Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Analysis of the False Prediction of the Logistic Regression Algorithm in SQL Payload Classification and its Impact on the Principles of Information Security (CIA)

Rehab Flaih Hasan[1], Omar Salah F. Shareef [2]*, Ammar Hatem Farhan[2]

[1]University of Technology, Department of Computer Sciences, Baghdad, Postcode, IRAQ
[2]University of Fallujah, Computer Centre, Fallujah, IRAQ

*Corresponding Author: Omar Salah F. Shareef

DOI: https://doi.org/10.52866/ ijcsm.2023.04.04.015
Receive July 2023; Accepted October 2023; Available online November2023

ABSTRACT: Securing sensitive data is of utmost importance for organizations and individuals to maintain data confidentiality, integrity, and timely availability. Additionally, the data of organizations and individuals may face increased risks due to attackers gaining unauthorized access, enabling them to misuse the data for illicit purposes. The consequences of such attacks can be severe, leading to significant financial losses and a breakdown of trust between individuals and organizations. Structured Query Language injection (SQL-i) stands out as one of the most prevalent methods employed to illicitly access data, exploiting a vulnerability in the query statement. This vulnerability grants an attacker swift and effortless access to the data. It consequently allows an unauthorized user to tamper with or erase data, or even hinder legitimate access to it. To counteract these attacks, this research aims to build a model using Machine-Learning (ML) techniques that classifies the type of payloads sent by users. This model aims to reduce the time required for payload classification and to scrutinize the false predictions when classifying SQL requests, along with their ramifications on principles of information security (Confidentiality, Integrity and Availability - CIA). The devised model incorporates a dataset containing harmful and benign payloads to train it, employing Logistic Regression (LR) and Singular Value Decomposition (SVD) techniques. The model demonstrated an impressive accuracy of 98.20%, precision of 98.02%, recall of 99.65% and an F1 score of 98.20%. Furthermore, the time taken to classify a payload was a mere 0.0029 seconds. The constructed model excels in accurately categorizing payloads and significantly reducing processing time, owing to the utilization of the LR model in tandem with SVD, which aids in selecting the most pertinent features for training the model.

Keywords: SQL injection, Logistic Regression, SVD, Confidentiality, Integrity, Availability

1. INTRODUCTION

Web application vulnerabilities refer to security weaknesses that can lead to various types of attacks, including, but not limited to, remote code execution, Structured Query Language injection (SQL-i) and remote file addition. According to the Open Web Application Security Project (OWASP) statistics, SQL-i attacks are prevalent and frequently launched against web applications [1, 2, 3].

The proliferation of web applications and their indispensable role in facilitating data exchange and task completion across various domains has necessitated developing, safeguarding and identifying vulnerabilities in these applications. Hence, considering the impacts of the three fundamental information security principles—confidentiality, integrity and availability—on web applications is imperative [4].

The SQL-i attack is a prevalent form of attack against web applications and is considered one of the top 10 vulnerabilities as per the OWASP classification. SQL-i attacks vary and occur when an assailant modifies the query entered on restricted data or makes unauthorized alterations to the data. The insufficient validation of user input is the most prevalent cause of SQL-i attacks, and considering this insufficiency during the application design phase is imperative for developers [2, 3].

The increasing daily transactions and exchange of data in many areas have made data susceptible to attacks and illegal access. SQL-i attack is one of the common attacks. Standard methods such as static and dynamic analysis to detect and prevent these attacks can provide sufficient results. However, these methods do not work effectively because static analysis yields large false positive results, which affects the CIA of data for organizations and individuals. As for dynamic analysis, it requires a large amount of time to classify the type of payload sent by the client, which affects the

availability of data when needed. Hence, it is necessary to develop another approach to address these problems and detect attacks against them.

This research presents a model for classifying the type of payload sent by a client using Logistic Regression (LR) with the Singular Value Decomposition (SVD) technique to overcome the previous studies, that often did not examine the time required to classify the type of payload sent by the client and whether it is benign or harmful. Additionally, they did not address the impact of incorrect predictions on the three information security principles of CIA. Hence, the objectives of this research are as follows:

Research Motivation and Expected Outputs

- Building a model using Machine-Learning (ML) techniques that classifies the loads sent by the user as whether they contain malicious or normal loads.
- Using the SVD technique to select the best features when building the model.
- Analyzing false predictions and their impact on the CIA of data for individuals and institutions.

Expected Outputs

- Obtaining the highest accuracy and the lowest number of incorrect predictions.
- Achieving the shortest time during the testing phase while implementing the model.
- Conducting a comprehensive analysis on the impact of false predictions regarding False Positive (FP) and False Negative (FN) on the confidentiality, integrity and availability of data.

In this research, we propose a model to detect SQL-I attack by building a model applying LR with SVD, which classifies the type of payloads sent by the client. However, the contributions of this model are as follows:

Primarily, building a trained model using LR with the SVD technique, which works to classify the type of demand sent by the client, determining if the sent request contains malicious or natural commands. This model prevents clients from sending their direct requests to the database without sufficient verification of these entries, thus protecting the server layer from illegal access. Additionally, it reduces the time required to identify the class of the sent payload by using one of the data dimensionality reduction techniques (SVD) and choosing the best features to enhance efficiency and increase classification accuracy.

