DOI: 10.24996/ijs.2025.66.1.29

ISSN: 0067-2904

The Effect of False Predictions of Machine Learning on the Security of the Big Data Environment

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

[1]*Computer Center, University of Fallujah, Anbar, Iraq*

[2]*Computer Sciences Department, University of Technology, Baghdad, Iraq*

Received: 16/5/2023 Accepted: 24/12/2023 Published: 30/1/2025

Abstract

The exchange of data between customers and organizations has become a major target for hackers who seek to illegally access this data, compromising the three main components of security information: confidentiality, integrity, and availability (CIA). Structured query language injection (SQLI) is one of the most common forms of cyberattack. However, most of the previous research has only looked at SQLI attacks that target web-based applications. There hasn't been much time to sort the kind of SQLI payload that the client sent into the vast amounts of data needed to create machine learning models. Additionally, there hasn't been a study that looks at the risks of machine learning models making mistakes and how they affect the three information security principles.

To address this gap, this research aims to create a model that serves as an intermediate protective interface that is a link between the customer's layers and the database server to improve security during communication from SQLI attacks. Additionally, it shortens the time required to identify the client's request type. Finally, study the impact of false predictions of machine learning algorithms on CIA. The proposed method is to train a model using a logistics regression technique (LR) with the Spark ML library that works to process big data containing SQL payloads (harmful and benign).

Comparing our proposed model with previous studies, the results obtained show that the proposed model achieved outstanding results, with an accuracy ratio of 98.10%, a precision ratio of 98.13%, a call ratio of 98.10%, and an F1 index of 98.10%. The results also showed that the time needed to detect and prevent such attacks was only 00.09 seconds.

Keywords: SQLI, Logistic Regression, Big Data, Confidentiality, Integrity, and Availability.

تأثير التنبؤات الخاطئة للتعلم الآلي على أمنية بيئة البيانات الضخمة

عمار حاتم فرحان[1]*، عمر صلاح فرحان[2]، رحاب فليح حسن[3]

[1]مركز الحاسبة الالكترونية، جامعة الفلوجة، الانبار، العراق

[2]قسم علوم الحاسبات، الجامعة التكنلوجية، بغداد، العراق

* Email: omar.alshareef@uofallujah.edu.iq

الخلاصة

نظرًا لحجم وسرعة تبادل البيانات بين العملاء والشركات، أصبحت هذه القناة هدفًا رئيسيًا للقراصنة الذين يسعون لسرقة البيانات الحساسة. يعد حقن SQL (SQLI) أحد أكثر أشكال الهجمات الإلكترونية شيوعًا. على هذا النحو، ركزت غالبية الدراسات السابقة بشكل أساسي على تحديد هجمات SQLI التي تستهدف التطبيقات المستندة إلى الوب، مما ترك الوقت المطلوب لتصنيف نوع حمولة SQLI التي يرسلها العميل في البيانات الضخمة والتي تعد ميزة أساسية عند بناء نماذج التعلم الآلي، فضلاً عن عدم وجود دراسة تحلل مخاطر التوقعات الخاطئة لنماذج التعلم الآلي وتأثيرها على المبادئ الثلاثة لأمن المعلومات.

ولمعالجة هذه الفجوة، يهدف هذا البحث إلى إنشاء نموذج يعمل كواجهة حماية وسيطة تكون بمثابة حلقة وصل بين طبقات العميل وخادم قاعدة البيانات لتحسين الأمان أثناء الاتصال من هجمات SQLI. بالإضافة إلى ذلك، فهو يقلل من الوقت المستغرق لاكتشاف نوع الطلب المرسل من قبل العميل. وأخيراً دراسة تأثير التنبؤات الخاطئة لخوارزميات التعلم الآلي على CIA. الطريقة المقترحة هي تدريب نموذج باستعمال تقنية الانحدار اللوجستي (LR) مع مكتبة Spark ML التي تعمل على معالجة البيانات الضخمة التي تحتوي على حمولات SQL الضارة والحميدة.

وبمقارنة نموذجنا المقترح مع الدراسات السابقة، أظهرت النتائج التي تم الحصول عليها أن النموذج المقترح حقق نتائج متميزة، حيث بلغت نسبة الدقة 98.10%، ونسبة الدقة 98.13%، ونسبة الاتصال 98.10%، ومؤشر 98.10% F1. كما أظهرت النتائج أن الوقت اللازم لكشف ومنع مثل هذه الهجمات كان 00.09 ثانية فقط.

1. Introduction

The rise of users accessing data through online apps has called for safety programs that can withstand assaults on database servers. Therefore, the three fundamental principles of information security—confidentiality, integrity, and availability—are essential when developing online applications [1].

