Journal of Theoretical and Applied Information Technology
15th August 2022. Vol.100. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DETECTION OF SQL INJECTION ATTACKS USING MACHINE LEARNING IN CLOUD COMPUTING PLATFORM

JAMILAH M ALKHATHAMI[1], SABAH M. ALZAHRANI[2]
Taif University, College of Computers and Information Technology, Department of Information Technology, Taif, Saudi Arabia[1]
Taif University, College of Computers and Information Technology, Department of Computer Science , Taif, Saudi Arabia[2]
E-mail: j.ojo2010@hotmail.com, sa.sabah@tu.edu.sa

ABSTRACT

The cloud computing holds a massive quantity of private and secret data and information, It uses the internet to communicate with another party so it contain a number of unreliable strings that can be demonstrated to be loopholes, therefore protecting data stored on the cloud is often a serious risk. one of the most serious security dangers to cloud are SQL Injection Attacks (SQLIA), It sends a susceptible query to affect server systems, allowing attackers to gain illegal access to databases, resulting in identity theft and security breaches. In this research we have studied and investigated the recent published approaches to SQL injection attacks detection, and we have presented SQL injection detection tool based on machine learning, The model classifies SQL injection queries into two categories: attack and legitimate. The model is training with four machine learning algorithms We have employed K-Nearest Neighbors (KNN), Multinomial Naive Bayes (MNB), Decision Tree (DT), and Support Vector Machine (SVM), after performing data preprocessing and feature extraction, we compared the values obtained by each model to determine the best model in SQL injection detection. The result show SVM produced the best results with an accuracy of 99.42%. after that The decision tree algorithm obtained results with an accuracy of 99.4%. Then MNB algorithm produced a 97.09%. However KNN produced the worst results, with an accuracy of 92.45 %. Therefore we have found that our models produce near-perfect results.
Keywords: *Cloud Computing, SQL Injection Attacks, SQL Queries, Detection, Machine Learning Algorithm.*

1. INTRODUCTION

Cloud Computing (CC) is a novel framework for coordinating and delivering services via the Internet that has recently emerged. Common financial constraints and rising computing costs necessitate data storage, processing, and presentation, which has necessitated significant changes to the current cloud architecture. CC refers to the on-demand availability of end-user resources, particularly data storage and processing power, with no need for client's active involvement. The term "distributed computing" is a catchphrase that means distinct things to distinct individuals, Distributed computing connects clients' both public and private data on a single server throughout the Internet. However, there are significant security issues with CC that are preventing wider implementation of the computing paradigm, such as client and association vulnerability [1] [2].Web apps [3], which provide client services and are frequently related to critical information housed backend databases, transfer a lot of information on a daily basis. The world is increasingly dependent on online applications [4], and as a result, providing security for these apps is critical [5][6], The necessity of securing a network, devices, and the privacy of personal and confidential information is a national concern [6]. In most apps, data is stored in backend databases[7], These backend databases are sometimes subject to destructive attack like SQL Injection Attacks . One of the most serious security threats to web apps and services is SQLIA, It launches a susceptible query that destroys linked server systems and allows attackers to access databases without permission, It gives you the power to remove, change, and restore valuable and personal data from databases [8]. The figure below shows a simplified scenario for the occurrence of this attack:

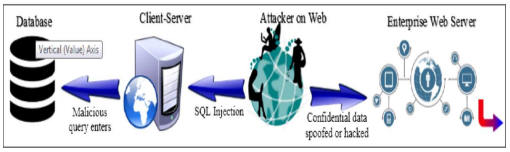

Figure 1: scenario of SQL injection attacks

Although SQL injection attacks have been around for a while, they still impose a severe threat [9]. Almost all web applications (98%) are vulnerable to numerous attacks, among them is SQL Injection which is classified by the Open Web Application Security Project (OWASP) as the most severe threat among the top ten vulnerabilities for the past fifteen years [10][11][12].

It is classified among the most serious SaaS vulnerabilities, which enables attackers to compromise the availability, confidentiality, and integrity of customer data, As a result, implementing robust data security protection mechanisms is a must that cloud computing manufacturers and providers must act upon While establishing the infrastructures of their cloud[13].

SQL Injection action occurs when a hacker attempts to input SQL code patterns into the application's input fields [14], Numerous tools and techniques have been designed and implemented to trace SQL injection attacks, Conventional approaches rely heavily on statistics, information, classification, and clustering [15].

