CP8320 Project Report
Detection of SQL Injection with a Machine Learning Approach

Urmi Patel (501064008)
Ryerson University, Toronto, Canada
urmi.patel@ryerson.ca

ABSTRACT

Cyber-attacks are increasing day by day because of various kind of online transactions, such as online banking and/or shopping. There are various kind of attacks performed against web applications such as Cross Site Scripting (XSS), Structured Query Language (SQL) injection attacks amongst others. SQL injection is one among the ten top vulnerabilities against web applications to manipulate the backend database through the server. Presently, machine learning (ML) and deep learning algorithms are used widely to discover or stop different cyber safety issues. This paper described various machine learning algorithms and compared those with each other to find best amongst them.

KEYWORDS

Model prediction, SQL injection, SVM, Vulnerability, Decision tree, Naïve bayes, Neural network, Accuracy, Regression, Classification techniques, KNN

1. INTRODUCTION

The primary goal of this project is to develop a machine learning (ML) based classifier using supervised learning methods (classification and regression) to identify whether the inputted data by users contains SQLi vulnerabilities.

1.1 MOTIVATION

SQL is a language that is used to communicate between databases, to access or store the information in it. Whenever someone tries to crack the communication or forge the data from the database, we called it an SQL attack.

SQL injection, also known as SQLi, is a common type of attack that uses malicious SQL code for manipulating the database to access the data that was not intended to be exposed.

Throughout the past decade, SQL injection (SQLi) vulnerabilities have been consistently ranked by the Open Web Application Security Project (OWASP) as a top security risk [1]. SQLi attacks target database systems by injecting SQL code into vulnerable input parameters that are not properly checked and sanitised. This injected code could change the application’s behaviour or even the system’s data. Additionally, recent vulnerability reports found that web-based systems can receive up to 26 attacks per minute [2].

1.2 RELATED WORK

Currently, machine learning algorithms are used to identify various cyber security attacks is being debated largely. Moreover, the power of using supervised and unsupervised learning methods to detect security problems cannot be questioned, the required resources and time to execute such difficult algorithms remains a major concern for the cyber security community [3]. Any machine learning methods are not perfect for any specific problem. The reason behind this is various dataset and different parameters that affect the algorithms result. Five algorithms were implemented and compared in this paper, to find which one gives the best accuracy and results.

1.3 RESEARCH PROBLEM

Figure 1 shows the process of how an attacker can attack the system and access the database. Normal user is any user that can access the web page.

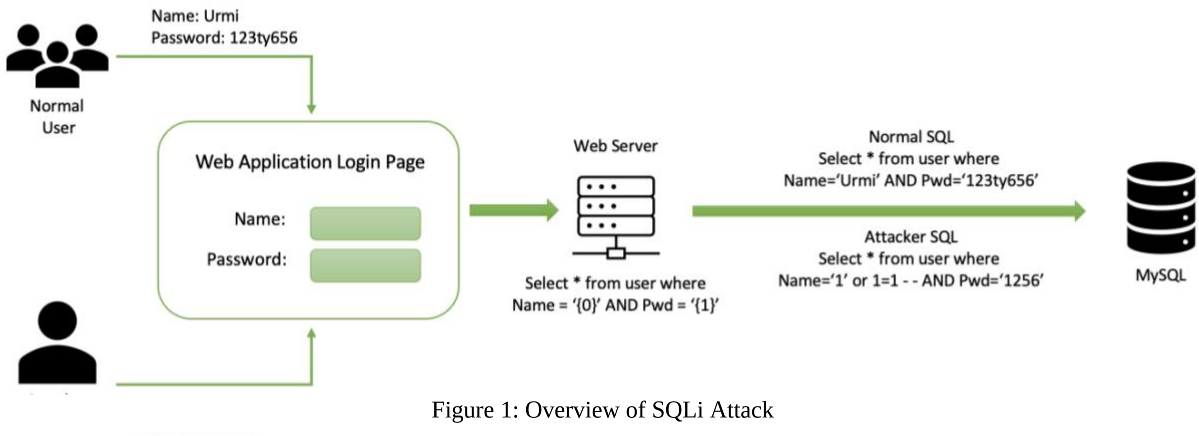

Name: Urmi
Password: 123ty656
Normal User
Web Application Login Page
Name:
Password:
Web Server
Select * from user where Name = '{0}' AND Pwd = '{1}'
Normal SQL
Select * from user where Name='Urmi' AND Pwd='123ty656'
Attacker SQL
Select * from user where Name='1' or 1=1 - - AND Pwd='1256'
MySQL