According to the categorization of the Open Web Application Security Projects (OWASP), SQL injection is one of the 10 most critical vulnerabilities in online applications and one of the most popular attacks against them. In this type of attack, the attacker alters the SQL query to retrieve data illicitly. Given the prevalence of SQLI attacks, developers of online applications must take special care to validate user input at every stage of the development process [2], [3], and [4].

The prevalence of big data and how it is processed and managed have emerged as important issues. Businesses can more reliably manage, analyze, and protect huge amounts of data through machine learning and artificial intelligence technology [5], [6].
Given that a growing amount of information is being transferred and stored online, users are at risk of cyberattacks aiming to steal private information from businesses and individuals [7], [8].

The confidentiality, integrity, and accessibility of private data held by businesses and individuals may be at risk due to inaccurate predictions from SQL payloads on big data. Accordingly, this research aims to:

- Creating a model that acts as an intermediate protective interface represents a separation layer between the client layer and the database server layer. The purpose of this layer is to increase security and ensure the integrity of data during communication processes and interaction between systems from SQL injection attacks.
- Interpret the effect of false predictions of machine learning models on the CIA. This study assesses the extent to which FP affects the confidentiality, integrity, and availability of data in

a timely manner, as well as the extent to which FN affects the availability of data in a timely manner.

- Reduce the time taken to determine the type of load sent from client to server by employing a Spark library that reduces the time taken when handling a large dataset.

This research presents a model to specify the type of requests sent by the client, whether they contain malicious or benign SQLI payloads. The model uses the Logistic Regression algorithm with the Spark ML library, which works as a model for separating the client and the database for the purpose of protecting data from direct access. However, the following are the contributions of our research:

- Firstly, build a model that protects big data from unauthorized attacks using LR with the Spark ML library. This approach works as a middle layer between the user and the database. This layer analyzes payloads sent from the customer's layer to determine whether the payload is harmful or safe.
- The second contribution is analyzing the predicted values by the proposed model, which assesses the extent to which inaccurate predictions affect the CIA of the data.
- The third contribution is reducing the required time to determine the payload type using the Spark ML library. The optimal use of the Spark ML Library indicates that it operates in memory, which saves time when classifying loads.

In the following paragraphs, we will provide a summary of how this paper is organized. The second section of this research will present previous studies, while the third section will explain the theoretical part of this study. The fourth section explains the methodology of this work. The fifth and sixth sections will explain the results, discussions, and conclusions of this work.

2. Related Work

Although there have been many previous studies on identifying and mitigating SQL injection attacks targeting web-based applications, it is worth noting that no research has yet been carried out that involves detecting and preventing such attacks in the big data environment, as well as analyzing the risks of mispredictions of machine learning models and their impact on the principles of data security. The majority of earlier studies did not mention how long it would take to classify the client's payload type. That is a key feature when building machine-learning models. This section will present the most of our research from previous studies.

This paper [9] looks at how to make a dataset with extracts from known attack patterns. The dataset will include SQL tokens and symbols that are present at injection points. To test how well the method worked, a web-based app was made to show how to use vector variables that contained dictionary word lists. This showed how to handle a lot of learning data. The provided dataset has undergone pre-processing, labeling, and feature hashing in preparation for supervised learning. The proposed solution involves deploying a trained classifier as a web service, which a custom dot-net application will utilize. This application will implement a web proxy API to intercept web requests and accurately predict SQL injection attacks, preventing malicious submissions from reaching the protected back-end database. This study presents a comprehensive demonstration of implementing a machine learning-based predictive analytics system and deploying the resulting web service. For accurate prediction and prevention of SQL injection attacks, the system is made. Its effectiveness is tested using Confusion Matrix (CM) and Receiver Operating Curve (ROC) analyses.

On the other hand, O. Hubskyi et al. [10] present a study that built a neural network-based approach to detecting possible risks associated with SQL injection in HTTP requests. The approach determines whether a Uniform Resource Locator (URL) constitutes a malicious attack or a harmless action. Furthermore, a quantifiable metric for evaluating the efficacy of

SQL injection detection is presented. Using the previously mentioned approach, the neural network model will be fed with 12 neurons. A multilayer perceptron based on Rumelhart's methodology was utilized to synthesize a neural network model. You can precisely change the weight coefficients of the neurons in Rumelhart's multilayer perceptron, which is a specific form of the Rosenblatt perceptron, by using the error-reversing process. The distinctive feature of this phenomenon lies in its multilayered nature, typically comprising two or three strata. The Rosenblatt perceptron, a type of neural network, categorizes input vectors into binary classes of zero and one. The categorization of each input vector is determined by utilizing the purposeful array Y, which is a constituent of the training sequence and comprises two arrays. The synthesized model has achieved 95% accuracy upon training on the input data.

