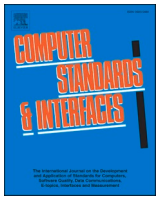
Computer Standards & Interfaces 86 (2023) 103738

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi



Hybrid unsupervised web-attack detection and classification – A deep learning approach

Seema Pillai [a,*], Dr. Anurag Sharma [b]

[a] MATS University, Aarang – Kharora, Highway, Arang, Chhattisgarh 493441
[b] Computer science engineering, MATS school of engineering, MATS University

ARTICLE INFO

Keywords:
SQL injection attack
Cross-site scripting attacks
Denoising autoencoder
Deep Boltzmann machine
Binary long term short term memory

ABSTRACT

Web requests made by users of web applications are manipulated by hackers to gain control of web servers. Moreover, detecting web attacks has been increasingly important in the distribution of information over the last few decades. Also, several existing techniques had been performed on detecting vulnerable web attacks using machine learning and deep learning techniques. However, there is a lack in achieving attack detection ratio owing to the utilization of supervised and semi-supervised learning approaches. Thus to overcome the aforementioned issues, this research proposes a hybrid unsupervised detection model a deep learning-based anomaly-based web attack detection. Whereas, the encoded outputs of De-Noising Autoencoder (DAE), as well as Stacked Autoencoder (SAE), are integrated and given to the Generative adversarial network (GAN) as input to improve the feature representation ability to detect the web attacks. Consequently, for classifying the type of attacks, a novel DBM-Bi LSTM-based classification model has been introduced. Which incorporates DBM for binary classification and Bi-LSTM for multi-class classification to classify the various attacks. Finally, the performance of the classifier in terms of recall, precision, F1-Score, and accuracy are evaluated and compared. The proposed method achieved high accuracy of 98%.

1. Introduction

Web applications have become the most popular Internet applications as the rate of Internet usage has increased. Web apps help companies to increase profitability and improve business operations, such as in the supply chain, by using virtualization or a business platform [1]. The Internet's stay connected characteristic has completely transformed business life [2–4]. It allows users to interact with each other where ever they desire. Many of the business activities which your company engages in regularly require the use of the internet. It is the repository of data and the home of cloud-based digital storage. It stores information that customers voluntarily provide through content management systems, shopping carts, login fields, and inquiry and submission forms. Each malicious attack on your website is different, and with so many various types of attacks occurring, it may appear impossible to defend itself against them all. Even so, there are several things you can take to protect your website against these attacks and reduce the chances that malicious hackers would target it. Due to their extensive availability and high usage, web apps are vulnerable to cyber-attacks.

Attackers attempt to make the Web application inaccessible by making a malicious request [5]. Intruders could corrupt a vulnerable Web application, affecting the confidentiality, integrity, and availability of the organization's resources [6–8]. As a result, organizations may suffer financial losses and irreversible harm. Web applications can be harmed in a variety of ways, including deleting specific resources, stealing data from databases, disrupting functioning, or gaining access to the application. Attacks can be classified into two types: targeted and untargeted. In untargeted operations, attackers attack as many devices, services, or users as possible without regard for their security. They do not even care who the victim is since there will be a large number of vulnerable computers or services. In a targeted attack, an organization is singled out because the attacker is interested in your market or has been paid to do so. It could take months to lay the framework for the attack so that they can figure out the best way to deliver their exploit directly to your computers (or users). Because it has been specially crafted to attack your systems, processes, or persons, a targeted attack is frequently more harmful than an untargeted attack. According to the Open Web Application, Security Project's (OWASP) Top 10 Security Vulnerabilities 2013 report, web application security threats include SQL injection [9], cross-site scripting (XSS) [10], server-side include [11], and broken

* Corresponding author
E-mail address: scholar.seemapillai213@gmail.com (S. Pillai).

https://doi.org/10.1016/j.csi.2023.103738
Received 5 February 2022; Received in revised form 22 December 2022; Accepted 25 February 2023
Available online 27 February 2023

authentication [12]. These programs, as universal and convenient as they are, are extremely vulnerable to web application attacks by cybercriminals. The most common methods of website security attacks

Phishing is one of the most harmful and hazardous criminal activities that is gaining popularity on the internet [13]. For the past several years, people who go online to utilize the services provided by the internet have been quickly caught up in phishing assaults [14]. To get the user's credentials, such as the user's login, password, card number, and other sensitive and private details, the attackers usually entice the victims to visit their faked websites [15]. These phishing assaults are profitable for the perpetrators. Typical targets for such assaults include e-banking, e-payment systems, and e-commerce apps [16].

Since it affects the core security services: confidentiality, authentication, authorization, and integrity, the Structured Query Language Injection (SQLI) attack is regarded as one of the most hazardous of the injection category [17]. To gain access to a database or change its data, SQLI attacks entail injecting (inserting) malignant SQL instructions into input forms or queries (e.g. sending the database contents to the attacker, modifying or deleting the database content, etc.) [18–19]. Inappropriate user input validation could contribute to the injection of an SQLI attack, which can have disastrous effects such as the database being deleted or sensitive and confidential data from web application customers being collected.

Cross-Site Scripting (XSS) vulnerabilities are one of the most prevalent high-risk online application cyber-attacks, putting users, web applications, and even the industrial industry in danger [20]. Because of these security flaws, hackers may be able to implant dangerous malware scripts into web pages that are presented to various end-users. As a result, it may be used to inflict harm, such as entirely altering the design or behavior of an organization's website, stealing critical enterprise or user information, acting on behalf of the actual user, and so on [21]. Various prevention and mitigation strategies have been suggested to address XSS-based vulnerabilities on the client-side, server-side, or pair-side, utilizing static, dynamic, or hybrid analysis methodologies [22]. Nevertheless, owing to the sophisticating and rising forms of XSS payloads [23], the suggested measures employing previous conventional approaches for such attack detection have become inadequate; moreover, most of them are not scalable over time and have an unavoidable situation of false positives [24]. Therefore, it is a greater necessity to develop a novel methodology to tackle such issues in the detection of web attacks.

Intrusion detection systems are configured with several signatures that support the detection of known attacks to detect web-based attacks. Unfortunately, with the large number of vulnerabilities discovered every day, it's hard to keep intrusion detection signature sets up to date. Custom web-based applications developed in-house may also introduce vulnerabilities. Developing ad hoc signatures to detect attacks on these applications is a time-consuming and error-prone procedure that requires a high level of security expertise.

The major issues concerning time complexity, low detection rate, high false-positive rate, and overfitting issues must be overcome. Hence a novel Web-Attack Detection and Classification using a deep learning approach to take down all the issues stated above is proposed.

- To detect such web-based attacks, a novel model has been introduced in this work named as hybrid unsupervised detection model. In which, DAE, as well as SAE, are integrated and given to the GAN as input to improve the feature representation.
- Consequently, for classifying the type of attacks, a novel DBM-BiLSTM-based classification model has been introduced. DBM is used for binary classification, and Bi-LSTM is used for multi-class classification, which leads to overfitting problems.

Moreover, the structure of the paper is organized as follows: The background of the research is discussed in Section 2. The framework for hybrid unsupervised detection models are developed in Section 3. Results and discussion are mentioned in section 4 and the experimental purpose, process, and evaluation are all described in Section 5. In the last section, we encapsulate our work.