The second contribution involves studying the impact of incorrect predictions and their implications for the CIA of data.

This paper is organized into several sections as follows: The second section discusses previous studies on the topic of interest. The third section describes the theoretical dimension. Section 4 explains the proposed model designed to detect SQL-i attacks. Sections five and six elucidate the results and draw conclusions.

2. RELATED WORK

This section provides an overview of previous studies on the using ML techniques to detect SQL-i attacks from 2021 to 2023:

1- S. S. A. Krishnan et al. [5] introduced an ML-based model to identify the issue of SQL-i detection. Primarily, using classification methodology to evaluate whether an incoming input comprise an SQL-i or plain text. The issue is subsequently classified utilising various ML algorithms, including the Support Vector Machine (SVM), the Convolutional Neural Network (CNN), the Passive–Aggressive classifier, the Naive Bayes (NB) classifier and LR. The NB classifier exhibits a 95% accuracy rate, while the SVM and Passive–Aggressive classifier demonstrate a 79% accuracy rate. Meanwhile, LR yields a 92% accuracy rate. Supervised learning methods employ multiple elementary classifiers to minimise error and enhance precision, resulting in more precise outcomes. CNN is employed to address the SQL-i classification issue, as opposed to alternative methodologies. The CNN algorithm can achieve an impressive 79% accuracy rate by concurrently adjusting and assessing numerous factors through automation.

2- O. Hubskyi et al. [6] aimed to construct a neural network framework to identify potential SQL-i hazards by analyzing HTTP requests. The methodology entails the classification of URL data into two distinct categories, namely, malicious and benign events. The text provides additional information regarding the expected precision of SQL-i detection through a numerical identification parameter.

3- N. Gandhi et al. [7] proposed the CNN–bidirectional LSTM (BiLSTM)-based model, which demonstrates high effectiveness in detecting SQL-i. This is achieved through the extraction of query information via convolutional layers. The BiLSTM architecture facilitates the acquisition of extended temporal relationships by appropriately sequencing data in forward and backward directions. CNN is utilized to conduct initial feature extraction. The feature extraction process from an embedding matrix involves utilizing a single-dimensional convolutional layer, which employs

filters of varying kernel sizes—specifically, filters with dimensions of 128, 256 and 512 and kernel sizes of 3, 4 and 5. The Bi-LSTM model can analyze data in forward and backward directions in time, leading to the generation of more intricate interpretations. In the end, the system navigates through two fully interconnected layers before reaching a SoftMax layer, scrutinizing the input to ascertain whether it is a malicious or benign query. The proposed CNN–BiLSTM model exhibits a remarkable 98% accuracy rate and outperforms alternative ML algorithms.

4- W. Zhang et al. [8] employed a Deep Neural Network (DNN) architecture to address SQL-i attacks on web applications. The model uses the Rectified Linear Unit (ReLU) to relieve loss and the dropout technique to enhance the model's generalization capability. The training phase contain the conversion of a dataset into a word vector utilizing word pause, followed by the formation of a sparse matrix, which is then passed to the method. The model achieves an accuracy rate 96%.

5- M. Alghawazi et al. [9] proposed a Deep Learning (DL) architecture that uses Recurrent Neural Network (RNN) autocoding to identify SQL attacks. In this paper, an RNN autoencoder with differing optimization techniques have been applied to a comprehensive SQL-i dataset. The outcomes show that the suggested method has an accuracy of 94% and an F1 score of 92%, underscoring the significance of the RNN autoencoder in recognizing SQL attacks. The effectiveness of the more intricate RNN autoencoder design for recognizing SQL attacks warrants further investigation. Additionally, it is recommended that the dataset used in this study be expanded and implemented in real-world scenarios in future studies.

3. THEORETICAL AND BACKGROUND

3.1 PRINCIPLE INFORMATION SECURITY

This section presents the fundamental concepts of information security, crucial elements in protecting web applications and maintaining data confidentiality, availability and integrity.

Confidentiality pertains to safeguarding confidential data and information from unauthorized access by individuals or entities. Specific customers or clients may engage in unauthorized attempts to gain ownership of other users' data with malicious intent.

Integrity relates to ensuring data accuracy. Data should be kept inaccessible to individuals lacking necessary authorization. Due to their accessibility via the Internet, web-based services are highly susceptible to attacks that compromise data integrity. Consistency, accuracy and reliability are imperative for data throughout its life cycle. Implementing multiple backup measures to mitigate the risk of data loss or corruption is a widely adopted approach towards preserving data integrity.

Availability is the third most crucial aspect of data security. This standard concerns the prompt delivery of crucial information from Internet-based applications to end users. Insufficient data regarding customer location can lead to substantial and irreversible financial damages [10, 11].

3.2 STRUCTURED QUERY LANGUAGE INJECTION (SQL-i)

OWASP classifies Structured Query Language injection (SQL-i) attacks as one of the most dangerous attacks towards web applications. Various techniques are employed to illicitly access databases of web applications and acquire information for this form of assault. SQL-I attack transpires when the initial query is unlawfully altered to violate prevailing limitations on the query string. The attack is an inadequate validation consequence of transmitted requests [2, 3].

