked with, and its features. The aim of using a dataset that is arranged according to a specific model is to assist in processing the required information. In our scope, the dataset contains the raw data that is necessary to prepare the input for the models. Many previous studies have created their own dataset, and half of these studies merge their dataset with a common dataset. In addition, the studies [30], [33], [34], [37], [39], [44]-[46] are training their approaches and testing them through collected datasets of real data from the web applications.

Table 3. The most commonly used datasets according to the previous models

Dataset	Scope	Range	Ref.	Features
CSIC 2010	HTTP requests	36,000 normal 25,000 abnormal	[30] [31] [35] [38] [39] [41] [43] [47]	- Contains the generated traffic targeted to an E-commerce web application developed. - Covering attacks: SQL injection, buffer overflow, parameter tampering, XSS, information gathering, CRLF injection, files disclosure
CSIC 2010v2	HTTP requests	104,000 normal 119,585 abnormal	[4] [48]	- CSIC2010v2 is the improved and updated version of the previous dataset CSIC 2010 with more requests. - Fix samples are not properly created in CSIC 2010.
Collected from HTTP traffic	HTTP requests	2,500 normal 2,500 abnormal	[30]	- It is a larger dataset without duplicated items. - Unifying domain names, resource paths and attribute keys to merge the different domains of the URLs. - Anomalies requests are dynamic attacks.
KDD CUP99	Network based IDSs	972,780 normal 3,925,650 abnormal	[32] [42]	- Comprises a standard collection of auditable data, including a wide range of intrusions simulated in a military network context. - Dataset contains 41 features and class labels are either attack category or normal.
NSL-KDD	Network based IDSs	60,591 normal 250,436 abnormal	[32]	- Fix the inherent problems of the KDD'99. - The number of records is reasonable. - There are no redundant records. - There are no duplicate records.
UNSW-NB15	Network traffic	93,000 normal 164,673 abnormal	[32]	- The data gathered from network traffic includes both realistic and synthetic modern malicious behaviors. - This dataset has nine types of attacks: Fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, Shellcode, Worms.
Collected from legitimate and phishing URLs	Phishing URLs	5,000 normal 12,003 abnormal	[33]	- From April to July 2018, phishing URLs were validated. - Legal URLs were collected from the 5,000 best websites homepage (2012), and 12,003 phishing URLs were collected from the PhishTank homepage (2019)
Collected from China internet companies	URL	2,953,700 normal 356,215 abnormal	[34]	- These businesses protect the security of many websites and collect about 2 TB of online logs daily.
Apache 2006	Real web log	14,432 records.	[36] [41]	- SAFE - ATTACKED.
Apache 2017	Real web log	796,6387 records.	[36]	- Recorded from November 2016 to August 2018.
CICIDS2017	Web logs	2,359,290 normal 471,453 abnormal	[40]	- Recorded from Monday 13 July 2017 at 9 a.m. to Friday 7 July 2017 at 5 p.m. - Created abstract behaviors of 25 users depending on the HTTP, HTTPS, FTP, SSH, and email protocols. - The attacks are: Brute Force FTP, Brute Force SSH, DoS, Heartbleed, web attack, infiltration, Botnet, DDoS,
HTTP Param	Web logs	19304 normal 11763 abnormal	[44]	- Anomaly values divided into various attack types: SQL injection attacks, XSS, command injection, path traversal - Different HTTP param: CSIC2010, SQLMAP, XSSYA, Vega Scanner, FuzzDB.

4. DATASETS RELATED TO THE STUDY

In the context of web-based attack detection, datasets are required for neural network training and evaluation whereas some previous approaches select more than one dataset or create their own dataset [49]. Obviously, the selection of the dataset depends on the research purpose and research scope. As a result, the previous studies of web-based attack detection have focused on the dataset utilized, making the dataset selection a sensitive function, especially when considering that the estimate and classify procedures are

influenced by the number and types of the dataset features according to Sommer and Paxson [50]. The dataset is an ordered group of data that have been gathered as tabular pattern depending on one or more of features, and it is saved permanently. Nonetheless, a dataset can consist of a collection of tables, schema, and other items. Find the correct data from an authenticated source is a very important function in any scientific experience. Therefore, the next section will explain the dataset selection characteristics. According to Sommer and Paxson [50] the lack of suitable public datasets is considered as one of the big challenges that faced anomaly detection. Therefore, this section is going to highlight the most important characteristics for the common datasets, and the necessary properties of the dataset's web-based attacks detection.