Figure 1: Overview of SQLi Attack

When attacker or normal user logs-in with name and password, it goes to a web server and then the database. The type of query going into the database can be observed, and when the attacker uses '1 or 1=1 --, it means the attacker will get information of the whole table. Therefore, sanitizing or cleaning the user input is very important in any web application. Any input that comes from the user side should never be trusted because it's origins, normal user or attacker, is unknown.

2. BACKGROUND

Main problem is SQL injection and to stop this type of attack a solution needs to be found. In this paper, machine learning is used as a solution for SQLi problems.

2.1 PROBLEM DESCRIPTION

Two solutions already exist, validation and/or sanitization. Now the problem is, how can the user input be validated? For example, "my name is urmi, and I am a student"; does this sentence contain any SQL related words? The answer is "NO." This means there is safety, and there is no SQL injection possible. But, if the user input contains sentences "1 or 1=1--" and drop table name where id = '12' " , then it can be said, alert there is a SQL injection. These types of sentences need to be blocked that come from the user's side.

2.2 ML APPROACH

The first step is to identify the SQL words and phrase, after which those sentences can be stopped. SQL words can be any such as: "select", "drop", "merge", "delete" or any numbers and signs. To identify this type of SQL words, machine learning models proved very helpful.

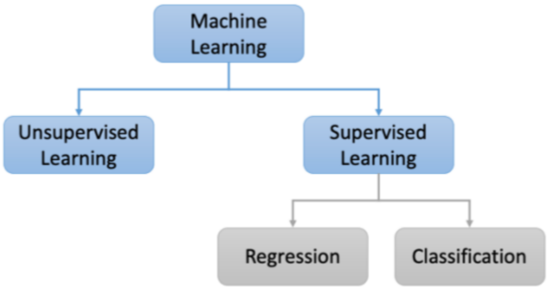


Figure 2: ML Techniques

Machine learning has two main techniques, one is supervised learning and the second is unsupervised learning. Supervised learning is defined by its use of labeled datasets to train the algorithms that helps to predict the outcomes accurately. This technique is further divided into two parts, classification, and regression. Classification techniques are used to detect fraud and spam emails, whereas regression techniques are used for risk assessment and score prediction.

3. APPROACH

As a machine learning process starts, the first step is to select the whole dataset and preprocess them. Preprocessing is the process of manipulating the data. In this project, preprocessing covers deletion of NULL values as well as removing duplicate values from the dataset.

In the next step, the dataset is divided into two parts for training and testing. Next step is all about selecting specific models for the defined problems and vectorization of the inputted data. The model training phase will now start and afterwards, the testing process will be done for the test dataset. Applying test data to the trained model gives the result pertaining to how accurate our models were trained. Below figure 3 shows the whole process of machine learning approach.

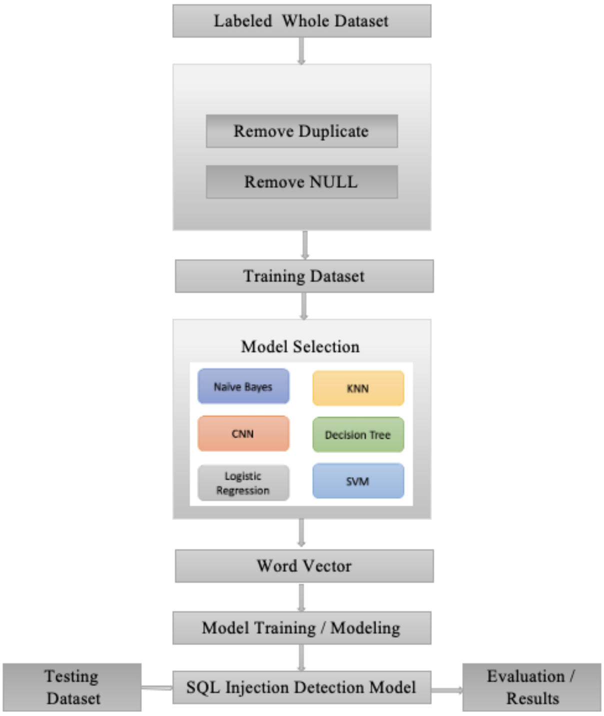
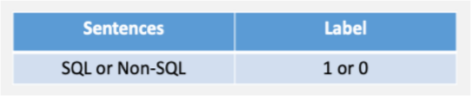