2. Literature survey

Hossain et al. [25] investigate machine learning approaches and assess their effectiveness when applied to datasets containing characteristics that can distinguish between a Phishing Website and a legitimate one. This emphasizes the most effective method for detecting one of the most prevalent cyber-attacks, allowing for speedier detection and blacklisting of such sites, resulting in a safer and more secure online browsing experience for everyone. Random forest, support vector machine, and logistic regression approaches are utilized in the detection of a phishing attack.

Shahrivari et al. [26] evaluated the classifiers such as Logistic Regression, Decision Tree, Support Vector Machine (SVM), Ada Boost, Random Forest, Neural Networks, KNN, Gradient Boosting, and XGBoost. The AdaBoost algorithm comes with its own set of benefits and drawbacks. In low-noise datasets, AdaBoost is relatively resistant to overfitting. To increase model performance, AdaBoost simply has a few hyper parameters that need to be tweaked. Furthermore, this method is simple to comprehend and visualize. However, the effectiveness of AdaBoost for noisy data is debatable, with some claiming that it generalizes well, while others claiming that noisy data leads to poor performance since the algorithm spends too much time learning extreme instances, skewing findings.

Balogun et al. [27] developed a meta-learning approach based on a Functional Tree (FT) for identifying phishing websites. FT and its variations were used to study how to improve phishing website detection. Meta learners such as bagging, boosting and rotation forest are utilized as ensemble models to identify the phishing attacks.

Jemal et al. [28] provided an insight into the SQL injection attack as well as a categorization of the latest detection and prevention techniques. Classify the various assault sources, objectives, and kinds as well. Discuss and categorize the most significant and recent proposed solutions to mitigate this threat, particularly those based on ontology and machine learning.

Latchoumi et al. [29] describe the shortcomings in a few known approaches for dealing with SQL injection attacks and proposes an efficient hashing technique to counteract them. The machine learning idea using the SVM method was proposed to combat SQL injection threats. It's a tool for detecting and avoiding SQL injection. The SVM algorithm will be trained with all potential harmful phrases before being used to create the model in this method. When a user submits a new query, SVM is applied to the model to determine whether or not the query contains any harmful phrases.

S. Kascheev et al. [30] discussed the challenge of identifying XSS with machine learning approaches. XSS is a web server interaction attack in which malicious code is placed on a website to communicate with the attacker's server. Payment document numbers or the administrator session token may be stored in the page or HTTP Cookie, making it extremely susceptible. It was developed a model for identifying XSS assaults. The support vector technique, decision tree, Naive Bayes classifier, and Logistic Regression are among the machine learning methods taken into account.

Maseer et al. [31] designed a self-adaptive model for an intrusion detection system (IDS) to improve the detection of attacks. Consequently, a novel hybrid weighted deep belief network (HW-DBN) algorithm is proposed for building an efficient and reliable IDS model to detect cyberattacks. The HW-DBN algorithm integrates an improved Gaussian–Bernoulli restricted Boltzmann machine (Deep GB-RBM) feature learning operator with a weighted deep neural networks (WDNN) classifier. The CICIDS2017 dataset was selected to evaluate the Deep Internet of Things (IoT). However, development required to improve the overall performance of IoT network in real-time.

Gong et al. [32] provided the model uncertainty in the form as the variance of a Bayesian model. By training this attack detection model on real web logs with annotation errors, proved that the wrongly tagged web logs tended to gain a higher variance. Therefore, by analyzing the variance result, the security operators easily locate these mistagged web logs. This helps to find unknown attacks neglected by data annotation and to refine the existing attack detection methods. Further it should detect attacks in the layers other than web logs in the application layer.

Mhamdi et al. [33] proposed a hybrid unsupervised Deep learning approach using the stack autoencoder and One-class Support Vector Machine (SAE-1SVM) for Distributed Denial of Service (DDoS) attack detection. The SAE-1SVM shows that it reduces the processing time significantly while maintaining a high detection rate. In summary, the SAE-1SVM work well with imbalanced and unlabeled datasets. However, the system cannot detect all types of attack.

Xiang et al. [34] proposed a semi supervised learning detection model combining spectral clustering and random forest to detect the DDoS attack of the WEB application layer. This semi supervised learning model has a certain improvement in the detection rate while ensuring a low false positive rate and is more suitable for the detection of WEB DDoS attacks. However, the system cannot detect all types of attack.

Kumar et al. [35] proposed an innovative method called obfuscation to protect web applications from attacks like XSS and SQL injection. Due to numerous evaluation phases, web applications have evolved into nonlinear, interactive, and dynamic systems. Because of its widespread use and consequent rely on it, web application security is essential for ensuring the safety, excellence, and precision of web applications. However, the suggested method can be used only to protect data from debugging attack, not to detect all type of web attacks.

Vartouni et al. [36] proposed a technique that uses isolation forests as a classifier and deep neural networks to learn features. On the CSIC 2010 data set, this strategy was compared with others that do not employ feature extraction models. To protect servers from HTTP traffic, web application firewalls employ intrusion detection techniques. These firewalls also use machine learning algorithms based on anomaly detection. The findings demonstrate that deep models are more accurate than techniques without feature extraction. However, it cannot detect the various types of web attacks.

For [25] works needs to be further enriched in order to Correlation is encoded, not causality or fundamental relationships and entities or spatial relationships between entities must not be encoded, for [26] works needs to be further enriched in order to the effectiveness of Ada Boost for noisy data is debatable, with some claiming that it generalizes well, while others claiming that noisy data leads to poor performance, for [27] suggestions are made for its FT and its variations were used to study how to improve phishing website detection, for [28] it is recommended to include other variables, SQL injection attack as well as a categorization of the latest detection and prevention techniques, for [29] uncertainty constraints for crucial time periods should be added, for [30]corrective actions in training are required and for [20] solve more complex versions of the problem that include XSS is a web server interaction attack in which malicious code is placed on a website to communicate with the attacker's server. [31] IoT network needs to be developed in order to increase overall performance in real-time. For [32] it should detect attacks in the layers other than web logs in the application layer. For [33–34] the system cannot detect all types of web attacks. For [35] it protects the data from the debugging attack but it cannot detect the web attack. For [36] technique without feature extraction, the deep models are more accurate to detect web attacks but this technique did not detect all type of web attacks with extraction feature. As a result, it is essential to detect all possible web attack with an unsupervised learning technique to perform efficient feature extraction in the classification of web attacks in order to improve performance.

3. Unsupervised web attack detection framework

Unsecured web applications can be used by hackers to gain access to corporate servers. In which, hackers modify web requests issued by users of web applications to gain control of web servers. It is crucial to detect such vulnerable web-based attacks to prevent the privacy of data. Moreover, several existing techniques had been performed on detecting vulnerable web attacks using machine learning and deep learning techniques. However, there is a lack in achieving attack detection ratio owing to the utilization of supervised and semi-supervised learning approaches. Also, some research can have focused on the linear unsupervised learning approach, however, if the modeled relationship is nonlinear, it is impossible to represent the real underlying input and output relations. Similarly, while utilizing an artificial neural network for web attack detection, the backpropagation might result in an overfitting problem, which also provokes time complexity. Thus the above works suffered to reduce the false-positive rate.