Ring *et al.* [51] had classified the dataset's properties under five categories to support the researchers to find a suitable dataset. Because of their belief that relevance of features vary depending on the evaluation scenario and should not be rated in a survey in general, as illustrated in Figure 13. The five categories are:

- a) General information: it includes the public information about the dataset, and it has four properties as Figure 13 described: year of creation, an intrusion detection dataset's age has a significant impact, due to the new attack scenarios and concepts emerge and drift in the network traffic. Therefore, the year that a dataset's underlying network traffic was captured is more important than the year in which it was published. Ring *et al.* [51] was focused on public availability property, whereas the datasets should be made publicly available so that different IDS can be compared as the first principle in Wilkinson *et al.* [52]. Normal Traffic, this property specifies whether a dataset contains usual user behavior whereas the dataset does not include normal user behavior property need to merge with other dataset for evaluating an IDS. Attack traffic, various attack scenarios should be included in the datasets. This characteristic specifies whether a data set contains malicious network traffic.

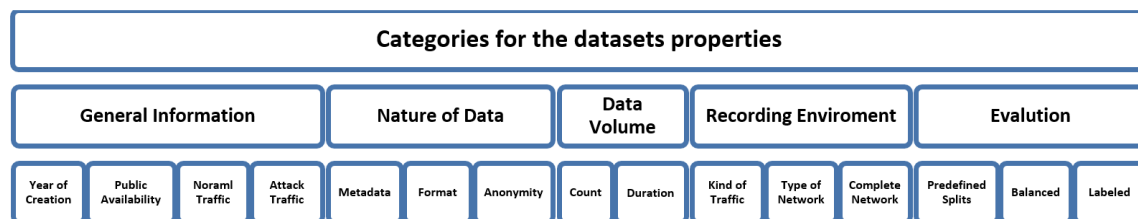


Figure 13. Categories for the datasets properties

- b) Nature of data: the three properties of this category describe the format of the datasets and the presence of information. Figure 13 show this category properties: metadata property, it includes the information such as: IP address, scenario of attack, and network traffic. The second property is format, it is essential where the datasets for network intrusion detection come in a variety of formats. According to Ring *et al.* [51] divided the formats into three types: flow-based network traffic and packet-based network traffic. Other types of data sets like flow-based traces with different features derived from packet-based data or even a log file stored on the host. The intrusion detection datasets are frequently not published or only available in anonymized form due to privacy concerns. Anonymity property specifies whether data has been anonymized, as well as which attributes have been affected.
- c) Data volume: the properties are two in this category describe the size and length of data sets: the count property determines the size of a dataset, which is either the number of enclosed packets/flows/points or the physical size in gigabytes. The duration property, the recording time of each dataset is provided by this property according to Ring *et al.* [51].
- d) Recording environment: the network environment and settings in which the datasets are acquired are defined by three properties in this category. Ring *et al.* [51] classified the kind of traffic into three main possible origins of network traffic which are real, emulated, or synthetic. They mean by real, happens when capturing the real network traffic through a productive network environment. While the emulated refers to the real network traffic were captured by a test bed or emulated network environment. And the term "synthetic" refers to network traffic that was generated artificially rather than captured by a real network device. Type of network, the underlying network environment where the dataset was formed is described by this property. Since the different environments necessitate different security solutions, and evaluation datasets must be tailored to the circumstances. Complete network, both Ring *et al.* [51], Sharafaldin *et al.* [53] agree on the importance of this property. It determines if the dataset contains all network traffic from a network with multiple hosts, routers, and so on.

- e) Evaluation: Figure 13 also describes the three properties in this category, which are all related to evaluating intrusion detection algorithms. The attributes denote the availability of preset subsets, the balanced dataset, and presence of labeled. Predefined splits property, it indicates whether a dataset includes preset subgroups for training and evaluation to help in to compare the quality of different IDS. The balanced property specifies whether datasets are balanced in terms of class labels. Labeled, this property indicates whether datasets are labeled. Labeled datasets are required for training supervised IDS and the evaluating both supervised and unsupervised intrusion detection methods Ring *et al.* [51].