Figure 3: Project Implementation Process

3.1 DATASET DESCRIPTION

Dataset would be downloaded from the Kaggle website [3]. Downloaded dataset is a labeled dataset containing 4200 rows and two columns. First column is for sentences and the second for labels.



Sentences	Label
SQL or Non-SQL	1 or 0

Mainly two types of sentences are there in the dataset, SQL and non-SQL, where providers use label 1 for SQL and label 0 for non-SQL sentences. Now that a clearer view of the dataset has been achieved, labelled data is easier to single out, so supervised learning method can be used for this problem.

3.2 PREPROCESS DATA

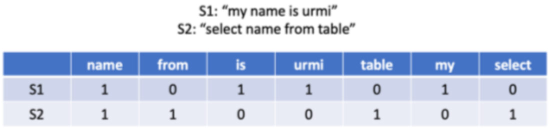
There was a total of 17 NULL values in the dataset. These NULL values negatively affect the performance and accuracy of any machine learning algorithm. It is essential to remove NULL values from the dataset before applying any machine learning algorithm to the dataset. Moreover, there was five duplicate data available in the dataset. Removing duplicates is very important to get accurate result because the same entry should not be counted multiple times.

3.3 VECTORIZATION

For this project, the count vectorizer method was used from scikit-learn library, and it converts a collection of text data to a matrix of token counts. The words need to be encoded as integer values, for use as inputs in machine learning algorithms. This process is also called feature extraction. When the count vectorizer method is called, it takes the bag of words approach, where each sentence inside the document is separated into tokens. For example, if the sentence is “my name is urmi” then the first token would be my, second token would be name and so on. Count vectorizer counts the number of times each token occurs in a dataset. For example, in the dataset my token occurs 200 times.

Additionally, fit_transform function was used to transform the sentences into numbers and store it into a matrix like format. Below, table 1 shows the example for how the vectorized data will look after the conversion. When the table shows 1, it means this word is occurring in a specific sentence. Furthermore, this type of output is used as training data.

Table 1: Sample Output of Count Vectorizer



S1: “my name is urmi”
S2: “select name from table”

	name	from	is	urmi	table	my	select
S1	1	0	1	1	0	1	0
S2	1	1	0	0	1	0	1

3.4 DATASET SPILT

To train the model, the dataset needs to be split into two parts, training, and testing. Splitting the dataset is beneficial because it helps to prevents overfitting. For this project, the dataset is divided into a 80:20 ratio, which means 80% of the data is used to train the model and 20% of the data is used to test the model.

3.5 SUPERVISED LEARNING METHOD

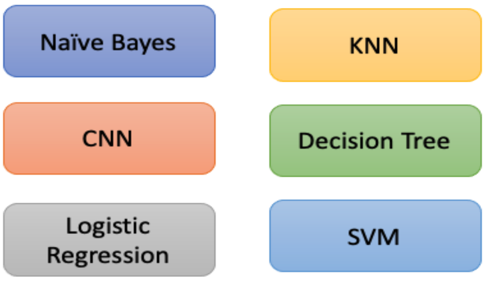


Figure 4: ML Models

There were various supervised learning methods available, but the top five of them will be used. A neural network (artificial neural network) is a type of machine learning model that is often used in supervised learning when the desired output is already known. CNN is a convolutional neural network that has one or more convolutional layers and are used mainly for image processing, classification, and segmentation. Nowadays, CNN is widely used by many papers, and it has proved to be very helpful in cases of classification problems. One of the papers [4] proved, CNN is the best model, and it gives highest accuracy with regards to classification problems. On the other hand, results are totally dependent upon the dataset and its type.