Furthermore, Kranthikumar and Velusamy [11] focus on the study of comparative research pertaining to the detection of SQL injection. The Support Vector Machine (SVM), Naive Bayes, the Gradient Boosting technique, and the Regular Expression (REGEX) class are four machine learning techniques developed and evaluated. The classification of SQL input, whether from the original query or the applied query, can be achieved by using a classifier known as REGEX. This classifier employs regular expressions as its primary instrument. The procedures were evaluated using a composite dataset comprising 20,474 SQL injection queries. The test results show that the REGEX method needed 3.98 seconds of computing time to find an SQL injection attack when the pattern scanning method was used. REGEX exhibits a high level of performance, boasting an accuracy rate of 97%.

Finally, N. Gandhi et al. [12] showed that a CNN-BiLSTM-based model can find SQL injection by using convolutional layers to get query information. The BiLSTM architecture facilitates the acquisition of extended temporal relationships by sequentially analyzing data in both forward and backward directions. Convolutional neural networks are utilized to conduct the initial feature extraction. To get features out of an embedding matrix, you need to use a single-dimensional convolutional layer with filters that have kernel sizes of 128, 256, or 512. These kernel sizes are 3, 4, and 5, respectively. The Bidirectional Long-Term Memory (Bi-LSTM) model can process data in both forward and backward directions, enabling more detailed analyses. In the end, the input passes through two fully interconnected layers before reaching a SoftMax layer, which analyzes it to ascertain whether the query is malicious or benign. The CNN-BiLSTM model that was suggested exhibited a notable accuracy rate of 98% and outperformed alternative machine learning algorithms. Table I shows a summary of the previous studies and the problems related to this study.

TABLE I. :A SUMMARY OF PREVIOUS STUDIES

Ref	Model	Accuracy	Problem Definition
[9]	SVM	98.6	• This research uses the synthetic minority over-sampling technique to create a balance in the data set used in this study that increases the accuracy of the model as well as ensures that the data set is not matched to the truth. Did not mention the time taken to detect the type of load sent by the customer. The impact of this model's false predictions on the CIA of the data was not mentioned.
[10]	Neural Network of Direct Signal Propagation	95	The problem with this study does not mention the time it takes to determine the type of payload the user is sending, and whether it contains benign or malicious payloads. Moreover, the effect of incorrect predictions on the performance of this model was not mentioned
[11]	Support Vector Machine Gradient boosting Naive Bayes classifier REGEX classifier	94.92 94.27 70.79 97.48	The consuming time to verify the load type is 3.98 seconds, which affects the response speed of sending a request.

[12]	CNN-BiLSTM	98	• The time it takes to verify the load type is 45 seconds, which affects the response speed of sending a request. • The impact of this model's false forecasts on the confidentiality, integrity and availability of data was not mentioned.

3. Theoretical Framework and Background

This section will discuss the basic concepts related to this study. The main focus will be on the basic principles of information security, methods of data processing, the theoretical aspect of the model proposed in this study, and the addition of evaluation methods.

3.1 Principal Information Security

The principles of information security discussed here are among the most crucial aspects of shielding online applications and preserving their data's confidentiality, availability, and integrity.

The security of web applications is assessed on the basis of the three safety standards of confidentiality, integrity, and availability. Regardless of the kind of system being developed, these principles outline the fundamental requirements for developing web applications. Consequently, predictive outcomes will be evaluated using the following three principles [7].

Confidentiality is a means of protecting sensitive information and data against unauthorized access by unidentified people. Consequently, some individuals may seek unauthorized access to the data and information of organizations and individuals, exposing them to substantial financial losses and violating their confidentiality.

Integrity is the process of guaranteeing data accuracy. Unauthorized parties should be prevented from modifying the data. Web-based services are highly susceptible to malware attempts that compromise the data's integrity when accessed via the Internet. Information must preserve its consistency, precision, and dependability throughout its lifecycle.

Availability is the third-most essential aspect of the security of information. This standard pertains to conveying time-sensitive data to individuals via web-based applications. A lack of data for businesses and clients can result in enormous and calamitous losses [13], [14].

3.3 Data Pre-Processing

Data preparation aims to reduce the quantity of data, establish associations among the data, standardize the data, remove noise and duplicate values, and fill in missing values [15].

Retrieving numbers from text and turning them into class features is a popular application of the CountVectorizer method. Only frequently occurring terms in the training material are taken into consideration. By employing a matrix fit transform, "CountVectorizer" is able to produce a term frequency matrix from which individual word frequencies can be determined [16] [17]. Figure 1 represents a description of the data pre-processing process.