The scope of this work is around the AD methodology especially the web-based attack detection area, because of its effectiveness in detecting new and unanticipated attacks. Thus, there few datasets that discussed by Ring *et al.* [51] whereas there are other datasets that are widely used in web-based attack detection does not discuss by Ring *et al.* [51]. Therefore, in the following sections, the KDD CUP99, NSL-KDD, UNSW-NB15, CSIC 010, CSIC2010v2, and CICIDS2017 datasets will be examined according to the all the five categories.

Table 4 illustrates those datasets according to the categories including the properties for each category. In the first category, general information, with its four properties is compared between the datasets. Whereas the first column is determined the datasets and the second column demonstrates when each dataset is exactly created, where it turns out that the oldest data sets are KDD CUP 99 and NSL-KDD which were created in 1998 and the datasets UNSW-NB15 was created in 2015. Moreover, the first version of CSIC2010 was created in 2007, and the CSIC2010v2 was in 2010. Last, the most recent dataset is CICIDS2017 was created in 2017. Third column describes the availability of e datasets. However, all the datasets were assigned by YES which means that are public availability. Whereas the fourth column declares whether the dataset contains the usual user behavior, and all datasets were assigned by YES, which means that all of them had normal traffic. Last column discusses if the dataset contains the attack traffic and the result showed that all the datasets contained it [51], [54]-[56].

Table 4. Examining the datasets according to the properties for each category

Dataset	Categories for the datasets properties												
	Year of creation	General information			Nature of data		Data volume		Recording environment			Evaluation	
		Public availability	Normal traffic	Attack traffic	Meta data	Format	Count	Duration	Kind of traffic	Type of network	Complete network	Predefined spilt	Labeled
KDD CUP99	1998	YES	YES	YES	NO	Other	5 million records	not specified	Emulated	Small	YES	YES	YES
NSL-KDD	1998	YES	YES	YES	NO	Other	150 thousand records	not specified	Emulated	Small	YES	YES	YES
UNSW-NB15	2015	YES	YES	YES	YES	Packet, Other	2 million records	31 hours	Emulated	Small	YES	NO	YES
CSIC 2010	2007	YES	YES	YES	NO	Other	61 thousand records	not specified	Real	Small	YES	YES	YES
CSIC 2010v2	2010	YES	YES	YES	NO	Other	224 thousand records	not specified	Real	Small	YES	YES	YES
CICIDA 2017	2017	YES	YES	YES	YES	Other	2.83 million records	5 days	Emulated	Small	YES	NO	YES

Similarly, for the nature of data category with its three properties is observed for the different datasets. Therefore, the first column is for the dataset names. The second column is the metadata property and there are only two datasets that contained the metadata information which are UNSW-NB15 and CICIDS2017. The third column specifies the network intrusion detection dataset format, all the datasets have other format except the UNSW-NB15 contains the packet and other format. The last column is determined if the dataset is anonymity, there is no anonymity dataset from the five. Likewise, the data volume category, with its two properties, is examined for the different datasets. Therefore, the first column is for the dataset names. The second column is to specify the size of the dataset records, which turn out that KDD CUP 99 is the largest dataset with five million records and the smallest dataset is CSIC 2010 with 61 thousand records. Whereas the NSL-KDD have 150 thousand records and CSIC2010v2 have 224 thousand records, the UNSW-NB15 have 2 million records and the CICIDS2017 have 2.8 million records. The second column is the duration property that provides the dataset recording time, all the datasets did not specify the time, except the UNSW-NB15 was recording data for 31 hours and the CICIDS2017 dataset was recording data for 5 days.

Correspondingly, the fourth category, recording environment, with its three properties. The first column is for the dataset names. Moreover, the second column is to define the network traffic of each dataset, the KDD CUP99, NSL-KDD, UNSW-NB15, and CICIDS2017 have emulated traffic. Though the CSIC2010 and CSIC2010v2 are all real traffic. The third column is network type, and all the datasets are small networks.