Structured Query Language (SQL) is the language used to manage databases, but it is also the target of one of the most dangerous attacks on web applications, known as SQL-i. This attack is carried out by attackers who tamper with server databases, resulting in data loss, deletion, modification and copying. The consequences of a successful SQL-i can have a significant impact on the overall security of the system, including confidentiality, data integrity and availability.

In the context of exploiting SQL injection vulnerabilities, an attacker injects malicious SQL statements during the interaction between the database server and the client. SQL is used to write queries and interact with Database Management Systems (DBMSs). The goal of malicious SQL statements is to use them to extract sensitive information or modify data in the database in an unauthorized manner. The success of this type of attack can lead to the leaking of confidential information, the manipulation of data, or even the destruction of the entire database. In addition, successful SQL-i attacks can sometimes bypass database boundaries and execute unauthorized commands on the host operating system, which can lead to more serious repercussions [12].

SQL-i vulnerabilities can be detected in any parameter used in a database query, enabling an attacker to launch a SQL-i attack. These vulnerabilities are a type of security attack that allow attackers to execute malicious queries on the

database in unauthorized ways. This scenario is known as an injection mechanism and can be classified into four main types:

1. **Injection through Cookies:** SQL-i vulnerability can be exploited through cookies. Cookies carry information created and stored on the client side of web applications. When a customer returns to an application, these files are used to restore its previous state information. An attacker can exploit this vulnerability by modifying the content of cookies. For example, if a web application uses cookie content to generate SQL queries, an attacker can easily send an attack by including malicious data in its cookie file.
2. **Injection through User Input:** An injection can be an exploit performed through user input, and is a commonly used way to target vulnerabilities. In this case, the user-supplied input is not monitored or controlled, and instead, is included directly in the SQL expression.
3. **Injection through Server Variables:** This represents a different type of vulnerability exploitation. Server variables include a variety of information such as network headers and environment variables. Web applications handle these variables in multiple ways, such as recording site usage statistics and monitoring browsing trends. When these variables are registered in the database without taking adequate security precautions, it can open the door to vulnerabilities that allow SQL-i to occur.
4. **Second-degree Injection:** This type of security attack is difficult and complex to detect. These attacks involve two stages. At first stage, part of the malicious data necessary to carry out the attack is entered, that will be activated in the second stage. For example, when registering on a web server, an attacker can use the username "'admin'–" to compromise the system. Once logged in successfully, the attacker can modify the password of the newly-created user. A malicious query typically appears as follows:

UPDATE users SET password = 'newpwd' WHERE username = ''admin'–' AND password = 'oldpwd'

Since "–" in SQL marks the beginning of comments, everything that comes after it will be ignored, so the attacker will modify the administrator password [13].

4. METHODOLOGY

This section outlines the various stages of constructing a model utilizing ML or deep learning (DL) techniques. The diagram presented in Figure 1 illustrates the sequential phases involved in constructing a model that classifies SQL queries transmitted to web application databases.

The first stage involves collecting data containing malicious and benign payloads, which will be used to train the proposed model. In the second stage**,** preprocessing is performed on the collected data. The third stage involves dividing the prepared data into two sets, one of which is used for training and the other for testing. In the fourth stage, the first part of dataset is used to trains the proposed model. The second part of dataset is used in the fifth stage to test the model. In the sixth stage, the performance of the model is evaluated using the confusion matrix and a variety of metrics.