Thus our framework proposed a model for vulnerable web attack detection. Moreover, to detect such web-based attacks, a novel hybrid unsupervised detection model has been introduced. Whereas, input is encoded with De-Noising Autoencoder (DAE), as well as Stacked Autoencoder (SAE), are combined and given to the Generative adversarial network (GAN) as input to improve the feature representation ability to detect the web attacks. Consequently, for classifying the type of attacks, a novel DBM-BiLSTM-based classification model has been introduced in which in DBM, it is distributed jointly and by increasing the log-likelihood of DBMs, the prediction is made simple, reducing information overfitting and computational complexity. It classifies the input as Malicious or Legitimate. In Bi-LSTM the predictions are made based on the context of components in the reality, tag each element's sequence using a finite sequence. Then, in Bi-LSTM, attack is classified into four types they are phishing attack, SQL injection attack, XSS attack and other attacks. As mentioned, to reduce the time complexity and to address the overfitting issues in the existing works, the Deep Boltzmann Machine (DBM) as well as bidirectional long short term memory (Bi-LSTM) models are employed for classifying the web attacks. Furthermore, the next section deals with a detailed step-wise explanation of the novel proposed method.

3.1. Hybrid unsupervised detection model

Web attacks are a serious threat to the environment because the increased use of cloud services, an increase in the number of web application users, and constantly evolving network technology all provide new cyber security challenges. Thus to detect such web-based attacks, a novel model has been introduced in this work named a hybrid unsupervised detection model. Instead of utilizing a linear unsupervised model, the non-linear unsupervised autoencoders are hybrid to improve the effectiveness of feature extraction from inputs. The input from CSIC2010v2 dataset URL are encoded with DAE as well as SAE, and then the encoded URLs are given to the GAN as input to improve the feature representation ability to detect the web attacks. This study presents a mathematical framework of non-linear autoencoders.

In DAE, initially, one input layer, one or more hidden layers, and one output layer are all included in autoencoders. In comparison to the input or output layers, autoencoders often have fewer units in their hidden layers. However, the size of an autoencoder's input and output layers is always the same. When encoding, an autoencoder takes a given input and, using its hidden layer, attempts to express it with fewer units than the input units (S). An autoencoder attempts to reconstruct the given input x using the encoded information in its hidden layer during decoding (S).

$$S(Wx+b)=y \tag{1}$$

$$S(Wy+b)=z \tag{2}$$

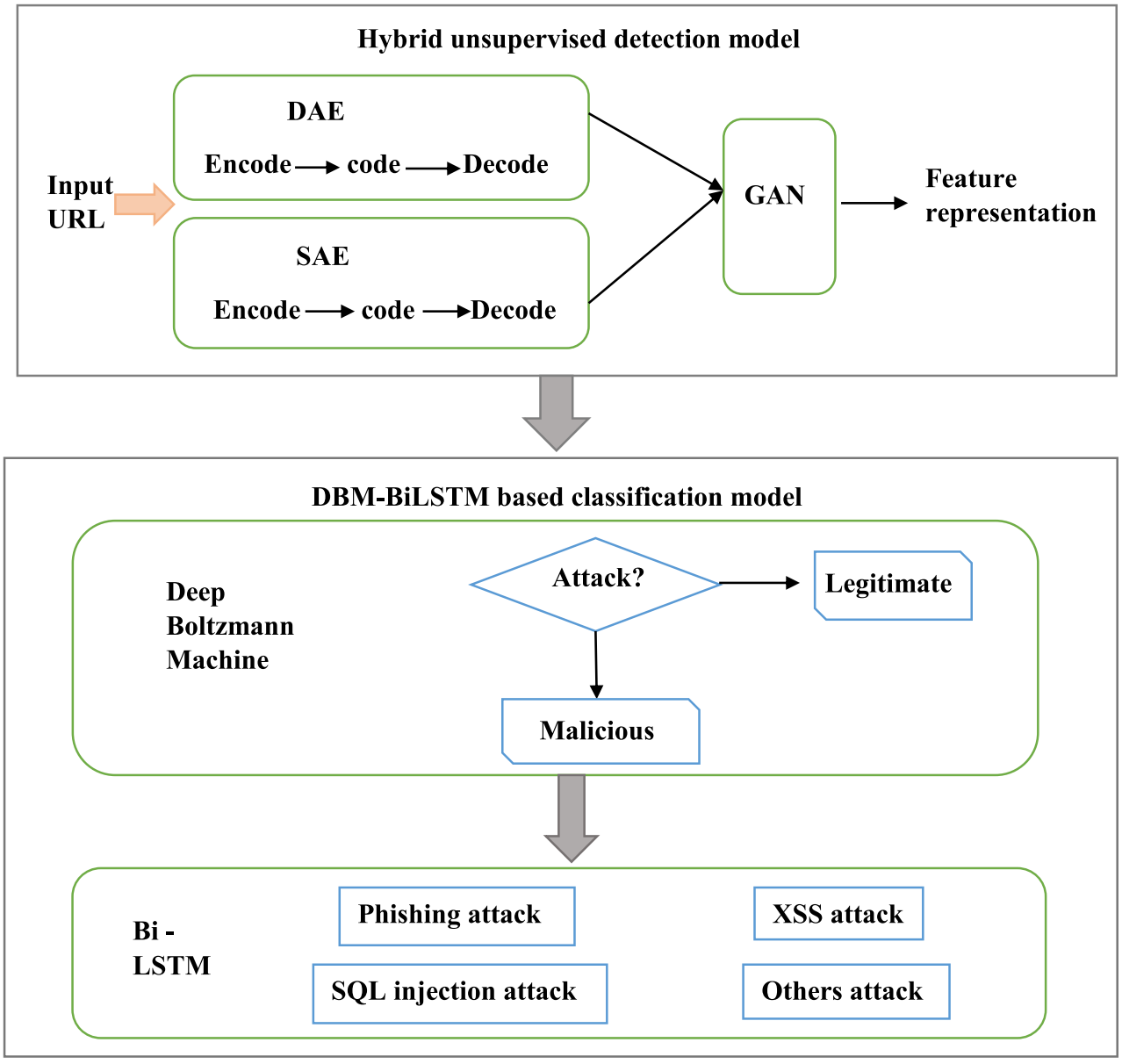


Fig. 1. Block diagram of the proposed framework.

$$SSE = \sum_{i=1}^{n} (xi - yi)^2 \quad (3)$$

Where

W unit weight of AE

b Bias values if AE

As shown in Equations (1) and (2), autoencoders perform deterministic mapping using encoding and decoding operations (2). The S function in Formula (1) denotes a non-linear activation function, which can be sigmoid depending on the problem's characteristics. Equation (3) represents the Sum of Square Error (SSE) which is the difference between observed value and the predicted value where,xi is one of the value in the sample and yi is the predicted value. The weight and bias values of an autoencoder unit are shown in the W and b variables, which are required for the encoding to generate the output y from the input x. Similarly, the W and b variables in Formula (2) indicate an AE unit's weight and bias values, which are required during decoding to generate the output z from the hidden value y.

Also, to detect abnormal HTTP queries, this work contains a pre-

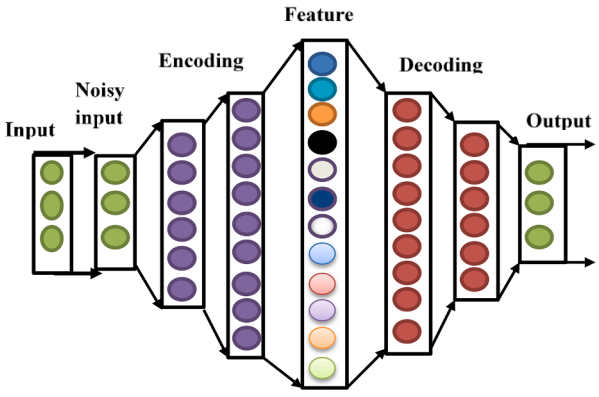


Fig. 2. Denoising Autoencoder.

processing of DAEs. Furthermore, these methods are adaptable to evolving web vulnerabilities, such as malicious code installed in different parts of the provocation and payload.

$$L\ (x, y)\ =\ ||xy||^2 \tag{4}$$

In which, DAE is used that is trained to reconstruct a clean input x from a corrupted version x. Specifically, x is mapped into a hidden representation $y = f(x)$, which is then decoded into input space using the $z = g(y)$decoding function.