The last column determines if the dataset was recorded data in a complete network, all the datasets recorded in a complete network. Finally, the fifth category, evaluation, with its three properties is inspected for the different datasets. Therefore, the first column is for the dataset names. The second column is to define if the dataset comes along with predefined subsets for training and testing, all datasets are predefined spilt except the UNSW0NB15 and CICIDS2017. The third column is to determine if the dataset balanced or not, all the datasets are unbalanced. The last column is to the labeled datasets; all datasets are labelled.

5. FINDING AND DISSCUTION

Based on the previous sections, we can come up with the following findings:

- Firstly, though the DL techniques are a branch of ML, but they are implemented in different ways. However, the DL were applied more than ML in the web attack detection recently, that because it does not the need the human feature learning, whereas DL directly learns feature representations from the original data and the DL approaches outperform ML significantly for large datasets [46].
- Secondly, the studies of the web-based attack detection are limited, because of that insufficient dataset according to [4]. The Figure 14 represents the rate of each used dataset. The studies [30], [33]–[35], [39], [45], [46] have created their own dataset, and some of these studies merge their dataset and popular dataset. We aim to measure multiple deep neural networks; it is more meaningful to us to use popular dataset. Figure 14 appears that CSIC2010 is the most common dataset in our literature review, therefore we select the updated version of the CSIC2010 which is CSIC2010v2 because of: it has more features and more organized, contain more data (HTTP requests), as well as it fixed the samples of CSIC2010 that were not properly created.

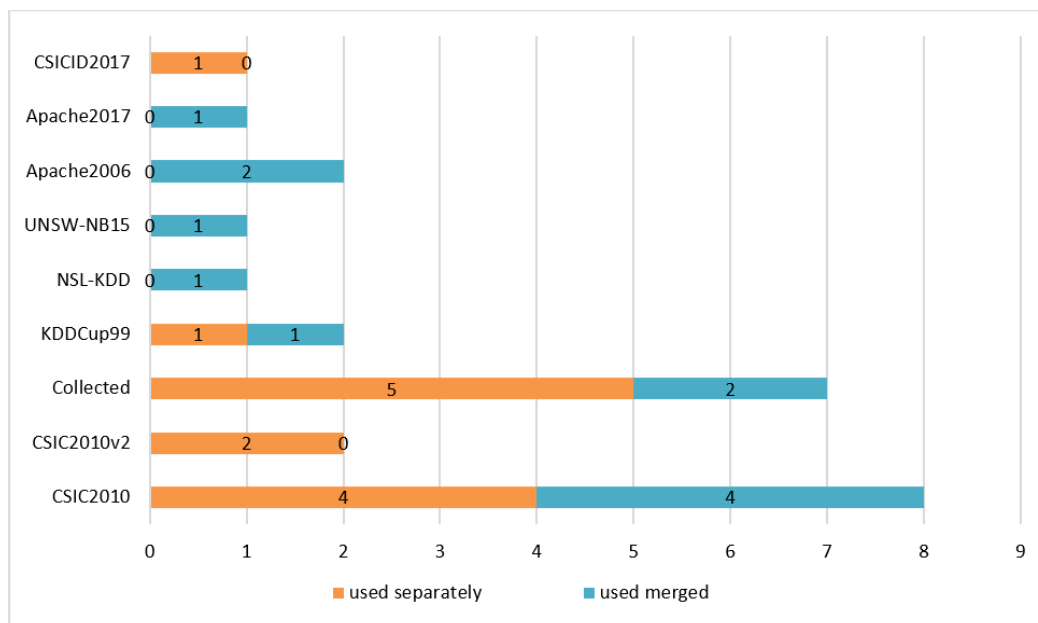


Figure 14. The used datasets in previous studies

- Thirdly, many issues that faced in previous approaches such as the study [30] mentioned that current anomaly web-based attack detection methods struggle unsatisfactory accuracy. Moreover, the performance will affect negatively on the results if the preprocessing process, not implemented well. As in the study [41] they achieved low performance because of eliminating data that can assist in differentiating the various traffic types in the preprocessing process.
- Fourthly, to evaluate the neural networks, previous studies used several performance metrics to evaluate their models. There are multiple metrics that have been used in the previous studies that are: accuracy, precision, recall, F-measure, confusion metrics, sensitivity, specificity, and false alarm. However, following Table 5 will determine the performance metrics that have been used in the previous studies. Figure 15 represents the rate of the performance metrics that are used in the previous studies.