Naïve Bayes:

It is a supervised learning algorithm that used Bayes theorem to solve classification problems. It is one of the simplest and most effective algorithms which helps to make quick predictions. It is a kind of probabilistic model, which means it predicts the output result based on the probability of a data. Firstly, the model converts the given dataset into frequency tables and generates another table by finding the probabilities of the given features. Next, the model uses the Bayes theorem to calculate the posterior probability and then predict the output. In the end, the result will predict whether the given data is regular data or SQL injection.

KNN:

KNN stands for K-Nearest Neighbour, which is used to assume the similarity between the new data and available data. The new data can be put into a category that is most likely match. In short, finding the nearest neighbour with the same category. KNN algorithm stores all the available data and classifies a new data point based on the their similarity.

This algorithm can be used for both regression and classification problems but is mostly used for classification. Firstly, select the number K of the neighbors and then calculate the Euclidean distance of neighbors. Next step is to take the K nearest neighbors as per the calculated Euclidean distance and count the number of the data points in each category. Furthermore, the new data points are assigned to that category for which the number of the neighbor is maximum. In the end, the test data is applied to the model to get optimal results.

Decision Tree:

It is called a decision tree because, it looks like a tree structure, ands starts with the root node, which further divided into branches and constructs a tree-like structure. The data is continuously split according to a certain parameter in this algorithm.

The tree has mainly two units, nodes and leaves. The leaves are the decisions or the final outcomes where the splitable data are called decision nodes. A decision tree simply asks a question and based on the answer whether it is yes or no, it further splits the tree into subtrees. A decision tree can contain categorical data as well as numerical data.

SVM:

The goal of the SVM algorithm is to create the best line, also known as a decision boundary, that can separate n-dimensional space into classes so that new data points can easily be plotted in the correct category. The decision boundary is called a hyperplane. Moreover, the algorithm itself chooses the vectors that helps in creating the hyperplane which is known as support vectors.

Logistic Regression:

This algorithm is used to predict the categorical dependent variable using a given set of independent variables. Therefore, the result must be a categorical value such as Yes or No, 0 or 1, etc., but instead of giving the exact value it gives the probabilistic values which lie between 0 and 1. In Logistic regression, instead of fitting a regression line, it forms an S-shaped curve which predicts two maximum values (0 or 1) and this is called the Sigmoid function or logistic function. The sigmoid function is a mathematical type of function used to map the predicted values to probabilities. Moreover, the used threshold value defines the probability of either 0 or 1.

CNN:

A Convolutional Neural Network is a one type of feedforward neural network where the convolutional layer is a very important part of the network. The purpose of the convolution operation is to extract different features of the input. The convolutional layers contain 64 and 128 filters of size 3x3. This is followed by a max pooling layer of pool size 2x2. The end of the model consists of two dense or fully connected layers containing 64 and 128 neurons each which act as hidden layers. All trainable layers use the ReLU activation function. The final layer is a sigmoid layer which outputs the probabilities.

Overall, the CNN model includes several convolutional layers for convolution calculation of data and pooling layers which can compress the number of data and parameters, reduce over-fitting, and improve the fault tolerance of the model. A fully connected layer is used to connect all the features and send the output value to the next layer. Hidden layers are also used to make the final processing before passing the output value to the classifier, to reduce the possibility of data over- fitting.

4 ANALYSIS

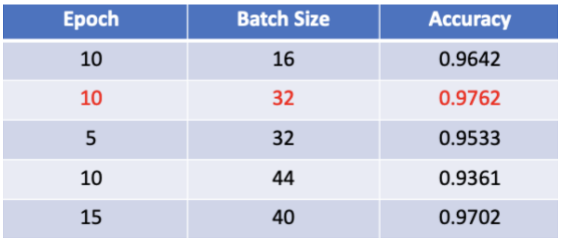
Performance enhancing strategy:

Hyper parameter tuning is very important because it controls the behavior of the training algorithm. The main goal behind this tuning is to find an ideal combination of hyperparameters that minimizes a predefined loss function to give better results as an output.

In this project, Naïve bayes and CNN both gave similar results. The idea behind this tuning strategy is to get more accurate results compared to Naïve bayes. Below, table 2 shows the combination of two parameters, one is epoch and second is number of batch size in network. After applying five various combinations, one of them proved best and is considered for further processes.

One combination takes approximately 10-15 min to run, hence it is time consuming, but the result is pleasing.