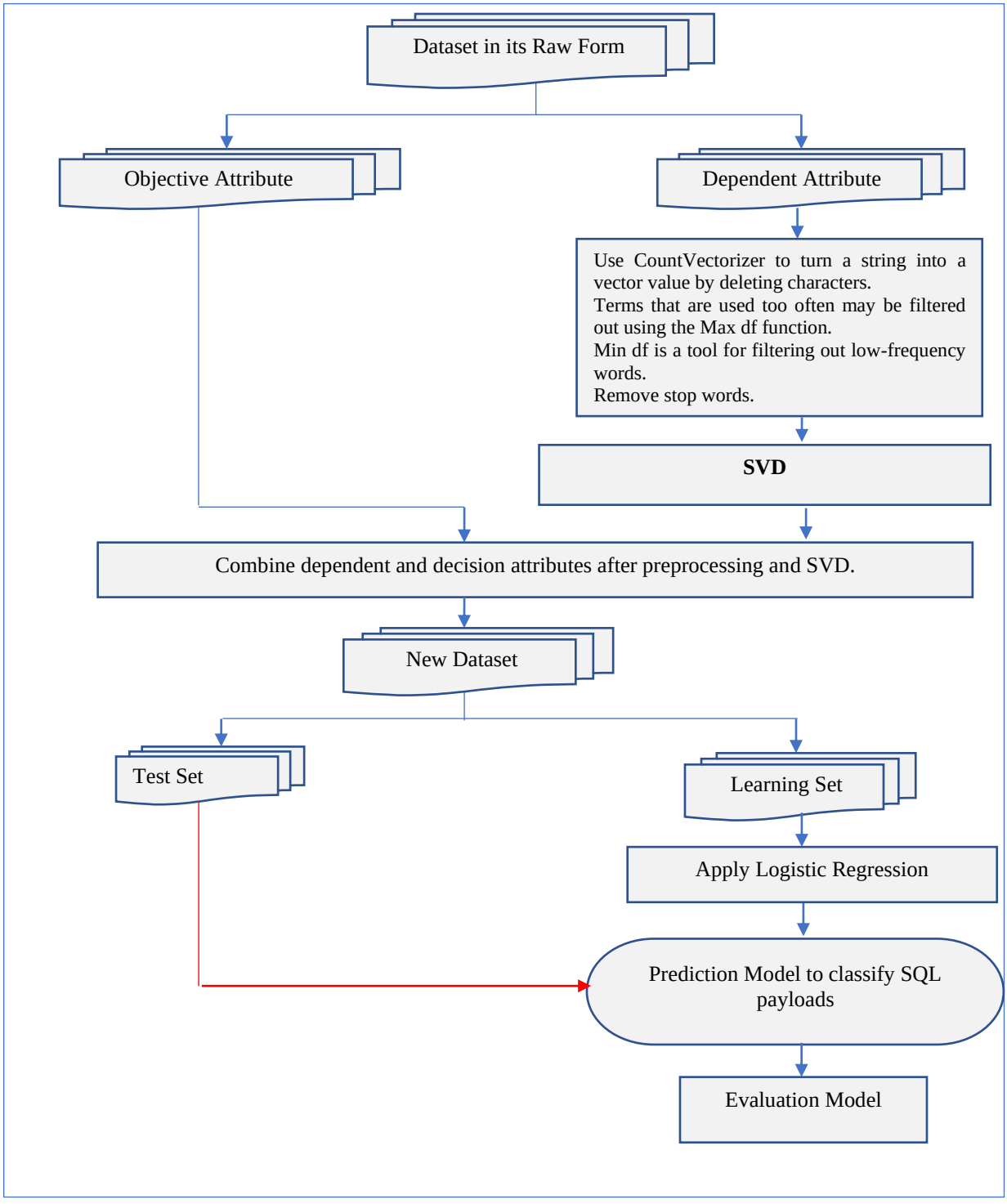

FIGURE 1. Stages of building an ML model

4.1 SQL-i DATASET

The initial step in implementing ML algorithms involves acquiring data, which is subsequently utilized for model training and testing. This study employs a dataset comprising 18,900 instances of detrimental and benign requests. The dataset is partitioned into two subsets. The initial phase pertains to the model training process, encompassing 80% of the complete dataset, whereas the remaining subset is allocated for model testing and evaluation, representing 20%. The figure below represents a sample of the data set used in this model.

```
' union all select @@version--,1
' or 'unusual'  =  'unusual',1
' or 'something'  =  'some'+'thing',1
' or 'text'  =  n'text',1
' or 'something' like 'some%',1
' or 2 > 1,1
' or 'text' > 't',1
' or 'whatever' in  ( 'whatever' ) ,1
' or 2 between 1 and 3,1
' or username like char ( 37 ) ;,1
' union select * from users where login  =  char ( 114,111,111,116 ) ;,1
' union select ,1
password:*/ = 1--,1
uni/**/on sel/**/ect,1
'; execute immediate 'sel' || 'ect us' || 'er',1
'; exec  ( 'sel' + 'ect us' + 'er' ) ,1
'/**/or/**/1/**/ = /**/1,1
' or 1/*,1
 or isNULL ( 1/0 )  /*,1
' or '7659' = '7659,1
" or isNULL ( 1/0 )  /*,1
```

FIGURE 2. Dataset before applying pre-processing

4.2 DATA PREPROCESSING

The second stage represents the application of preprocessing to the data set, and it is necessary to distinguish between state attributes and decision attributes to facilitate data preprocessing. This stage is crucial in building any model because it enables proper fitting of the data and elimination of redundant values and outliers.

Preprocessing is necessary for utilizing ML models, as datasets may frequently need to be more intelligible to them. The primary aim of this methodology is to facilitate data preparation. The data preparation process involves the following:

- Minimizing data quantity.
- Establishing connections among data.
- Standardizing data.
- Eliminating anomalies and redundant values.
- Eliminating null values [14].

Numeric vectors are utilized to represent textual data to overcome ML methodology constraints in processing natural language. As an illustration, all phrases can be encompassed by an isolated vector, whereas article words can denote specific category attributes. Converting data into vector format is commonly referred to as vectorization [15]. Regarding text vectorization, CountVectorizer and TF–IDFVectorizer have frequently utilized options. The CountVectorizer method is a traditional approach to producing class attributes and acquiring numerical attributes from textual information. To optimize the system training process, considering prevailing words and phrases present in the training text is imperative. The CountVectorizer tool can determine the frequency of occurrence of individual words in a given text. The outcome above is attained via implementing the salubrious transformation function of the matrix, which produces a matrix comprising word frequencies [16].

The following algorithm depicts the preprocessing phase utilizing CountVectorizer.