Moreover, in stacked encoder deep learning, a set of approaches is utilized in this exploration. Whereas, an autoencoder neural network is an unsupervised learning technique that uses backpropagation to establish target values equal to inputs, such as $y(i) = x(i) for i = 1...n$.The encoder $h = f(x)$ and the reconstruction part $r = g(h)$ are always included in an autoencoder. The activation functions of neurons are $f(\cdot)$ and $g(\cdot)$, with g generally being linear. However, f can represent a variety of activation functions.

In which, we used an autoencoder to extract attributes from data that are complex abstract. Deep learning algorithms can be utilized in SAE mode to achieve a hierarchical learning process. Deep learning models can learn high-quality semantic features. In this segment, the feature is extracted from DAE and SAE which is the input for the generative adversarial network, it is explained in the following,

Thus, we propose using a GAN to generate more attack samples to increase the detection model's generalization ability. In which, a three-layer neural network with a data input layer, a hidden layer, and an output layer makes up the discriminator. Whereas, the encoded outputs of DAE, as well as SAE, are integrated and given to the GAN as input to improve the feature representation ability to detect the web attacks. Initially, the header parts of URL requests are eliminated first in the preprocessing because the header portions in the URL requests dataset are often the same, they reduce the size of the data by removing unnecessary data and eliminating data duplication in datasets. The URL and payloads are segmented into a sequence of words during pre-processing. The URL and payloads are picked and divided into variables and values (variable=value) by a specific variable and values filter that includes special characters such as "/," "&," "+," and so on. Using the Bag-of-Words (BoW) technique, the number of identical payloads in different HTTP requests is calculated as frequency. The BoW is a deep learning modeling method for extracting features from the text that is simple and flexible. Then, the malicious keywords are loaded into GAN network, and the risk value of each URL is calculated.

GANs are generative model which utilize the idea of confrontation which helps in interaction during the training and identify the risk value of URL. GANs model architecture is used for training a generative model and discriminator models. The proposed GAN model architecture is made up of two sub-models: a generator model for creating new instances and a discriminator model for determining whether the generated samples are real or fake. The generator in the standard generative confrontation network model is used to capture the probability distribution of real data samples, and the discriminator judge whether the input samples are real data samples or generated samples, which is essentially equivalent to a binary classification model, which is inspired by game theory's zero-sum game theory. The objective function is given by

$$V(D, G) =\ E_{x-Pdata(x)}\left[\log D(x)\right] + E_{z-P(z)}\left[\log(1 - D(G(z))\right] \tag{5}$$

Where G is the generator, D is the discriminator, $Pdata(x)$ is the distribution of real data, $P(z)$ is the distribution of the generator, x is the sample from $Pdata(x)$, $z$is the sample from $P(z)$, $D(x)$ is the discriminator network and $G(z)$ is the generator network.

The actual data in the training set and the created data are mixed and sent to the discriminator to discriminate between the genuine and generated samples throughout the training process. The generator resembles real data as closely as possible so that the discriminator accurately identifies the data. Throughout the process, there is constant confrontation, which aids in achieving Nash balancing. In the view of confrontation and interaction, the unlabeled data are converted into labeled data. From the formula, it is observed that the discriminator has to train and evolve the discrimination accuracy continuously between real samples and generated samples such that the $D(x)$ is maximized and $D(G(z))$ is minimized. Also, there is a need to train the generator so that the model generates samples similar to the real data so as to minimize $D(x)$ and maximize $D(G(z))$. The GAN structure of proposed method is shown in Fig. 3.

Then fake data instances are created by the Generator. The Discriminator classifies both real data and fake data from the generator. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real. The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network. Finally, the feature map is generated by using the generative adversarial network, and generated feature map is classified in the forthcoming section.

3.2. Classification of attacks by using a deep learning model

The feature extracted from the hybrid unsupervised detection model is an input for the deep Boltzmann machine bidirectional long short-term memory. Consequently, for classifying the type of attacks, a novel DBM-BiLSTM-based classification model has been introduced. As mentioned, to reduce the time complexity and to address the overfitting issues in the existing works, the DBM as well as Bi-LSTM models are employed for classifying the web attacks. As can learn complex feature representation, DBM has been utilized for binary classification to classify the input as legitimate or malicious. Subsequently, the Bi-LSTM can be employed for multi-class classification to classify the type of attack with addressing the generalization issue through the ability of long pattern learning.

Firstly, the URL's unstructured information (e.g., textual description) is processed and converted to a numerical vector in this phase so that it may be fed into the pre-training method. The unsupervised filters were used to remove noisy attribute values by replacing them with zero, and missing data were filled with the mean and mode values of the attributes from the numerical vectors. In addition, Z-score normalization is applied to the numerical vector to normalize values in the 0 to 1 range. For the normalization process, the following equation is used:

$$Z\ = \frac{x - \mu}{\sigma} \tag{6}$$

Where x is the numeric value for each sample of the feature, μ is the feature's mean, and σ is the feature's standard deviation. The absolute value of z represents the distance in standard deviations between the numeric value and the feature's mean. When the numeric value is below the mean, z is negative is indicated as an attack; when it is above, it is positive is indicated as normal. If the indicated as an attack, then it classified using the DBM and Bi-directional LSTM to identify the type of attack.

3.2.1. Deep boltzmann machine with Bi-LSTM

The Deep Boltzmann Machine is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $v \in \{0, 1\}^D$, and a sequence of layers of hidden units $h^1 \in \{0, 1\}^{F_1}$, $h^2 \in \{0, 1\}^{F_2}, ...,$ $h^L \in \{0, 1\}^{F_L}$. There are connections only between hidden units in adjacent layers, as well as between the visible units and the hidden units in the first hidden layer. The proposed method has 3 layers then the value of L is 3. The energy of the state $\{v, h\}$ is defined as:

$$\mathrm{E}(\mathrm{v},\ \mathrm{h};\ \theta)\ =\ -\mathrm{v}^T W^1 h^1 -\ h^{1T} W^2 h^2\ -\ h^{2T} W^3 h^3 \tag{7}$$

In equation (7), $h = \{h^1, h^2, h^3\}$ are the set of hidden units, $\theta = \{W^1, W^2, W^3\}$are the model parameters, representing visible-to-hidden and hidden-to-hidden symmetric interaction terms. All units in DBM are

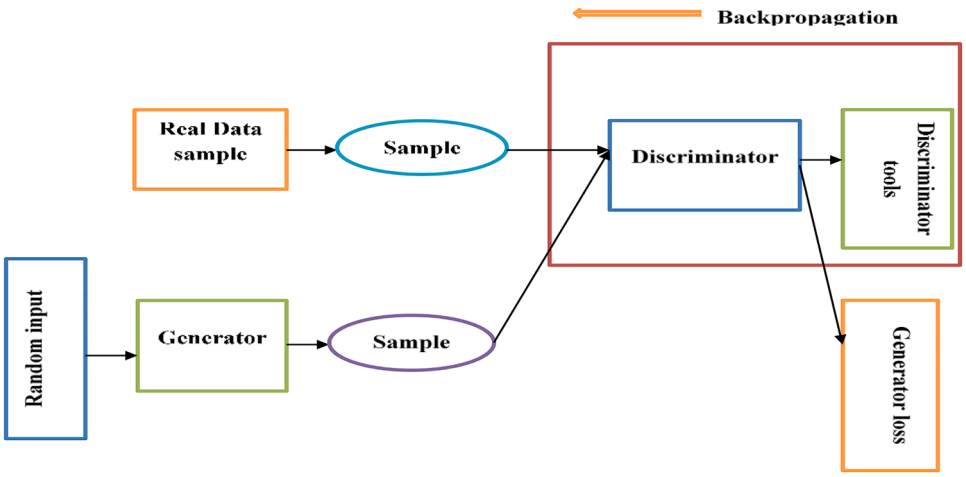


Fig. 3. Structure of GAN.