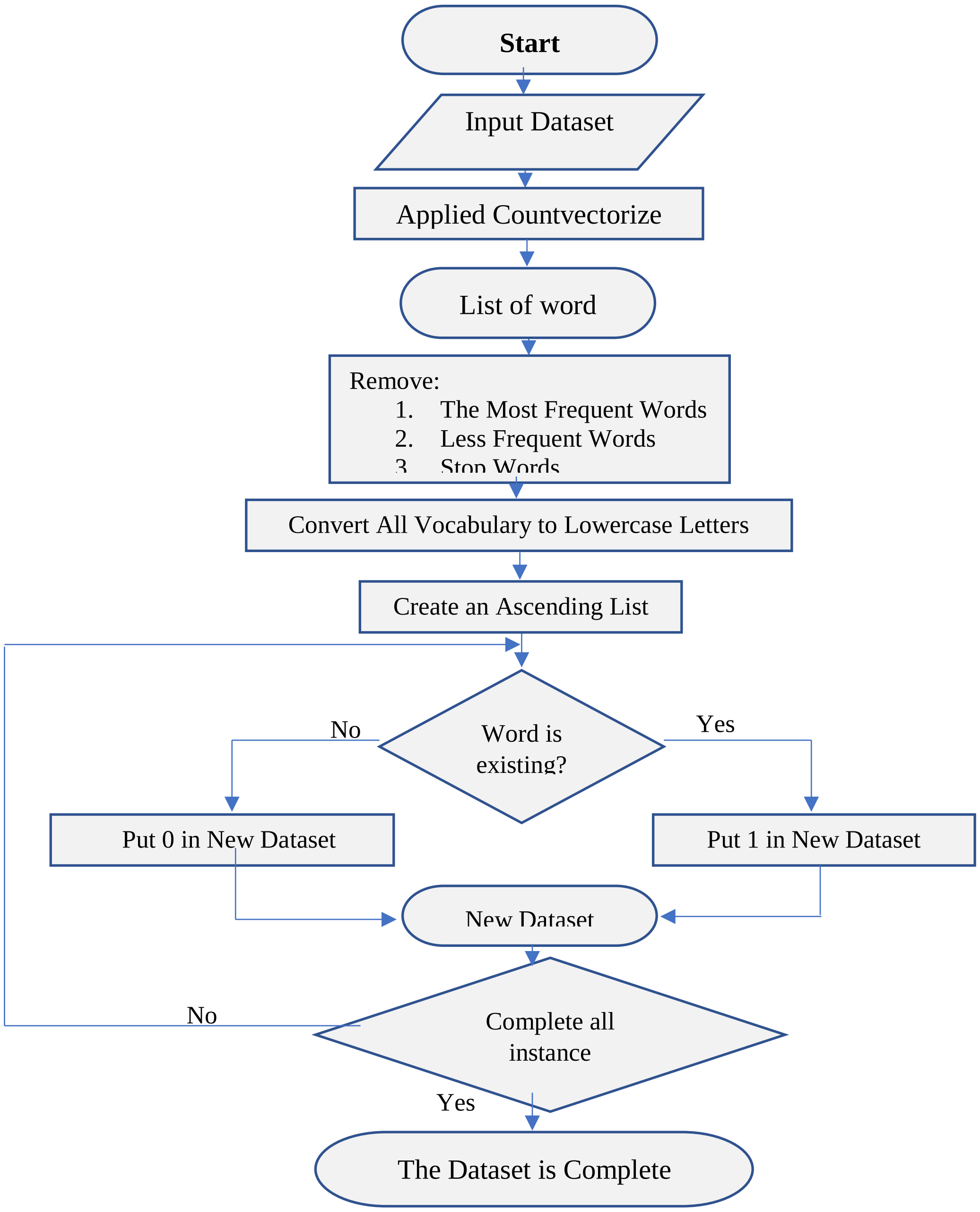


Figure 1: pre-processing

In this study, the countvectorize method was used to convert the text data set into digital data. This method enters the data set and applies a series of procedures, as shown below:

- Convert text to a word list using the CountVectorizer.
- Remove redundant phrases.
- Remove everything except the most frequently used words.
- Remove all stop words.
- Change all vocabulary to lowercase.
- Create an ascending list of all the words.
- A value of 1 indicates the presence of the term in the text, and a value of 0 indicates its absence.
- Repeat 1–7 to convert the text dataset to a digital form.

3.4 Training Testing

In the present research, the holdout strategy is used to divide the data set into a set for training representing 80% of the data and a test set representing 20%.

3.5 Prediction Model

The present research uses supervised ML. This approach divides queries into two types (0 and 1), namely, benign and hazardous. This strategy aims to develop a classification that identifies the relationship between independent and dependent variables.

The linear regression procedure in the below equation determines how the logistic regression approach works (1).

$$j = h_0(i) = \theta^T i. \tag{1}$$

Equation (1) is ineffective when coping with binary data. Consequently, equation (2) is utilized to predict whether the submitted query contains a malicious payload (the likelihood 1) or a safe payload (the likelihood 0).

$$p(j = 1|i) = \ h_\theta(i) = \frac{1}{1+\exp(-\theta^T i)} = \sigma(\theta^T i),$$

$$p(j = 0|i) = \ 1 - p(j = 1|i) = 1 - h_\theta(i). \tag{2}$$

Equation (3), which refers to the sigmoid function, can retain the value of $\theta^T i$ within [0, 1]. Then, we look for a number such that $p(j = 1|i) = \ h_\theta(i)$, i.e., $p(j = 0|i)$, is big when i belongs to the 0 class and small when i belongs to the 1 class.

$$\sigma(t) = \frac{1}{(1+e^{-t})}. \tag{3}$$ [19],[20]

3.6 Performance Evaluation of Prediction Model

In the last stage of constructing prediction models, the model is evaluated using a variety of indicators, including accuracy, time, precision, and recall, to determine the results.