Algorithm 1: Data preprocessing using CountVectorizer
Input: Dataset before preprocessing **Output: Word array**
Begin:
Phase 1: The CountVectorizer method converts a given string into a dictionary. **Phase 2:** Minimizes the frequency of recurring terms. **Phase 3:** Eliminates the least commonly occurring repetitions. **Phase 4:** Removes stop words. **Phase 5:** Converts all vocabulary words to lowercase. **Phase 6:** Organizes the vocabulary in ascending order (recommended). **Phase 7:** The presence of a term in the text is denoted by 1, whereas its absence is indicated by 0. Iterating from Steps 1 to 7 once again to convert the residual text within the dataset into numerical values is necessary.
End

4.3 TRAINING AND TESTING

In ML, splitting data involves partitioning datasets into two subsets. One subset is utilized for training the prediction model, while the other subset is reserved for testing and evaluating model performance. This study employs the holdout technique, which involves partitioning the dataset into two subsets: a training set comprising 80% of the data and a testing set comprising the remaining 20% [17, 18].

The following table shows a breakdown of the dataset that will be used in this model.

Table 1. - INFORMATION OF DATASET

Name of Dataset	Number of Instances	Learning 80%	Testing 20%
SQLIA	18900	15120	3780

4.4 PREDICTION MODEL

The fourth stage involves building a model to specify the type of SQL requests sent by the user. There are various standard methods for analyzing requests sent by the user and classifying them as containing malicious or normal payloads, such as static, dynamic and hybrid analysis. However, these standard methods have several disadvantages. The static analysis method has a high percentage of FP because the model only analyzes the source code and lacks accuracy in classifying new attacks. Dynamic analysis, on the other hand, analyzes requests during runtime, which consumes a significant amount of time to classify the type of payload, impacting the CIA of data within the required timeframe [19]. To address this issue, ML techniques were employed by training and testing the model on real data containing both malicious and natural payloads. This approach yields accurate results as the model learns from the data it was trained on, enabling it to predict new payloads that may be malicious. In addition, the model achieves faster classification of the type of payload sent by the user by using auxiliary techniques to choose the best features, such as Principal Component Analysis (PCA) and SVD.

ML models are categorized into groups: Supervised, Unsupervised, Semi-supervised and Reinforcement Learning. This section will provide an overview of these ML techniques.

Table 2. - SUMMARY OF ML TYPES

Type of Learning	Model	Examples
Supervised	Algorithms or models learn from labeled data using a task-based approach.	Classification, regression
Unsupervised	Algorithms or models use unlabeled data using a data-driven approach.	Clustering, associations, dimensionality reduction
Semi-supervised	Models are created using a collection of data, whether it is labeled or unclassified.	Classification, clustering
Reinforcement	Models receive rewards or punishments according to an environment-driven approach.	Classification, control

In the ML and data science literature, a range of classification techniques have been proposed, below are the most commonly used methods:

- Decision Tree (DT).
- Extreme Gradient Boosting (XGBoost).
- Linear Discriminant Analysis (LDA).
- Support Vector Machine (SVM).
- Adaptive Boosting (AdaBoost).
- Logistic Regression (LR).
- Naive Bayes (NB).
- K-nearest Neighbors (KNN).
- Random Forest (RF).
- Stochastic Gradient Descent (SGD) [20].

This research employs the LR model, which is a supervised ML approach. The paradigm above classifies requests into two distinct groups: safe (category 0) and unsafe (category 1). This approach endeavors to establish a classification that precisely depicts the correlation between independent and dependent variables.

The LR approach is founded on the linear regression methodology, as demonstrated in Equation 1.

$$y = h_0(x) = \theta^T x. \quad (1)$$

Equation 1's predictive performance is suboptimal when the input values are binary. As a result, Equation 2 is utilized to forecast the target feature values accurately.

$$p(y = 1|x) = h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} = \sigma(\theta^T x)$$

$$p(y = 0|x) = 1 - p(y = 1|x) = 1 - h_\theta(x) \quad (2)$$

The utilization of Equation (3), commonly used as the sigmoid function, enables the preservation of the value of $\theta^T x$ within the range of (0, 1). Subsequently, the objective is to identify a numerical quantity such that the probability of y being equal to 1 given x, denoted by p (y = 1|x), is maximized when x is a member of the '1' category but minimized when x is a member of the '0' category, which is equivalent to having a significant probability of y being equal to 0 given x, that is, p (y = 0|x).

$$\sigma(t) = \frac{1}{(1+e^{-t})} \quad (3) [21, 22]$$

4.5 SINGULAR VALUE DECOMPOSTION (SVD)

Upon completion of the preprocessing stage and subsequent conversion of words into a set of vectors utilizing the CountVectorizer, the resultant dataset exhibits high dimensionality. This, in turn, adversely impacts the model performance concerning the time required to classify sent requests. Thus, SVD is employed to decrease the amount of data returns and minimize the impact on model efficiency.

SVD can be defined as a mathematical method that enables the derivation of matrix representations. It can also represent matrices with high dimensions in low dimensions by discarding less significant components and producing approximations with suitable ranks [23].

Equation (4) is referred to define SVD as follows:

$$A_{m*n}=U_{m*r} * \sum_{r*r} * V^T_{r*n} \quad (4)$$

where:

A: denotes the matrix of MxN.

U: denotes the matrix of MxM (its vector is orthogonal, the vector in U is regarded as a left singular vector).

∑: represents the matrix of MxN (all elements are 0 and diagonal items, referred to as singular value).