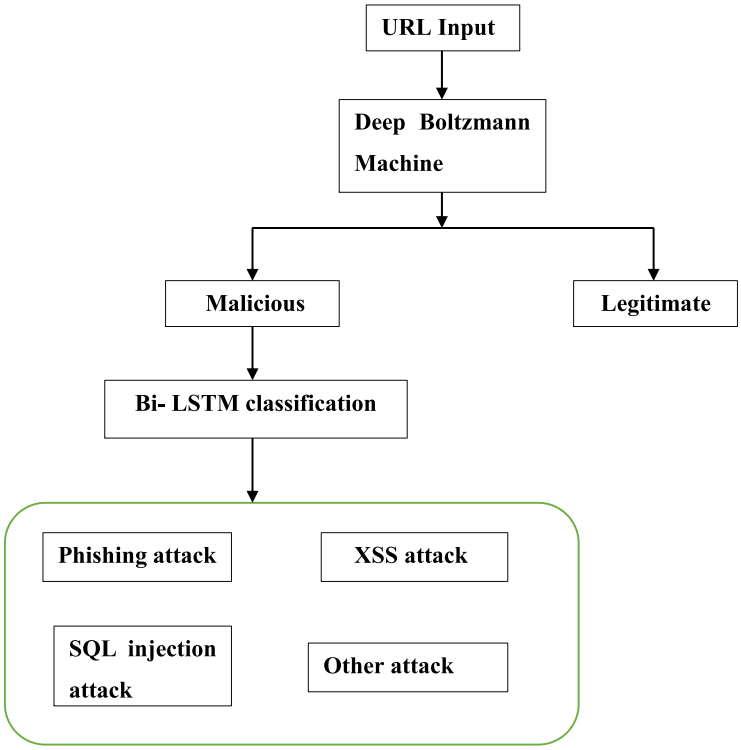


Fig. 4. Classification of web attacks using the DBM-Bi-LSTM model.

distributed jointly according to the following formula equation (8):

$$S(v,h|\theta) = \frac{1}{N(\theta,\alpha)} e^{-\alpha E(v,h;\theta)} \tag{8}$$

In which normal distribution is defined by $N(\theta,\alpha) = \sum_{v,h} e^{-\alpha E(v,h;\theta)}$. Where $\sum_{v,h} is$ the sum over all visible and hidden units. This is often accomplished by maximizing the log-likelihood of DBMs expressed by equation (9)

$$L(\theta) = \frac{1}{S}\sum_{\beta=1}^{S} \sum_{(v,h)\backslash(v,h)^{(0)}} S\Big((v,h)^{\beta,(0)},\ ((v,h)\backslash(v,h)^{(0)}|\theta)\Big) \tag{9}$$

The log-likelihood differences with respect to model parameters is given in equation (10)

$$\frac{L(\theta)}{W} = \frac{1}{s}\sum_{\beta=1}^{S} \sum_{(v,h)\backslash(v,h)^{(0)}} v,h\ S((v,h)\backslash(v,h)^{(0)}\Big|\ (v,h)^{\beta,(0)},\theta\Big) - \sum_{v,h}(v,h)S((v,h)|\theta) \tag{10}$$

From equations (8), (9), and (10), it is noted that the units of DBM are jointly distributed and the prediction is made easy by maximizing the log-likelihood of DBMs thereby reduces the overfitting of information and computational complexity. DBM and Bi-LSTM were utilized to perform both binary as well as multiclass classification of attacks.

Then the Bi-directional LSTM predictions tag the sequence of each element using a finite sequence based on the context of elements in the past and future. This is the result of two LSTMs running in parallel, one processing from left to right and the other from right to left. The forecast of a given target signal is known as composite output. This method is

Algorithm 1
Pre-training of DBM with Bi-LSTM.

Step 1: Make two copies of the visible vector and tie the visible-to- hidden weights W^1. FitW^1 of the 1st Bi-LSTM layer data.
Step 2: Freeze W^1 that defines the 1st layer of features, and use samplesh^1 from$P(h^1|v,2W^1)$) as the data for training the next layer DBM with weight vector $2W^2$.
Step 3: Freeze W^2 that defines the 2nd layer of features and use the samples h2 from $P(h^2|h^1,2W^2)$ as the data for training the 3rd layer DBM with weight vector $2W^3$
Step 4: When learning the top-level DBM, double the number of hidden units in the Bi-LSTM and tie the visible-to-hidden weights W^3.
Step5: Use the weights $\{W^1,W^2,W^3\}$ to compose a Deep Boltzmann Machine.

effective. When using a typical LSTM network to predict time series, the prediction effect is lost due to the failure to learn all of the sequences and the disregard of future context information.

Then, the data preparation module is in charge of extracting events (URL request packet payload) as well as tokenizing and encoding the payload data extracted. The raw URL request packet is a single URL stream that is requested by a user or client to obtain Web server services. We utilize it to process (extract) the payload of URL request packets. Payloads are extracted by processing the raw URL request packet, which is the resource's true representation. We only need the payload part of an URL request packet for our proposed Web attack detection system because an URL request packet contains numerous components. On the extracted URL request packet payload data, tokenization is implemented. Using the "&" symbol, which separates a series of parameters with their values, it will tokenize into a sequence of tokens or pieces. A unique number or mapping of tokens to integers is used to represent payload data tokens. Because machine learning models use arrays of numbers as input, we encode to process (train and test) the suggested system model.

Also, XSS is a type of security flaw that is commonly found in web applications. Attackers can utilize XSS to inject client-side scripts into web pages that are being viewed by other users. An attacker could exploit a cross-site scripting vulnerability to get around access controls like the same policy. The trained model is the end consequence of using the learning methods to train the tokens payload feature dataset. In other words, it's a representation of the HTTP request packet payload data's learned attack and benign behavior. The detection module is in charge of putting the trained model to the test by separating new (untrained) tokens payload test data and classifying them as malicious or benign based on their learned behavioral characteristics. Untrained tokens payload data will be provided to the trained model to test them.

Before using binary classifier Bi-LSTM to detect and categorize phishing, XSS, and SQL attack the training sample (80%) and testing sample (20%) are separated from the pre-trained feature values. Furthermore, training the suggested model with both normal and attack payload data to understand and extract their patterns helps to reduce false positives and negatives. As a result, the proposed work can efficiently extract the features and classify the types of web attacks.

Table 1
Features of SQL attack.