Table 5. The used performance metrics in previous studies

Study	ACC	PRE	REC	F1	FN	FP	TN	TP	SN	SP	FA
[4]	✓	✓	✓	✓	✓	✓	✓	✓			
[30]	✓								✓	✓	
[31]	✓	✓	✓	✓	✓	✓					✓
[32]	✓	✓	✓	✓							✓
[33]	✓	✓	✓	✓							
[35]	✓										
[36]	✓	✓	✓	✓							
[37]	✓	✓	✓			✓					
[38]	✓										✓
[39]	✓	✓	✓	✓							
[40]	✓	✓	✓	✓	✓	✓	✓	✓			
[41]	✓	✓	✓	✓	✓	✓	✓	✓			
[44]	✓										
[45]	✓										
[47]	✓	✓	✓	✓						✓	
[48]	✓	✓	✓	✓							

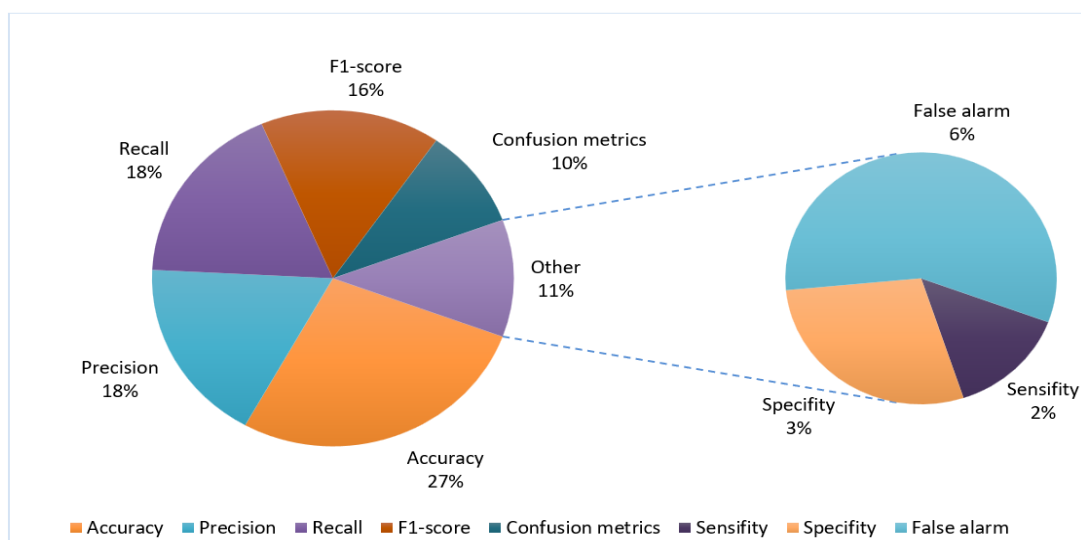


Figure 15. The used performance metrics in previous studies

However, some previous studies present different results according to perform more than experiment because of following factors like: the number of proposed models, as in [30]-[34], [40], [43], the number of hyper-parameters [5], and the number of the used datasets [30], [32], [36], [39], [41]. Nonetheless, using more than one dataset for one of two reasons: include more attack types or gain more reliability in previous approaches comparison and the number of iterations [4].

6. CONCLUSION

With the increase of the web applications significance, the demand for secure web applications increases. However, recent studies propose DL to detect web application attacks, whereas many of the previous studies achieve a high detection rate. This paper is to cover up the main aspects of web-based attack detection to build a sufficient background. Furthermore, the related studies are included to gain more information about the previous approaches, dataset, and performance metrics. In the end, there are no specific criteria to determine the best approach because it depends on the aim of choosing the approach.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Lujain Alghofaili	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Dina M. Ibrahim	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [Lujain Alghofaili], upon reasonable request.