VT: which is transposition of V, represents the matrix of NxN (its vector is orthogonal; the vector in V is referred to as a right singular vector).

Multiplying the three matrices on the right produces a matrix that is near to A, as r is closer to n and the multiplication outputs are likewise closer to A. The area of the combined matrices is also less than that of the original A matrix. Therefore, only matrices U and V must be stored when the original matrix A is represented in reduction space [24].

4.6 PERFORMACE EVALUATION MEASURES OF THE PREDICTION MODEL

The next section explains the six stages of the model building process, which involves evaluating the model for verification using a set of metrics. The mentioned metrics are derived from a confusion matrix that includes different values. Table 3 displays the confusion matrix values related to four distinct classes, namely, FP, FN, true negatives (TN) and true positives (TP), which serve as the classified outputs of several measurements.

Table 3. - CONFUSION MATRIX

		Predict class	
		Class X	***Class Y***
True class	***Class X***	TN	FP
	Class Y	FN	TP

TP: Used to indicate instances that are classified as malicious and are actually harmful.

FN: Refers to the case where a payload pattern is classified as malicious but is actually benign.

FP: Refers to an entity that is mistakenly classified as harmless despite being malicious.

TN: Concerns cases that are correctly classified by the model as benign loads [25, 26].

In this research, a set of measures are used, as shown in the following equations:

Accuracy refers to the overall number of accurate predictions, encompassing positive and negative classes. The following notation represents a mathematical equation:

$$\textbf{Accuracy} = \frac{\textit{No. of properly identified observations } (TP+TN)}{\textit{Total no.of observation } (TP+TN+FP+FN)} * \mathbf{100} \quad (5)$$

Precision is a metric that calculates the ratio of TP to the sum of TP and FP. The mathematical expression is formally specified as follows:

$$\textbf{Precision} = \frac{\textit{No.of true positives } (TP)}{\textit{No. of true positive+false positive}(TP+FP)} * \mathbf{100} \quad (6)$$

Recall refers to the ratio of TP to the sum of TP and FN. The mathematical formula is formally specified in the following manner:

$$\textbf{Recall} = \frac{\textit{No.of true positives } (TP)}{\textit{No. of true positive+false negative } (TP+FN)} * 100 \quad (7)$$

F1 score refers to the proportionate mean of accuracy and recall. The following is the mathematical expression:

$$\text{F1 score} = 2 * \frac{Precision*Recall}{Precision+Recall} * 100 \quad (8)$$ [27, 28]

5. RESULT AND DISCUSSION

This section shows the results of the proposed model.

5.1 HARDWARE AND SOFTWARE REQUIREMENTS

In developing a model or system, considering some software and hardware requirements is necessary. Tables 4 and 5 present the required standards.

Table 4. - SOFTWARE PREREQUISITE

Software Prerequisite	
Programming Language	Python Programming Languages (Spyder (Anaconda3))
System Type	64-bit OS, x64-based Processor

TABLE 5. - HARDWARE REQUIREMENT

Hardware Prerequisite	
Processor	Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz 2.40 GHz
Installed Random Access Memory (RAM)	8GB RAM
GPU	AMD Radeon Graphics Processor HD (8500M)
Hard Disk	500 GB

5.2 RESULT OF THE PROPOSED MODEL

This section details the outcomes of utilizing an LR approach with SVD to identify potentially malicious payloads within SQL queries transmitted from clients to servers. The following table presents a summary of the obtained results.

TABLE 6. – PREDICTION MODEL RESULT

Sequence	Parameter Name	Value
1.	Learning Phase Score	98.43
2.	Test Phase Score	98.20
3.	Time Complexity	0.0029
4.	Accuracy	98.20
5.	Precision	98.02
6.	Recall	99.65
7.	F1_score	98.20
8.	TP	2881
9.	TN	831
10.	FP	58
11.	FN	10

A comparison between the previous studies and our model is shown in the following table.

TABLE 7. – COMPARION BETWEEN PREVIOS WORK AND THE PROPOSED MODEL

Ref	Model	Accuracy	Time Consuming
[5]	CNN	97	No time mentioned
[6]	The Neural Network of Direct Signal Propagation	95	No time mentioned
[7]	CNN-BiLSTM	98	45 s
[8]	SQLNN	96.16	No time mentioned
[9]	RNN Autoencoder Model	94	No time mentioned
	Proposed Model	98.20	0.0029S

The following figure represents a chart of the results obtained, which represent the accuracy resulting from each model for previous studies and the current study.