Features	Value	Calculation method	Descriptions
LP	Numeric	Len(payload)	Length of the payload in URL
NK	Numeric	$\sum_{k\in keywords} payload(k)$	Number of keywords in the payload
KWS	Numeric	$\sum_{k\in keywords} weight(k) * payload(k)$	Sum of keywords weights in the payload
NSPA	Numeric	Payload[space] + payload [%"20"]	Number of spaces in payload
RSPA	Float	NSPA\LP	Ratio of spaces
NSPE	Numeric	$\sum_{s\in special\ characters} payload(s)$	Number of special characters
RSPE	Float	NSPE\LP	The ratio of special character

Table 2
Features of Phishing attack.

Features	Value	Descriptions
Num Dots	Numeric	The number of dots In the URL.
Subdomain Level	Numeric	Determines the number of subdomain levels.
Path Level	Numeric	Determining the level of the path in the URL.
Url Length	Numeric	Length of each URL used in the dataset. The length contains the number of letters or symbols used to create the URL.
Num Dash	Numeric	A total number of the dash in a URL.
Num Dash In Hostname	Numeric	The number of dashes in a hostname

4. Experimental design

4.1. Experimental setup

This work has been implemented and the simulation of the system was then done in the python platform with the OS of Windows 8 and Intel core i5 processor as well as the capacity of the RAM is about 8GB.

4.2. Dataset description

The proposed model is tested and evaluated using the CSIC2010v2 dataset in this study. The (CSIC) 2010v2 dataset, developed by the Spanish Research National Council (CSIC) in 2010 at the Information Security Institute is one of the most well-known and widely used datasets in the field of Web security. When compared to older datasets such as KDD99 and DRAPA, the CSIC2010v2 specializes in Web attack detection with 104,000 normal and 119,585 malicious requests created on the shopping cart application of an e-commerce Web site. SQL injection, buffer overflow, information gathering, CRLF injection, XSS, and parameter tampering are among the anomaly requests. Moreover, in this study, the dataset is generated automatically and contains over 25,000 anomalous requests in addition to 36,000 normal requests. The URL requests are classified as normal or abnormal, and attacks are included in the dataset.

4.3. Evaluation metrics

The simulation of the proposed hybrid unsupervised web-attack detection and classification is evaluated with the metrics such as accuracy, precision, F1-score, recall, and false positive rate.

4.3.1. Accuracy

The accuracy of the clinical text data is calculated using,

$$\text{Accuracy} = \left[\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\right]* \tag{11}$$

Where, TP is the True Positive Value, TN is the True Negative Value, FP is the False Positive Value and FN is the False Negative Value respectively.

4.3.2. Precision

The closeness of two or more measurements to each other is known as precision. The formula is presented as,

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{12}$$

Where TP is the True Positive and FP is the False Positive value.

4.3.3. Recall

A recall is defined as the ability of the model to accurately predict the output. The formula of recall is defined as,

$$Recall = \frac{TP}{TP\ +FN} \tag{13}$$

Where TP and FN are the True positive and False negative value respectively.

4.3.4. F1 score

F1 Score is defined as,

$$F1 = \frac{2 \times (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \tag{14}$$

4.3.5. False positive rate

The false-positive rate is calculated as,

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

where FP represents the number of false positives and TN determines the total of true negatives (FP+TN = total number of negatives). It's the probability of a false alarm being stimulated: a positive result being given while the true value is negative. The false-positive rate of the proposed system is 0.005.

5. Result and discussion

This section describes various simulation outputs and the proposed model's performance analysis, as well as a comparison section to ensure that the proposed system is enhanced.

5.1. Simulation outputs

Fig. 5 represents the DBM has been utilized for binary classification to classify the input as legitimate or malicious. The mean value for the collected sample is 0.410464 and its standard deviation is 0.491922. A total of 61,065 samples is collected. As a prediction, the samples are split where anomalous samples are 25,065 and normal samples are 36,000. Only 17 feature samples were extracted from 61,065 samples. Also, the following simulation results show malicious and normal attacks.

Using a deep Boltzmann machine, the collected samples demonstrate that they are all malevolent and normal attacks as shown in Fig. 6. Moreover, our proposed Bi-LSTM can be employed for multi-class classification to classify the type of attack with addressing the generalization issue through the ability of long pattern learning. In, the total number of phishing attacks in the collected sample is 1811, SQL injection attack in the collected sample is 9930, XSS attack in the collected sample is 2507, and another attack in the collected data sample is 10,817. Our proposed DBM-Bi LSTM shows that accuracy as 97.6%, precision as 98.78%, recall as 98.78%, F1-score as 98.78% and also FPR as 98.78%. Thus our proposed work reduces the false positive rate and overfitting issues.

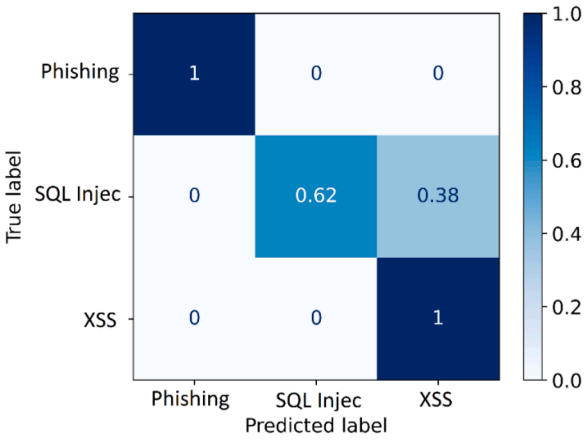


Fig. 6. Confusion matrix for Multi-class classification of attacks.

From Fig. 7, to detect the web attack training and testing sample shows that the reduction in overfitting issues. When comparing the training data and testing data performance in each epoch training data loss gradually decreases. For the epoch range of 0 to 20 training data occurs the loss level from 0.15 to 0.13 and the testing data occurs the loss level from 0.17 to 0.18. For the epoch range of 80 to 100, the training data occurs the loss level from 0.11 and the testing data occurs the loss level of 0.19. Hence, the proposed technique accurately detects the types of web attack.

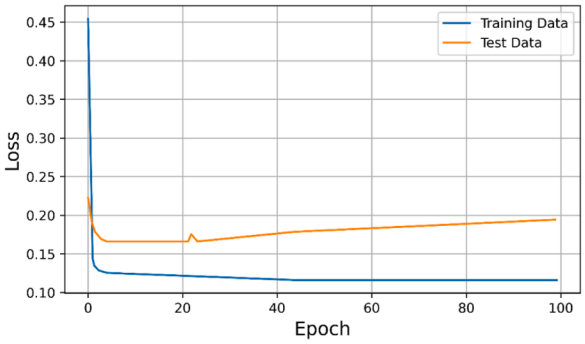


Fig. 7. Training and testing sample to detect web attack.

```
mean      0.410464
std       0.491922
25%       0.000000
50%       0.000000
75%       1.000000
max       1.000000
(61065,   17)
Prediction
Anomalous 25065
Normal        36000
dtype:    int64
Total Number of Samples: 25065
Total Number of Features/Sample: 17
2021-08-17  14:05:55.529639:  I
tensorflow/stream_executor/platform/default/dso_loader
```

Fig. 5. Binary Classification of attacks.

5.1.1. Comparison of overall performance metrics of proposed system

The Fig. 8 shows that, the overall performance metrics of the proposed system such as accuracy, precision, recall, and F1 score.

This section describes the resultant performance of the proposed system. The next section describes a comparison of the various performance of the previous research with the performance of the proposed method.

5.2. Comparison metrics

This section describes the various performance of the proposed method compared with the results of previous methodologies and depicts their results based on various metrics.