These scales are generated using a confusion array with multiple values. As shown in Table II, the confusion matrix, which contains elements (FP, FN, TN and TP), represents the outcomes of a classifier applied to multiple scales.

TABLE: CONFUSION MATRIX

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

True positive (TP): the percentage of cases that are accurately identified by the prediction model as containing dangerous payloads.
False negatives (FN): are negative results that the approach expected accurately.
False positives (FP): are positive results that the approach expected inaccurately.
True negative (TN): indicates how many instances a prediction model correctly identifies a case as having a benign payload [21], [22], [23]
The formulas below represent the metrics used to assess the model and learn about its performance efficiency.
Accuracy refers to the sum of all correct forecasts, whether positive or negative. The formula is as follows:

$$\mathbf{Accuracy} = \frac{\boldsymbol{No.of\ properly\ identified\ observations\ (TP+TN)}}{\boldsymbol{Total\ no.of\ observation\ (TP+TN+FP+FN)}} * \mathbf{100}. \tag{4}$$

Precision indicates the TP to sum TP and FP. The mathematical equation is defined as follows:

$$\mathbf{Precision} = \frac{\boldsymbol{No.of\ true\ positives\ (TP)}}{\boldsymbol{No.\ of\ true\ positive+false\ positive(TP+FP)}} * \mathbf{100}. \tag{5}$$

Recall indicates the ratio of TP to the sum of TP and FN. The mathematical equation is defined as follows:

$$\text{Recall} = \frac{No.of\ true\ positives\ (TP)}{No.\ of\ true\ positive+false\ negative\ (TP+FN)} * 100. \tag{6}$$

The F1-score is the proportional mean of precision and recall. The mathematical expression is as follows:

$$\text{F1} - \text{score} = 2 * \frac{Precision*Recall}{Precision+Recall} * 100. \tag{7}$$

4. Methodology

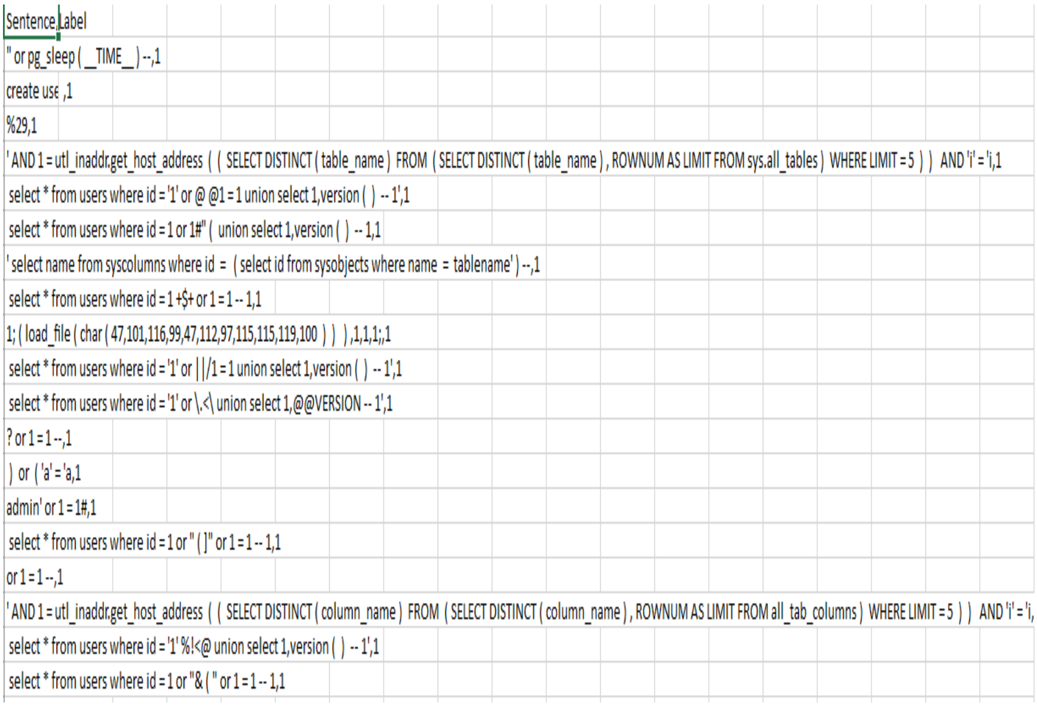
4.1 SQLI Datasets

The model in this work uses a dataset that contains 85,974 samples, divided into two sets based on sample type. The first set used 80% of the dataset for a training phase. while the second set uses 20% of the dataset to test and evaluate the model. The dataset used in this study is summarized in the table below.