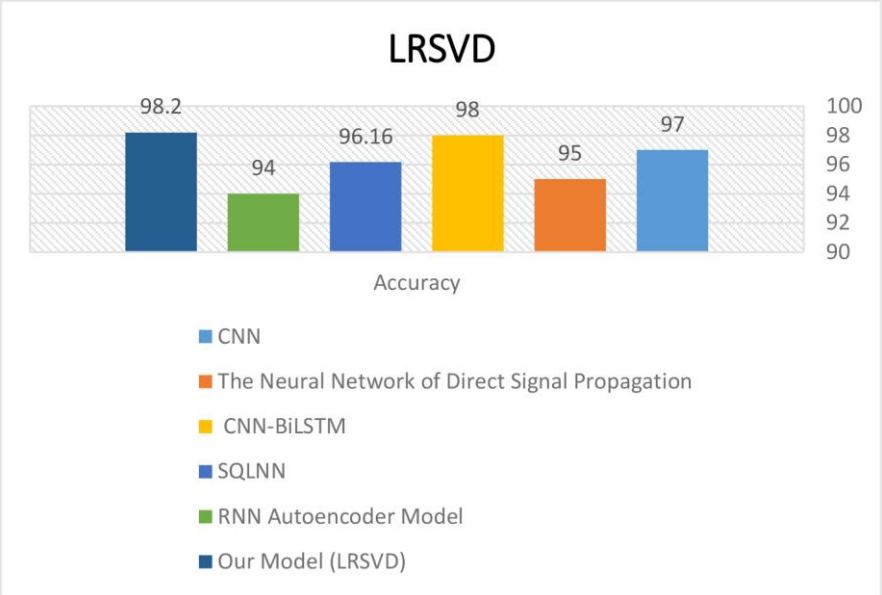

FIGURE 3. Result analysis

The results obtained are derived from applying the model using a dataset containing 18,900 payloads, which were divided into two parts. The first part involves of 15,120 payloads, while the second part contains 3,780 payloads representing harmful and benign content. In terms of dataset separation, a random separation approach was employed, allocating 80% of the dataset for training stage and 20% for testing and evaluation.

Upon comparing prior research, in it became evident that the model developed in this research achieved superior accuracy. Regarding processing time, except for the study [7], none provided specific timeframes. In the mentioned study, it took 45 seconds to determine the type of attack. In contrast, our model achieved an impressively short processing time of 0.0029 seconds, attributed to the implementation of the SVD technique, which effectively reduced data dimensions and selected optimal features. Consequently, our model demonstrated both exceptional accuracy and efficiency.

5.3 IMPACT FALSE PREDICTION ON (CIA)

The subsequent segment of this section entails an examination of the repercussions of inaccurate prognostications and their effects on the fundamental tenets of information security, namely, confidentiality, integrity and availability. These principles serve as the cornerstone for web application development, thus warranting a thorough analysis of the implications of false predictions on the CIA triad.

When a load is incorrectly classified as a benign load by the model that is in fact a harmful load, this case is classified as representing a FP estimate. This error indicates a violation of all three safety standards because the form will allow an unauthorized user to access the data, which gives them the right to steal the data, delete it or prevent organizations and customers from accessing their data.

When a model incorrectly classifies FN a payload as malicious, the model prevents a customer from accessing its data. This results in a violation of the availability of data at the required time, thus violating one of the three CIA information security principles.

The following table represents the impact of incorrect predictions on the three information security principles CIA.

Table 8. IMPACT FALSE PREDICTION ON (CIA)

Model	Confidentiality	Integrity	Availability
FP	Breach of Confidentiality	Breach of Integrity	Breach of Availability
FN	Not Breach Confidentiality	Not Breach Confidentiality	Breach of Availability

6. CONCLUSION

Protecting web applications and identifying potential breaches targeting them in a timely manner is of utmost importance in maintaining the confidentiality, integrity and availability of data for organizations and customers, in addition to adhering to the basic principles of information security. This study proposed a model to detect the type of payloads sent by the user, whether they contain natural or malicious payloads. This is done by applying the model using a dataset containing normal and malicious samples. The model achieved the highest accuracy and the lowest training time.

The main contributions of this model are achieving high accuracy in correct predictions and the lowest number of FPs and FNs, in addition to the shortest time in classifying the load type. This is because when using the SVD technique, the accuracy of classifications is increased and the time taken to detect attacks is reduced due to the use of relevant features in building the model.

The second contribution is when building a model using ML techniques; requests are classified into four values, as shown in Table 3. In this research, the effect of both FP and FN on the three basic principles of CIA information security was studied. Therefore, when the model predicts payloads as FP, this causes the model to classify a malicious payload as benign, which results in users being granted unauthorized access to the data, leading to data theft, destruction or modification. When the model predicts cases as FN, this leads to the model classifying a benign payload as malicious, as this affects the ability to access user data and prevents the user from retrieving this data. The results indicate that reducing the number of FPs in building a model using ML, DL or other technical approaches is essential.

Funding

None.

ACKNOWLEDGEMENT

None

CONFLICT OF INTEREST

None

REFERENCES

[1] O. C. Abikoye, A. Abubakar, A. H. Dokoro, O. N. Akande, and A. A. Kayode, “A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm,” *Eurasip J. Inf. Secur.*, vol. 2020, no. 1, pp. 1–14, 2020, doi: 10.1186/s13635-020-00113-y.

[2] M. S. Aliero, K. N. Qureshi, M. F. Pasha, I. Ghani, and R. A. Yauri, “Systematic Review Analysis on SQLIA Detection and Prevention Approaches,” *Wireless Personal Communications*, vol. 112, no. 4. Springer, pp. 2297–2333, 2020. doi: 10.1007/s11277-020-07151-2.

[3] D. Das, U. Sharma, and D. K. Bhattacharyya, “Defeating SQL injection attack in authentication security: an experimental study,” *Int. J. Inf. Secur.*, vol. 18, no. 1, pp. 1–22, 2019, doi: 10.1007/s10207-017-0393-x.