The accuracy of data is compared with the accuracy of the various previously proposed techniques. From graph 9, it is clear that the stack accuracy of the proposed output achieves 97% which is 3% higher than the existing output when compared with baseline, Convolutional Neural Network (CNN), K-means, K-Nearest Neighbors (KNN), Principal Components Analysis (PCA), and Angle Based Outlier Detection (ABOD).

The precision of data is compared with the precision of the various previously proposed techniques. From graph 10, it is clear that the stack precision of the proposed output achieves 99.6% which is 2% higher than the existing output when compared with baseline CNN, K-means, KNN, PCA, and ABOD.

The recalls of data are compared with the recalls of the various previously proposed techniques. From graph 11, it is clear that the stack recalls of the proposed output achieve 98.8% which is 3% higher than the existing output when compared with baseline, CNN, K-means, KNN, PCA, and ABOD.

The F1-Score of data is compared with the F1-Score of the various previously proposed techniques. From graph 12, it is clear that the stack F1-Score of the proposed output achieves 98.6% which is 5% higher than the existing output when compared with baseline, CNN, K-means, KNN, PCA, and ABOD.

The FPR of data is compared with the FPR of the various previously proposed techniques. From graph 13, it is clear that the stack FPR of the proposed output achieves 20% which is 10 % lower than the existing output when compared with CNN, K-means, KNN, PCA, and ABOD.

In Fig. 14, the proposed system time complexity is compared with existing methods such as RWAF, HMM-web, and LTD. Firstly, when the proposed system is compared to the existing technique of RWAF method, RWAF method obtained more than 250m/s to detect the web attack which is high time consumption but the proposal method need nearly 5m/s to detect the web attack which is low time consumption. Also, the existing technique like HMM-web and LTD need more than 30m/s, but the proposal system obtain the web attack at nearly 5m/s. Hence, when the proposed system is compared to the existing technique it takes less time consumption to detect the web attack.

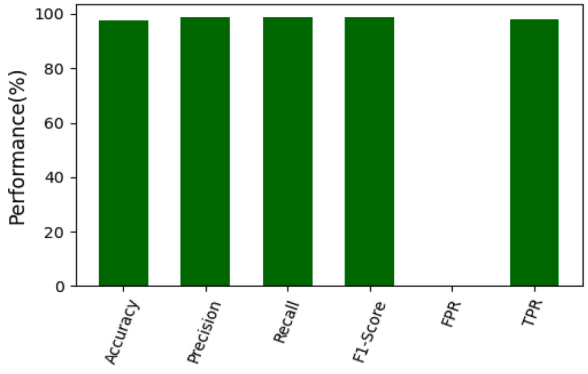


Fig. 8. Overall performance metrics.

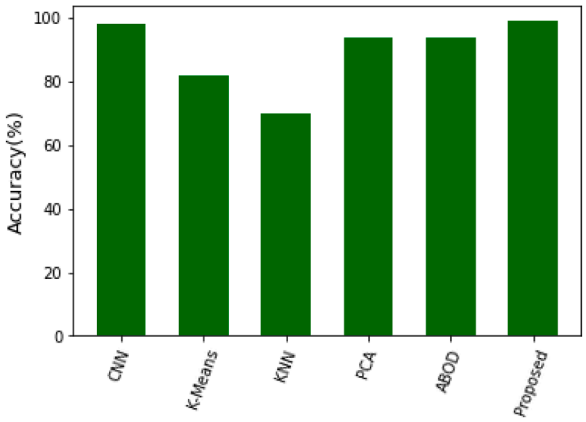


Fig. 9. Accuracy Comparison.

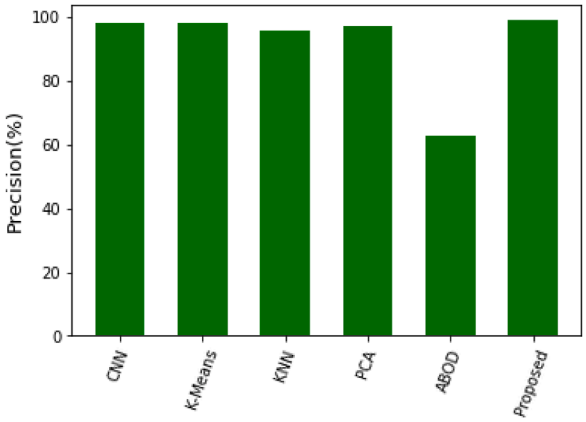


Fig. 10. Precision Comparison.

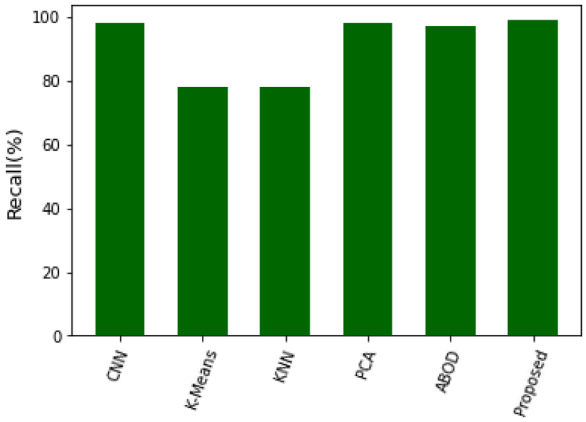


Fig. 11. Recall Comparison.

6. Conclusion

Web applications have drawn increased attention over the years from companies, organizations, and social media, making them a prime target for cyber-attacks. Thus to detect and classify the web-based attacks, a novel model has been introduced which incorporates a hybrid unsupervised detection model and a novel DBM-BiLSTM-based classification model. The proposed method is extracted with high accuracy, F1-Score, recall, precision, false-positive rate of 97.62 %, 98.78%, 98.78%, 98.78%, and 98.78% respectively. The findings demonstrate that when the proposed method is compared to other existing techniques, it

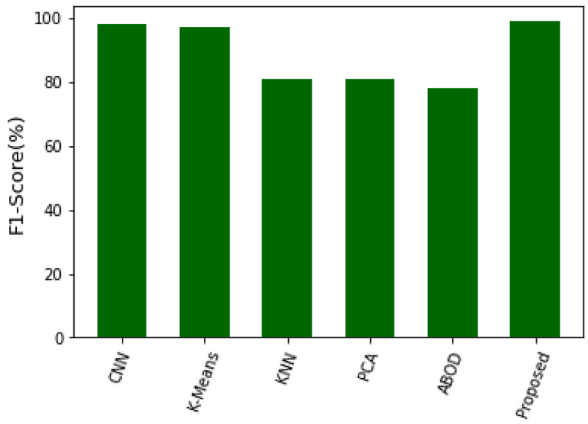


Fig. 12. F1-Score Comparison.

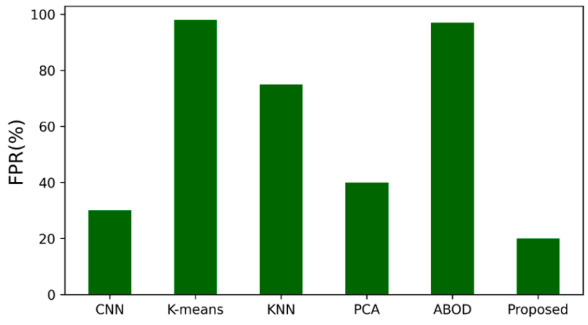


Fig. 13. False Positive Rate comparison.