Table 2: Hyper Parameter Tuning for CNN



Epoch	Batch Size	Accuracy
10	16	0.9642
10	32	0.9762
5	32	0.9533
10	44	0.9361
15	40	0.9702

Training graph of CNN model:

The combination of epoch 10 and batch size 32 proved effective, so we analyze the detailed training graph for the CNN model with new combination of parameters.

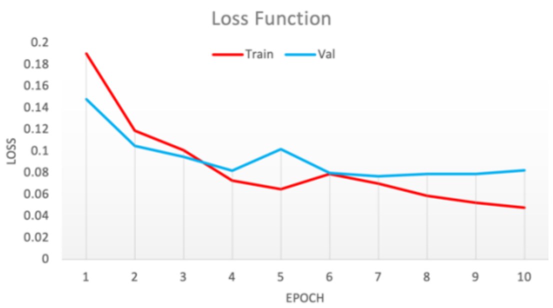


Figure 5: Loss Graph During Training Process of CNN

The above graph shows, loss function for the CNN model during the model training process. It gives an idea of the training process and the direction in which the network learns. Each row consists of a loss and accuracy value and for each epoch, an average of those rows. The gap between training loss and validation loss is not too large, so it can be said that the model trained in a good way.

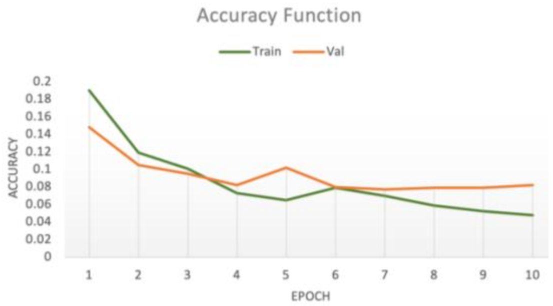


Figure 6: Accuracy Graph During Training Process of CNN

The above graph shows, accuracy function for the CNN model during the model training process. Accuracy of validation set is higher than accuracy of training in most of the epoch. The gap observed between the two lines is not too big, so overall the model is trained well.

Evaluation Metrics:

The model is evaluated based on its predictions on the test set. The failure analysis is performed on the Precision and Recall values. They are given by Equations 1 and 2, respectively.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

To evaluate model performance, the F1Score was primarily used, given by Equation 3.

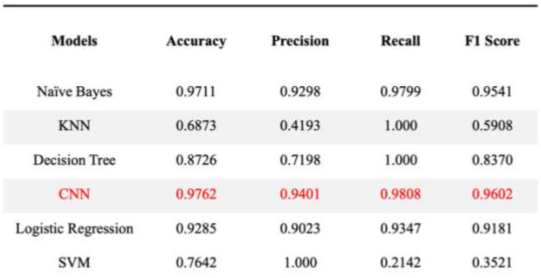
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

The accuracy of the model is also used to check the overall performance of the model on the test set.

5 EXPERIMENTAL RESULTS

Below, table 3 shows the results of Accuracy, Precision, Recall and F1-score of all models. It can be clearly observed that the CNN model performs best on all fronts.

Table 3: Results of all used models



Models	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	0.9711	0.9298	0.9799	0.9541
KNN	0.6873	0.4193	1.000	0.5908
Decision Tree	0.8726	0.7198	1.000	0.8370
CNN	0.9762	0.9401	0.9808	0.9602
Logistic Regression	0.9285	0.9023	0.9347	0.9181
SVM	0.7642	1.000	0.2142	0.3521

The below, figure 7 provides a visual representation showing the accuracy of all six models used for this project. It is observed that Naïve bayes and CNN gave almost identical accuracy. Hence, CNN proved to be better. After all the model performances and analysis, one small experiment would be done as a testing. The saved CNN model used for this experiment.