TABLE II. DATASET SUMMARY

Dataset Name	Number of cases	Learning step	Testing step
SQLI	85974	68604	17370

The dataset used in this paper was collected from the Kaggle website [24], where the original data contained 109,518 samples, including both malicious and benign payloads. The data set used in this work is illustrated in the following figure:



```
Sentence,Label
" or pg_sleep ( __TIME__ ) --,1
create use ,1
%29,1
' AND 1 = utl_inaddr.get_host_address ( ( SELECT DISTINCT ( table_name ) FROM ( SELECT DISTINCT ( table_name ) , ROWNUM AS LIMIT FROM sys.all_tables ) WHERE LIMIT = 5 ) ) AND 'i' = 'i,1
select * from users where id = '1' or @ @1 = 1 union select 1,version ( ) -- 1',1
select * from users where id = 1 or 1#" ( union select 1,version ( ) -- 1,1
' select name from syscolumns where id = ( select id from sysobjects where name = tablename' ) --,1
select * from users where id = 1 +$+ or 1 = 1 -- 1,1
1; ( load_file ( char ( 47,101,116,99,47,112,97,115,115,119,100 ) ) ) ,1,1,1;,1
select * from users where id = '1' or ||/1 = 1 union select 1,version ( ) -- 1',1
select * from users where id = '1' or \.<\ union select 1,@@VERSION -- 1',1
? or 1 = 1 --,1
) or ( 'a' = 'a,1
admin' or 1 = 1#,1
select * from users where id = 1 or " ( ]" or 1 = 1 -- 1,1
or 1 = 1 --,1
' AND 1 = utl_inaddr.get_host_address ( ( SELECT DISTINCT ( column_name ) FROM ( SELECT DISTINCT ( column_name ) , ROWNUM AS LIMIT FROM all_tab_columns ) WHERE LIMIT = 5 ) ) AND 'i' = 'i,
select * from users where id = '1' %!<@ union select 1,version ( ) -- 1',1
select * from users where id = 1 or "& ( " or 1 = 1 -- 1,1
```

Figure 2: Dataset Before Pre-Processing

4.2 Framework

The model framework that is built to detect and prevent SQL injection attacks consists of three layers. The first layer represents the data transmission, which represents the client layer through which requests are sent to the database server. The hidden layer represents the protection layer, which contains the logistic regression framework using Spark ML, which deals with big data, for client-side request classification. The third level is the database server layer, which stores data for organizations and individuals.

When the client sends the request, the intermediate layer (the protection layer) receives it and examines it to see if the payload is malicious or benign. If the payload is determined to be dangerous, the model will stop it from moving on to the third layer and store it in the dataset instead. When the number of harmful payloads provided surpasses 1,000, the model is retrained. The goal of preserving these payloads is to improve the training of the model on new samples that were not previously present in the basic dataset.

The proposed approach to identifying and preventing attacks on large data sets is depicted in the figure below:

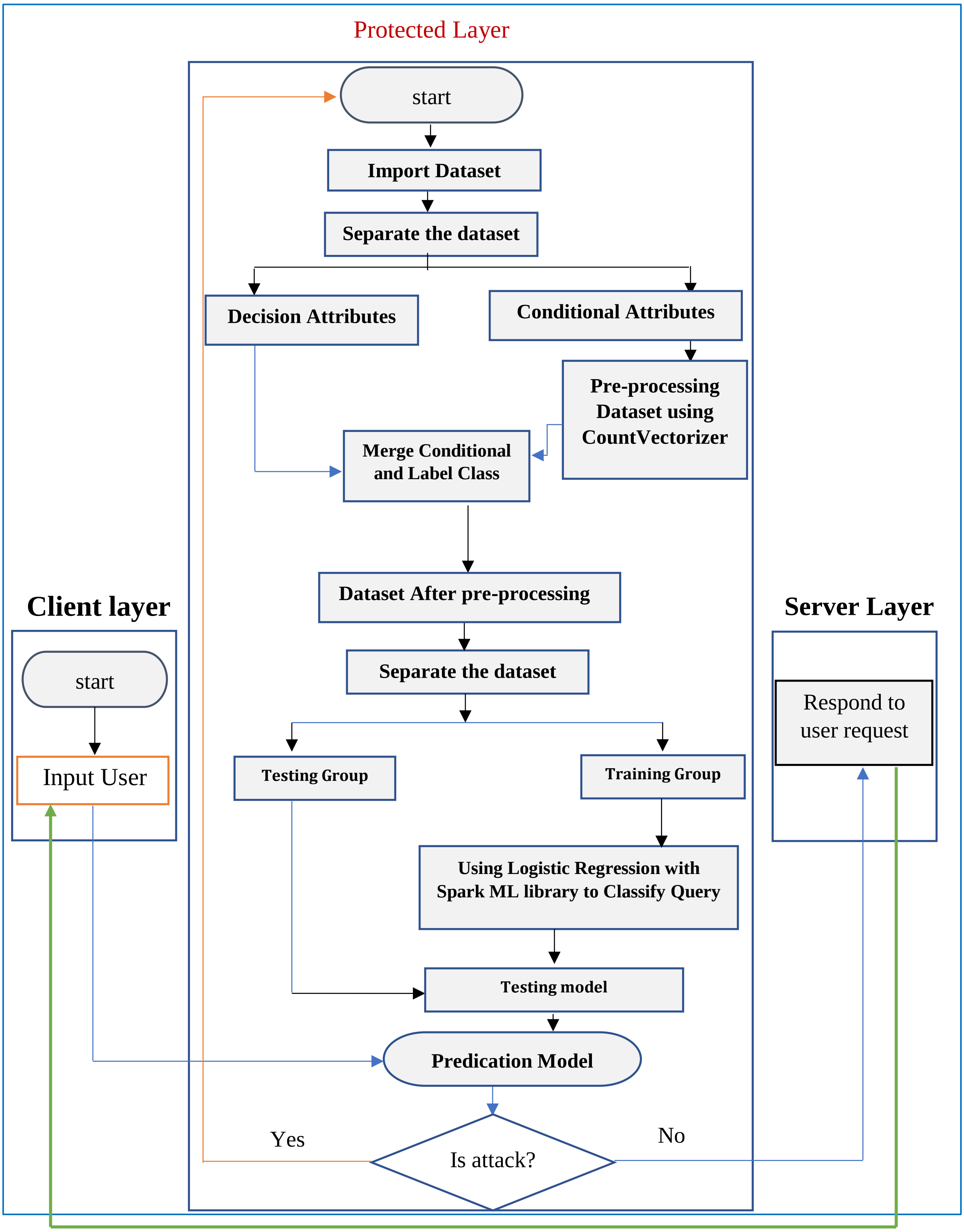


Figure 3: The Proposed System to Classify Submitted Queries

5. Result and Discussion

The proposed approach examines the results of the values (FP and FN) that are obtained when applying machine learning models and then analyzes the impact of these values on the performance of these models. In this section, we present details of the impact of incorrect predictions on the CIA. In addition to the physical and software requirements used in this study, as well as the results obtained when implementing the logistic regression model, compare the results of previous studies with the current study.

5.1 Requirements

The basic requirements used in this work will be explained in Tables IV and V.

TABLE III. SOFTWARE REQUIREMENT

Software Requirement	
System type	64 - bit OS , x64-based processor
Programming Language	The programming language Python (utilizing Spyder within Anaconda3).

TABLE IV. HARDWARE REQUIREMENT

Hardware Requirement	
Processor	Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz 2.40 GHz
Installed RAM	8GB
Hard disk	500 GB.

5.2 Logistic Regression Model Results

The results are shown in the table below that were obtained after applying this model to a dataset containing 85,974 payloads, which were separated into two parts. The first part includes 45,051 safe loads, while the other part contains 40,923 harmful loads. The rejection method was used to divide the data, with 80% of the group selected for training, leaving the remaining percentage for testing and evaluation. However, the following tables show the results of our work, where table VI shows the accuracy, precision, recall, and F1 score of the proposed model. Table VII represents the number of correct and incorrect predictions. Table VIII presents the time taken during the training and testing phases.

TABLE V. RESULT OF THE PREDICTION MODEL

Name of Parameter	Value
Accuracy	98.10
Precision	98.13
Recall	98.10
F1 score	98.10

TABLE VI. CORRECT AND INCORRECT PREDICTIONS

Name of Parameter	Value
True positive	8135 case
False positive	260 case
True negative	8657 case
False negative	64 case

TABLE VII. TIME CONSUMING

Name of Parameter	Value
Time Training set	30.05s
Time Testing set	00.09s

5.3 Effect of Incorrect Predictions

False positive prediction (FP) shows that the proposed approach fails to meet the three safety requirements, as the method incorrectly predicts that the carried payload is harmless when it is dangerous. As a result, the web application receives a fraudulent request, providing the attacker with unauthorized access to the data and compromising the confidentiality, integrity, and availability of the data.

A false negative prediction (FN) indicates that the system violated one of the three safety standards because the model incorrectly predicted that the transported payload was hazardous when in fact it was harmless. Thus, it hinders the web application from making a benign request, thus preventing the actual user from receiving the data and resulting in a loss of availability.

5.4 Comparison with Previous Studies

The following table describes the comparison between the results of previous studies and our approach.