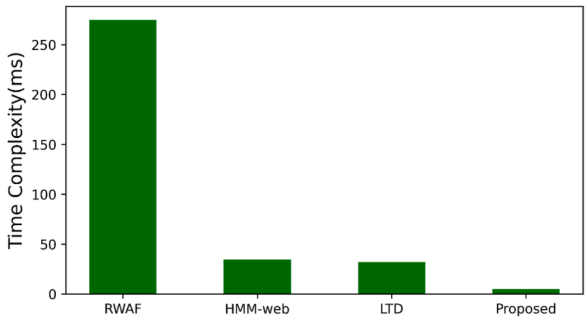


Fig. 14. Time Complexity comparison.

performs better than any of them. The proposed architecture was found to be fully comprehensive based on the metric results. Thus, the proposed architecture can be utilized to protect web applications from web-based attacks, according to the results. However, in future some improvements are needed to detect all types of web attacks with high accuracy, by incorporating optimization based approaches.

Author statement

None.

Declaration of Competing Interest

None.

Data availability

No data was used for the research described in the article.

References

[1] N.Cor Verdouw, et al., Virtualization of food supply chains with the internet of things, J. Food Eng. 176 (2016) 128–136.
[2] Venkatesh Shankar, et al., Mobile marketing in the retailing environment: current insights and future research avenues, J. Interactive Marketing 24 (2) (2010) 111–120.
[3] Eleonora Pantano, Ubiquitous retailing innovative scenario: from the fixed point of sale to the flexible ubiquitous store, J. Technol. Manage. Innov. 8 (2) (2013) 84–92.
[4] Eleonora Pantano, Constantinos-Vasilios Priporas, The effect of mobile retailing on consumers' purchasing experiences: a dynamic perspective, Comput. Hum. Behav. 61 (2016) 548–555.
[5] G.Todd Booth, Karl Andersson, Elimination of dos UDP reflection amplification bandwidth attacks, protecting TCP services, in: International Conference on Future Network Systems and Security, Springer, Cham, 2015.
[6] Latifa Ben Arfa Rabai, et al., A cybersecurity model in cloud computing environments, J. King Saud Univ.-Comput. Inf. Sci. 25 (1) (2013) 63–75.
[7] Irshad Ahmed Sumra, Halabi Bin Hasbullah, Jamalul-lail Bin AbManan, Attacks on Security Goals (confidentiality, integrity, availability) in VANET: a survey, Vehicular Ad-Hoc Networks for Smart Cities, Springer, Singapore, 2015, pp. 51–61.
[8] Yulia Cherdantseva, et al., A review of cyber security risk assessment methods for SCADA systems, Computers & security 56 (2016) 1–27.
[9] G.William Halfond, Jeremy Viegas, Alessandro Orso, A classification of SQL-injection attacks and countermeasures, in: Proceedings of the IEEE international symposium on secure software engineering 1, 2006.
[10] Rahul Johari, Pankaj Sharma, A survey on web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection, in: 2012 International Conference on Communication Systems and Network Technologies, IEEE, 2012.
[11] Puspendra Kumar, R.K. Pateriya, A survey on SQL injection attacks, detection and prevention techniques, in: 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), IEEE, 2012.
[12] Md Maruf Hassan, et al., Broken authentication and session management vulnerability: a case study of web application, Int. J. Simul. Syst., Sci. Technol. 19 (2) (2018) 6–11.
[13] Avisha Das, et al., SoK: a comprehensive reexamination of phishing research from the security perspective, IEEE Commun. Surv. Tutorials 22 (1) (2019) 671–708.
[14] Jiajing Wu, et al., Who are the phishers? phishing scam detection on ethereum via network embedding, IEEE Trans. Syst., Man, Cybern.: Systems (2020).
[15] Sen Chen, et al., Gui-squatting attack: automated generation of android phishing apps, IEEE Trans. Dependable Secure Comput. (2019).
[16] Qi Li, Mingyu Cheng, Junfeng Wang, Bowen Sun, LSTM based phishing detection for big email data, IEEE Transactions on Big Data (2020).
[17] G. Deepa, et al., Black-box detection of XQuery injection and parameter tampering vulnerabilities in web applications, Int. J. Inf. Secur. 17 (1) (2018) 105–120.
[18] Yong Fang, et al., WOVSQLI: detection of SQL injection behaviors using word vector and LSTM, in: Proceedings of the 2nd international conference on cryptography, security and privacy, 2018.
[19] Qi Li, et al., LSTM-based SQL injection detection method for intelligent transportation system, IEEE Trans. Veh. Technol. 68 (5) (2019) 4182–4191.
[20] O. Andreeva, S. Gordeychik, G. Gritsai, O. Kochetova, E. Potseluevskaya, S. I. Sidorov, A.A. Timorin, Report, Kaspersky Lab, 2016.
[21] V. Nithya, S. Lakshmana Pandian, C. Malarvizhi, A survey on detection and prevention of cross-site scripting attack, Int. J. Security Applications 9 (3) (2015) 139–152.
[22] Upasana Sarmah, D.K. Bhattacharyya, K. Jugal Kalita, A survey of detection methods for XSS attacks, J. Netw. Comput. Appl. 118 (2018) 113–143.
[23] Yun Zhou, Peichao Wang, An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence, Comput. Security 82 (2019) 261–269.
[24] G.E. Rodríguez, J.G. Torres, P. Flores, D.E. Benavides, Cross-site scripting (XSS) attacks and mitigation: a survey, Computer Networks 166 (2020), 106960.
[25] Sohrab Hossain, Dhiman Sarma, Rana Joyti Chakma, Machine Learning-Based Phishing Attack Detection, Machine Learning 11 (9) (2020).
[26] Vahid Shahrivari, Mohammad Mahdi Darabi, Mohammad Izadi, Phishing Detection Using Machine Learning Techniques, arXiv preprint, 2020. arXiv:2009.11116.
[27] O.Abdullateef Balogun, et al., Improving the phishing website detection using empirical analysis of Function Tree and its variants, Heliyon (2021) e07437.
[28] Ines Jemal, et al., Sql injection attack detection and prevention techniques using machine learning, Int. J. Appl. Eng. Res. 15 (6) (2020) 569–580.
[29] T.P. Latchoumi, ManojSahit Reddy, K. Balamurugan, Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention, Eur. J. Mol. Clinical Medicine 7 (2) (2020) 2020.
[30] S. Kascheev, T. Olenchikova, The Detecting Cross-Site Scripting (XSS) Using Machine Learning Methods, in: 2020 Global Smart Industry Conference (GloSIC), 2020, pp. 265–270, https://doi.org/10.1109/GloSIC50886.2020.9267866.
[31] K.Ziadoon Maseer, et al., DeepIoT. IDS: hybrid deep learning for enhancing IoT network intrusion detection, CMC-Comput., Mater. Continua 69 (3) (2021) 3945–3966.
[32] Xinyu Gong, et al., Estimating web attack detection via model uncertainty from inaccurate annotation, in: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), IEEE, 2019.
[33] Lotfi Mhamdi, et al., A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs, in: 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), IEEE, 2020.

[34] L. Zhou, Y. Zhu, T. Zong, Y. Xiang, A feature selection-based method for DDoS attack flow classification, Future Gen. Comput. Syst. 132 (2022) 67–79.
[35] D. Kumar, Enhance Web Application Security Using Obfuscation, Turk. J. Comput. Mathematics Educ. (TURCOMAT) 12 (12) (2021) 1984–1989.
[36] A.M. Vartouni, S.S. Kashi, 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), IEEE, 2018, pp. 131–134. February.