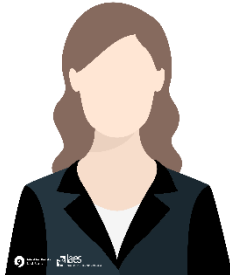
REFERENCES




- [1] A. Alzahrani, A. Alqazzaz, Y. Zhu, H. Fu, and N. Almashfi, "Web application security tools analysis," in *Proceedings - 3rd IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2017, 3rd IEEE International Conference on High Performance and Smart Computing, HPSC 2017 and 2nd IEEE International Conference on Intelligent Data and Security, IDS 2017*, May 2017, pp. 237–242, doi: 10.1109/BigDataSecurity.2017.47.
- [2] X. Li and Y. Xue, "A survey on web application security," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 6, no. 5, pp. 223–228, 2020.
- [3] M. Maithem and G. A. Al-sultany, "Network intrusion detection system using deep neural networks," *Journal of Physics: Conference Series*, vol. 1804, no. 1, p. 012138, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012138.
- [4] A. Tekerek, "A novel architecture for web-based attack detection using convolutional neural network," *Computers & Security*, vol. 100, p. 102096, Jan. 2021, doi: 10.1016/j.cose.2020.102096.
- [5] G. Kumar, K. Kumar, and M. Sachdeva, "The use of artificial intelligence based techniques for intrusion detection: a review," *Artificial Intelligence Review*, vol. 34, no. 4, pp. 369–387, Dec. 2010, doi: 10.1007/s10462-010-9179-5.
- [6] L. Hung-Jen, R. L. Chun-Hung, L. Ying-Chih, and T. Kuang-Yuan, "Intrusion detection system: a comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013, [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804512001944>.
- [7] X. Kai and H. Jiankun, *Handbook of information and communication security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [8] M. Neelam, "Aspects of AI," *Learning outcomes of classroom research*, pp. 250–256, 2022.
- [9] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.
- [10] E. Horvitz and D. Mulligan, "Data, privacy, and the greater good," *Science*, vol. 349, no. 6245, pp. 253–255, Jul. 2015, doi: 10.1126/science.aac4520.
- [11] W. Pitts and W. S. McCulloch, "How we know universals the perception of auditory and visual forms," *The Bulletin of Mathematical Biophysics*, vol. 9, no. 3, pp. 127–147, Sep. 1947, doi: 10.1007/BF02478291.
- [12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [13] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.
- [14] "OWASP Press Release." <https://owasp.org/www-project-top-ten/> (Last accessed: May 2024).
- [15] M. M. Hassan, M. A. Ali, T. Bhuiyan, M. H. Sharif, and S. Biswas, "Quantitative assessment on broken access control vulnerability in web applications," *International Conference on Cyber Security and Computer Science (ICONCS'18), Oct 18-20, 2018 Safranbolu, Turkey*, no. October, pp. 1–7, 2018.
- [16] PortSwigger, "Access control vulnerabilities and privilege escalation," *PortSwigger*, 2024, [Online]. Available: <https://portswigger.net/web-security/access-control#top>.
- [17] S. Alotaibi, K. Alharbi, B. Abaalkhail, and D. M. Ibrahim, "Sensitive data exposure: data forwarding and storage on cloud environment," *International journal of online and biomedical engineering*, vol. 17, no. 14, pp. 4–18, Dec. 2021, doi: 10.3991/IJOE.V17I14.27365.
- [18] "SQL-injection." PortSwigger, [Online]. Available: <https://portswigger.net/web-security/sql-injection> (Last accessed: June 2024).
- [19] "Cross-site Scripting." PortSwigger, [Online]. Available: <https://portswigger.net/web-security/cross-site-scripting> (Last accessed: June 2024).
- [20] S. Loureiro, "Security misconfigurations and how to prevent them," *Network Security*, vol. 2021, no. 5, pp. 13–16, May 2021, doi: 10.1016/S1353-4858(21)00053-2.
- [21] Y. Pang, X. Xue, and A. S. Namin, "Early identification of vulnerable software components via ensemble learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2017, pp. 476–481, doi: 10.1109/icmla.2016.0084.