[4] K. N. Durai, R. Subha, and A. Haldorai, “A Novel Method to Detect and Prevent SQLIA Using Ontology to Cloud Web Security,” *Wirel. Pers. Commun.*, vol. 117, no. 4, pp. 2995–3014, 2021, doi: 10.1007/s11277-020-07243-z.

[5] S. S. A. Krishnan, A. N. Sabu, P. P. Sajan, and A. L. Sreedeep, “SQL Injection Detection Using Machine Learning,” *Rev. GEINTEC-GESTAO Inov. E Tecnol.*, vol. 11, no. 3, pp. 300–310, 2021.

[6] O. Hubskyi, T. Babenko, L. Myrutenko, and O. Oksiiuk, “Detection of sql injection attack using neural networks,” *Adv. Intell. Syst. Comput.*, vol. 1265 AISC, pp. 277–286, 2021, doi: 10.1007/978-3-030-58124-4_27.

[7] N. Gandhi, J. Patel, R. Sisodiya, N. Doshi, and S. Mishra, “A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks,” in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, pp. 378–383.

[8] W. Zhang *et al.*, “Deep Neural Network-Based SQL Injection Detection Method,” *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/4836289.

[9] M. Alghawazi, D. Alghazzawi, and S. Alarifi, “Deep Learning Architecture for Detecting SQL Injection Attacks Based on RNN Autoencoder Model,” *Mathematics*, vol. 11, no. 15, pp. 1–12, 2023, doi: 10.3390/math11153286.

[10] A. A. Mishra, K. Surve, U. Patidar, and R. K. Rambola, "Effectiveness of confidentiality, integrity and availability in the security of cloud computing: A review," *2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018*, pp. 1–5, 2018, doi: 10.1109/CCAA.2018.8777537.

[11] S. Pande, A. Khamparia, D. Gupta, and D. N. H. Thanh, *DDOS Detection Using Machine Learning Technique*, vol. 921. Springer Singapore, 2021. doi: 10.1007/978-981-15-8469-5_5.

[12] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review," *J. Cybersecurity Priv.*, vol. 2, no. 4, pp. 764–777, 2022, doi: 10.3390/jcp2040039.

[13] I. S. Crespo-Mart\'\inez, A. Campazas-Vega, Á. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Álvarez-Aparicio, and C. Fernández-Llamas, "SQL injection attack detection in network flow data," *Comput. Secur.*, vol. 127, p. 103093, 2023.

[14] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4102–4107, 2017.

[15] H. El Rifai, L. Al Qadi, and A. Elnagar, "Arabic text classification: the need for multi-labeling systems," *Neural Comput. Appl.*, vol. 34, no. 2, pp. 1135–1159, 2022, doi: 10.1007/s00521-021-06390-z.

[16] J. S. Yang, C. Y. Zhao, H. T. Yu, and H. Y. Chen, "Use GBDT to Predict the Stock Market," *Procedia Comput. Sci.*, vol. 174, no. 2019, pp. 161–171, 2020, doi: 10.1016/j.procs.2020.06.071.

[17] M. Rafało, "Cross validation methods: Analysis based on diagnostics of thyroid cancer metastasis," *ICT Express*, vol. 8, no. 2, pp. 183–188, 2022, doi: 10.1016/j.icte.2021.05.001.

[18] O. S. F. Shareef, R. F. Hasan, and A. H. Farhan, "Analyzing SQL payloads using logistic regression in a big data environment," *J. Intell. Syst.*, 2023, [Online]. Available: https://doi.org/10.1515/jisys-2023-0063

[19] G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," *Inf. Softw. Technol.*, vol. 74, pp. 160–180, 2016, doi: 10.1016/j.infsof.2016.02.005.

[20] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–21, 2021, doi: 10.1007/s42979-021-00592-x.

[21] C. Zhu, C. U. Idemudia, and W. Feng, "Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques," *Informatics in Medicine Unlocked*, vol. 17. 2019. doi: 10.1016/j.imu.2019.100179.

[22] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augmented Human Research*, vol. 5, no. 1. 2020. doi: 10.1007/s41133-020-00032-0.

[23] D. Falk Soylu, "Dimension Reduction Methods for Predicting Financial Data," *U.U.D.M. Proj. Rep.*, 2015.

[24] Y. Wang and L. Zhu, "Research and implementation of SVD in machine learning," *Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017*, pp. 471–475, 2017, doi: 10.1109/ICIS.2017.7960038.

[25] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[26] I. S. I. Abuhaiba and H. M. Dawoud, "Combining different approaches to improve arabic text documents classification," *Int. J. Intell. Syst. Appl.*, vol. 9, no. 4, pp. 39–52, 2017, doi: 10.5815/ijisa.2017.04.05.

[27] F. K. Alarfaj, "applied sciences Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks," 2023.

[28] A. H. Farhan and R. F. Hasan, "Detection SQL Injection Attacks Against Web Application by Using K-Nearest Neighbors with Principal Component Analysis," in *Proceedings of Data Analytics and Management: ICDAM 2022*, Springer, 2023, pp. 631–642.