TABLE VIII. COMPARISON BETWEEN CURRENT AND PREVIOUS STUDIES

Ref	Model	Accuracy	Size of Dataset
[8]	SVM	98.6	181303
[9]	Neural Network of Direct Signal Propagation	95	30,233
[11]	Support Vector Machine Naive Bayes classifier Gradient boosting REGEX classifier	94.92 70.79 94.27 97.48	20474
[14]	CNN-BiLSTM	98	4,200
	Our model	98.10	85,974

6. Conclusions

This study explains how to build a supervised logistic regression model to detect and prevent SQL injection attacks. The model has obtained an accuracy of 98.10 results, while the time taken to detect these attacks is 00.09 seconds as a result of using the Spark ML library. However, the Spark ML performs memory operations in a distributed manner. Thus, the use of this library reduces the time required to detect and prevent data attacks and improves the accuracy of the approach in classifying the sent payloads. When ML models are used, they generate a set of values. Each value affects the model’s effectiveness regarding confidentiality, integrity, and data availability. From these values, the results are as follows:
False positive values impact the three information security principles of the CIA because the approach classifies requests sent with malicious payloads as benign payloads. Thus, unauthorized users are able to access and violate the fundamental principles of information security, as false positives result in significant losses for institutions and individuals. Thus, authorized institutions and individuals lose access to their data.

The false negative occurs when a model incorrectly identifies a benign payload as harmful, thus blocking access to the data even for authorized users. As a result, authorized users lose access to their data.
For the real advantages and disadvantages, it does not affect the availability of data to users because the model classifies the payloads sent correctly.
Therefore, while developing a predictive model with ML approaches, deep learning, or any other technology, the number of false positives and negatives that can slow down web apps and prevent legitimate users from accessing crucial data for their business must be reduced.

However, the proposed model can classify one type of attack, which is only an SQL injection attack. Future work is needed to classify more than one type of attack, such as scripting attacks across sites or DDOS attacks, using a deep learning algorithm.

References

[1] 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.

[2] 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.

[3] Shareef, Omar Salah F., Hasan, Rehab Flaih and Farhan, Ammar Hatem. "Analyzing SQL payloads using logistic regression in a big data environment," *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20230063, 2023. https://doi.org/10.1515/jisys-2023-0063

[4] 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.

[5] A. Haldorai, S. Devi, R. Joan, and L. Arulmurugan, "Big Data in Intelligent Information Systems," *Mob. Networks Appl.*, no. October 2021, pp. 997–999, 2022, Doi: 10.1007/s11036-021-01863-w.

[6] A. H. Farhan and R. F. Hasan, "Using random forest with principal component analysis to detect SQLIA," in *AIP Conference Proceedings*, vol. 2839, no. 1, p. 040012, 2023. https://doi.org/10.1063/5.0167783

[7] M. J. Awan *et al.*, "Real-time ddos attack detection system using big data approach," *Sustain.*, vol. 13, no. 19, pp. 1–19, 2021, Doi: 10.3390/su131910743.

[8] O. S. F. Shareef and A. M. Sagheer, "Implementing a Distributed Certificate Authority Using Elliptic Curve Cryptography for Big Data Environment," in *2020 2nd Annual International Conference on Information and Sciences (AiCIS)*, IEEE, 2020, pp. 132–140.

[9] S. O. Uwagbole, W. J. Buchanan, and L. Fan, "Applied Machine Learning predictive analytics to SQL Injection Attack detection and prevention," *Proc. IM 2017 - 2017 IFIP/IEEE Int. Symp. Integr. Netw. Serv. Manag.*, pp. 1087–1090, 2017, Doi: 10.23919/INM.2017.7987433.

[10] 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.

[11] B. Kranthikumar and R. L. Velusamy, "SQL injection detection using REGEX classifier," *J. Xi'an Univ. Archit. Technol.*, vol. Volume XII, no. VI, pp. 800–809, 2020.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Rehab Flaih Hasan, O. S. F. Shareef, and Ammar Hatem Farhan, "Analysis of the False Prediction of the Logistic Regression Algorithm in SQL Payload Classification and its Impact on the Principles of Information Security (CIA)," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 4, pp. 191–203, Nov. 2023.

[19] 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, p. 100179, 2019. Doi: 10.1016/j.imu.2019.100179.

[20] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification,” *Augment. Hum. Res.*, vol. 5, p. 12, 2020, Doi: 10.1007/s41133-020-00032-0.

[21] 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.

[22] 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.

[23] Alarfaj, Fawaz Khaled, and Nayeem Ahmad Khan, "Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks," Applied Sciences, vol. 13, no. 7, p. 4365, 2023. https://doi.org/10.3390/app13074365

[24] “https://www.kaggle.com/datasets/gambleryu/biggest-sql-injection-dataset?resource=download.”