- [22] S. K. Gupta, S. Vanjale, S. Rasal, and M. Vanjale, "Securing IoT devices in smart city environments," in *2020 International Conference on Emerging Smart Computing and Informatics, ESCI 2020*, Mar. 2020, pp. 119–123, doi: 10.1109/ESCI48226.2020.9167630.
- [23] M. M. Hassan *et al.*, "Broken authentication and session management vulnerability: a case study of web application," *International journal of simulation: systems, science & technology*, May 2018, doi: 10.5013/ijssst.a.19.02.06.
- [24] "Server Side Request Forgery (SSRF) in Depth." <https://www.geeksforgeeks.org/server-side-request-forgery-ssrf-in-depth/> (accessed Jun. 28, 2022).
- [25] "Buffer Overflow," *OWASP Press Release*. https://owasp.org/www-community/vulnerabilities/Buffer_Overflow (Last accessed: May 2024).
- [26] "Buffer Overflow," *TeachTarget*. <https://www.techtartget.com/searchsecurity/definition/buffer-overflow> (Last Accessed: June 2024).
- [27] B. Hall, "Countering web injection attacks: a proof of concept MSc project background report," University of Manchester UK.
- [28] A. Abbasi, "CRLF Injection," *OWASP Press Release*, 2010. https://owasp.org/www-community/vulnerabilities/CRLF_Injection.
- [29] "Web parameter tampering," *OWASP Press Release*. https://owasp.org/www-community/vulnerabilities/CRLF_Injection (Last accessed: May 2024).
- [30] J. Liang, W. Zhao, and W. Ye, "Anomaly-based web attack detection: a deep learning approach," in *ACM International Conference Proceeding Series*, Dec. 2017, pp. 80–85, doi: 10.1145/3171592.3171594.
- [31] X. Kuang *et al.*, "DeepWAF: detecting web attacks based on CNN and LSTM models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11983 LNCS, 2019, pp. 121–136.
- [32] K. N. Rao, K. V. Rao, and P. R. Prasad, "A hybrid intrusion detection system based on sparse autoencoder and deep neural network," *Computer Communications*, vol. 180, pp. 77–88, Dec. 2021, doi: 10.1016/j.comcom.2021.08.026.
- [33] X. Xiao *et al.*, "Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets," *Computers and Security*, vol. 108, p. 102372, Sep. 2021, doi: 10.1016/j.cose.2021.102372.
- [34] X. Jin, B. Cui, J. Yang, and Z. Cheng, "Payload-based web attack detection using deep neural network," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 12, 2018, pp. 482–488.
- [35] M. Zhang, B. Xu, S. Bai, S. Lu, and Z. Lin, "A deep learning method to detect web attacks using a specially designed CNN," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10638 LNCS, 2017, pp. 828–836.
- [36] X. Gong *et al.*, "Estimating web attack detection via model uncertainty from inaccurate annotation," in *Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019*, Jun. 2019, pp. 53–58, doi: 10.1109/CSCloud/EdgeCom.2019.00019.
- [37] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu, "Detection of SQL injection based on artificial neural network," *Knowledge-Based Systems*, vol. 190, p. 105528, Feb. 2020, doi: 10.1016/j.knsys.2020.105528.
- [38] L. Nan and G. Sheng, "Application of convolution optimization algorithm based on neural network in web attack test," *Journal of Physics: Conference Series*, vol. 1325, no. 1, p. 012001, Oct. 2019, doi: 10.1088/1742-6596/1325/1/012001.
- [39] X. Gong, J. Lu, Y. Wang, H. Qiu, R. He, and M. Qiu, "CECoR-Net: a character-level neural network model for web attack detection," in *Proceedings - 4th IEEE International Conference on Smart Cloud, SmartCloud 2019 and 3rd International Symposium on Reinforcement Learning, ISRL 2019*, Dec. 2019, pp. 98–103, doi: 10.1109/SmartCloud.2019.00027.
- [40] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," *Journal of Physics: Conference Series*, vol. 1192, no. 1, p. 012018, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012018.
- [41] X. Gong, J. Lu, Y. Zhou, H. Qiu, and R. He, "Model uncertainty based annotation error fixing for web attack detection," *Journal of Signal Processing Systems*, vol. 93, no. 2–3, pp. 187–199, Mar. 2021, doi: 10.1007/s11265-019-01494-1.
- [42] N. T. Van, T. N. Thinh, and L. T. Sach, "An anomaly-based network intrusion detection system using deep learning," in *Proceedings - 2017 International Conference on System Science and Engineering, ICSSE 2017*, Jul. 2017, pp. 210–214, doi: 10.1109/ICSSE.2017.8030867.
- [43] D. Atienza, A. Herrero, and E. Corchado, "Neural analysis of HTTP traffic for web attack detection," in *Advances in Intelligent Systems and Computing*, vol. 369, 2015, pp. 201–212.
- [44] X. D. Hoang, "Detecting common web attacks based on machine learning using web log," in *Lecture Notes in Networks and Systems*, vol. 178, 2021, pp. 311–318.
- [45] T. Liu, Y. Qi, L. Shi, and J. Yan, "Locate-then-Detect: real-time web attack detection via attention-based deep neural networks," in *IJCAI International Joint Conference on Artificial Intelligence*, Aug. 2019, vol. 2019-August, pp. 4725–4731, doi: 10.24963/ijcai.2019/656.
- [46] V. H. García, R. Monroy, and M. Quintana, "Web attack detection using ID3," in *IFIP Advances in Information and Communication Technology*, vol. 218, 2006, pp. 323–332.
- [47] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked auto-encoder," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2018*, Feb. 2018, vol. 2018-January, pp. 131–134, doi: 10.1109/CFIS.2018.8336654.
- [48] S. Pillai and D. A. Sharma, "Hybrid unsupervised web-attack detection and classification – a deep learning approach," *Computer Standards and Interfaces*, vol. 86, 2023, doi: 10.1016/j.csi.2023.103738.
- [49] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," *Applied Sciences (Switzerland)*, vol. 9, no. 20, p. 4396, Oct. 2019, doi: 10.3390/app9204396.
- [50] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in *Proceedings - IEEE Symposium on Security and Privacy*, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.
- [51] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers and Security*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.
- [52] M. D. Wilkinson *et al.*, "Comment: The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.
- [53] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *Software Networking*, vol. 2017, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.
- [54] "Impact - HTTP Dataset Csic 2010," 2012. https://impactcybertrust.org/dataset_view?idDataset=940 (accessed: June 2024).
- [55] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000*, 2000, vol. 2, pp. 130–144, doi: 10.1109/DISCEX.2000.821515.