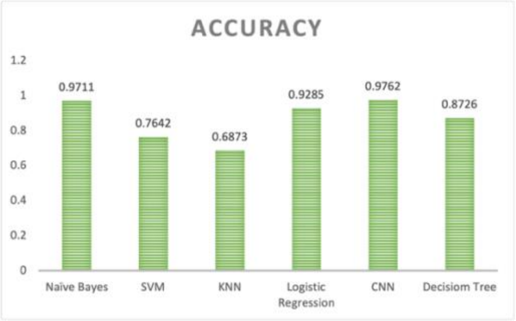
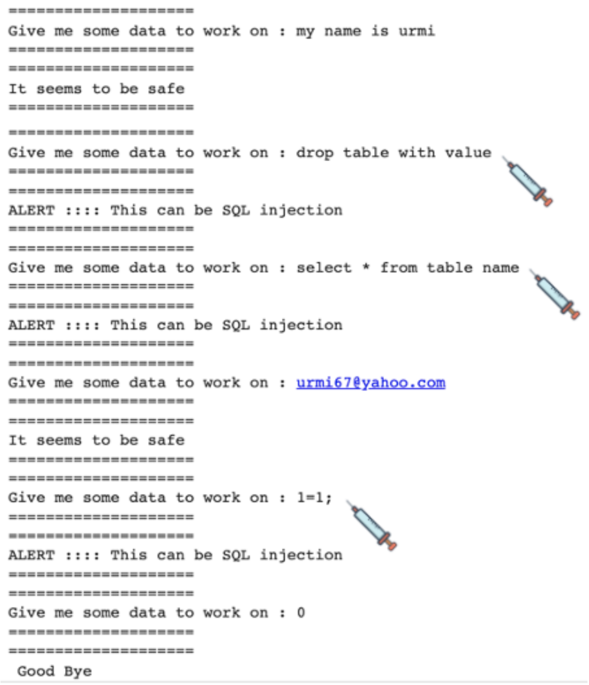


Figure 7: Accuracy Graph of all Used Models

The below snapshot gives an idea about how the model works. For example, when a user gives input such as “my name is urmi” or an email like “urmi67@yahoo.com”, the model predicts that this data seems safe. On the other hand, when data like “1=1;” or “drop table with value” is entered as an input to the system, the model predicts that the given data may contain SQL injection. This experiment shows, how accurately our model can predict the sentences whether it is SQL injection or not.



```
====================
Give me some data to work on : my name is urmi
====================
====================
It seems to be safe
====================

====================
Give me some data to work on : drop table with value 💉
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : select * from table name 💉
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : urmi67@yahoo.com
====================
====================
It seems to be safe
====================
====================
Give me some data to work on : 1=1; 💉
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : 0
====================
====================
 Good Bye
```

Figure 8: Snapshot of Experimental Result

6 CONCLUSION

A total six models were performed for this project and analysis of the best model with greater detail. Convolutional neural network proved best among all the models. Hyper parameter tuning also performed to enhance the performance of the model. Small experimental task also performed using CNN model. In the end, the result of the experiment proved that the trained model gave perfect results. However, the accuracy and model may vary as the dataset is changed. The model can detect SQL injection from the user input.

Future work for this project can include more complex algorithms and neural networks with various parameter tuning. Moreover, NLP based Bert model can be tried as a language model because nowadays it is very popular.

Some challenges that were faced, included slightly different loss and accuracy results each time the model was trained or ran because of its nature of shuffling data each time. Another is hyper parameter tuning for the neural network which was really time consuming.

7 REFERENCES

[1] "OWASP," 2021. [Online]. Available: https://owasp.org/www-project-top-ten/.

[2] T. B. a. N. Niv, "Web application attack report," 2013.

[3] "Kaggle," 2020. [Online]. Available: https://www.kaggle.com/syedsaqlainhussain/sql-injection-dataset.

[4] W. H. W. F. Ao Luo, "A CNN-based Approach to the Detection of SQL Injection Attacks," *IEEE,* 2019.

[5] K. Zhang, "A Machine Learning based Approach to Identify SQL Injection Vulnerabilities," *International Conference on Automated Software Engineering (ASE),* 2019.

[6] R. A. E. O. J. A. I. I. R. G. J. Morufu Olalere, "A Naïve Bayes Based Pattern Recognition Model for Detection and Categorization of Structured Query Language Injection Attack," *International Journal of Cyber-Security and Digital Forensics,* 2018.

[7] Q. Y. C. W. J. Z. Ding Chen, "SQL Injection Attack Detection and Prevention Techniques Using Deep Learning," *Journal of Physics: Conference Series,* 2021.

[8] H. ,. ,. ,. ,. TareekPattewar, "Detection of SQL Injection using Machine Learning: A Survey," *International Research Journal of Engineering and Technology (IRJET),* 2019.