- [56] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

BIOGRAPHIES OF AUTHORS



Lujain Alghofailia    is a committed scholar and researcher in computer science and cybersecurity. She studied computer science from 2014 to 2019 at Qassim University, where she graduated with a bachelor's degree. She pursued her education at Qassim University after completing her undergraduate studies, where she studied for a master's degree in cyber security from 2020 to 2023. As of November 2023, she has been contributing to the Computer Science Department of the Applied College at Imam Mohammad Ibn Saud University as a collaborating lecturer. Her job is to teach computer science students by drawing on her vast expertise and experience. Lujain's research interests focus on cybersecurity, with particular emphasis on social media security and attack prevention. She has published significant works in this area, including a paper titled "Security in social media: awareness of phishing attacks techniques and countermeasures," presented at the ICCIT 2022 conference and a review article titled "Cross site scripting attack review," published in The ISC journal. She can be contacted at email: Lujainlg@gmail.com.



Dina M. Ibrahim    is associate professor at Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia from September 2015 till now. In addition, she works as an assistant professor at Computers and Control Engineering Department, Faculty of Engineering, Tanta University, Egypt. She was born in the United Arab Emirates, her B.Sc., M.Sc., and Ph.D. degrees have taken from Computers and Control Engineering Department, Faculty of Engineering, Tanta University in 2002, 2008, and 2014, respectively. She works as consultant engineer, then a database administrator, and finally acts as a vice manager on Management Information Systems (MIS) Project, Tanta University, Egypt, from 2008 until 2014. Her research interests include networking, wireless communications, machine learning, and the internet of things. She has published more than 60 articles in various refereed international journals and conferences. She is serving as a reviewer in Wireless Network (WINE) the Journal of Mobile Communication, Computation, and Information since 2015 in Springer publisher and recently, from 2021, in the MDPI journals, IEEE ACCESS Journal, and International Journal of Supply and Operations Management (IJSOM). She also acts as a Co-Chair of the International Technical Committee for the Middle East Region of the ICCMIT conference since 2021. She can be contacted at email: d.hussein@qu.edu.sa or dina.mahmoud@f-eng.tanta.edu.eg.