SQL injection is a technique that uses an entry node to inject a malicious script into a susceptible internet app, The back-end database is then bypassed, The web application is then forced to generate database discoveries using queries that should never be run on a regular basis, An attacker can acquire access to all database data by bypassing the web application's authentication and authorization, This exploit not only provides unlimited access, and yet it can also be used to harm data integrity by inserting, removing, or changing records. SQL injection attacks are intended to find flaws in a web application's defenses, such as when input data is not adequately validated or filtered, or when it is written carelessly and executed unexpectedly, It also occurs when the code, or programming language, has faults, It's a vector that can attack both applications and SQL databases, Unauthorized access to the underlying data, structure, and database management system is possible [12]. SQL assaults are divided into ten categories, each of which is briefly detailed below:

A. Tautologies

The purpose of the tautology injection attack is to circumvent authentication and gain accessibility to the database without possessing a legitimate identity, Additionally, identifying injectable parameters, which means locating an injection point for a SQL injection attack before extracting the data, The basic command for the Tautologies attack is run as follows: SELECT * FROM users WHERE id=100OR1=1; This expression evaluates the query as TRUE, returning all the data in the table (users), that implies the hacker will circumvent the user credentials, gain entry to the table, and obtain the data via this injection [16].

```
SELECT * FROM USER
WHERE Name=' admin ' AND
Password=' ' OR 1='1 ' --
```

Figure 2: bypassing password [17]

B. Piggy-Backed Query

This kind is utilized to combine an additional query with the initial query, As a result of this fraud, the targeted database gets flooded with queries to execute. Consequently, the attacker may issue commands to remove, edit, or insert. The far more harmful code of this kind is one which deletes the contents table [18].

C. Union Query

The hacker uses the UNION operator to attach a malicious query to the first query in a union query attack, The fraudulent query's results will be used to connect the results of the first query, allowing the attacker to obtain the contents of columns in additional tables [19].

D. Stored Procedures

This approach employs malicious SQL code to perform integrated built-in functions that increase privilege, ensure service denial, or allow remotely execution. Actually, the majority of database vendors produce database systems that include standard stored procedures and features that improve database performance and integrate it with the computer system. As a result, once a hacker is aware of the backend database, SQLIAs can be generated to execute scripts on it [12].

E. Illegal/Logically Incorrect Queries

The purpose of this type of SQL exploit is retrieving data as well as identifying and retrieving injectable parameters, Owing to its capacity to determine the database's structure and

injectable parameters, it is frequently utilized as the initial stage in many hacks. Basically, it occurs when the application server delivers the default error page, that includes details about vulnerabilities, Also, it assists programmers in correcting their programmers, It may also display the list of tables and columns. Eventually, attackers can target the database weather by extracting injectable code or data from error pages [20].

F. Inference

This sort of attack generates queries that alter the behavior of an application or database dependent on the query's output, An attacker can use these ways to acquire data from a database and figure out which settings are vulnerable. Inference-based attacks include the following: Blind injection and timing attacks are two types of attacks [12].

G. Alternate Coding

This kind is used to evade detection, The attacker injects scripts into the system to evade detection mechanisms using signatures such as Char (), EXEC (),BIN (), ASCII (),HEX (), UNHEX (), DEC (), BASE64 (), ROT13 (), and so on, This accomplished through the verification of user input data and the banning of meta-characters [21].

H. Blind Injection

An attacker executes queries that return a Boolean value, If the response is yes, the program works properly; if the answer is false, an error occurs, As a result, the attacker can obtain an indirect reply from the database[22].

I. Timings Attacks

It's used to extract data using (If-Then) expressions in which the assailant logs database response time delays, Based on technique of data transfer for incoming and outgoing data, SQL assaults are divided into three types, These are the three categories: inferential, in-band, and out-of-band. The attacker obtains data from the precise channel used to send the query or carry out the operation in an in-band SQL attack. In an inferential SQL injection attack, the attacker does not use any channels to retrieve data, instead relying on other attacks to assess the web application's behavior [12].

J. End of Line Commend

This statement registers the attacker as an administrator account.

SELECT * FROM Accounts WHERE account Name ='admin' AND password =' [12].

1.1 Machine Learning

Recently, there has been much debate about using machine learning algorithms to identify and mitigate different cyber security threats [23], Machine learning is a computer science field that includes the research and development of methods that allow computers to self-study based on incoming data to solve particular issues [24], There are three types of machine learning methods: supervised, unsupervised, and semi supervised learning [23], In its most basic form, supervised learning is a kind of machine learning in which labeled training data is used to train the classifier [23], The machine learns based on the input patterns to build the classifier, which will then be used to estimate classifications for additional data. Unsupervised learning, on the other hand, makes use of unlabeled training data, In this scenario, the machine learns to construct the classifier by studying the data's features. Semi-supervised learning is a technique that combines the benefits of both supervised and unsupervised learning, The input training dataset for semi-supervised learning includes labeled and unlabeled data, Each technique has its own set of benefits and drawbacks, as well as its own application domain [24].

we got that The Support Vector Machine algorithms produced the best accuracy with 99.42%, Decision Tree algorithm with 99.4% accuracy and the accuracy of the other two algorithms we used, MNB and KNN algorithms were 97.09% and 92.45%, respectively. Finally, we find that our models yield near-perfect results with a very low error rate.

2. LITERATURE REVIEW

SQL injection has been the subject of numerous research, and it is detected using a variety of techniques such as static and dynamic analysis, combination techniques, machine learning, deep learning, and so on, in this chapter we will explore several of these studies in this field.

Singh, N, et al. (2019) describe a hybrid technique that has a higher information value than other systems, The technique focuses on static, dynamic, and runtime detection and prevention mechanisms by using a demonstration design, It also classifies various harmful queries and ensures that the system is correctly configured for a secure operating environment. The associativity formula was utilized and the experimental implementation yielded a probability of success of 0.775, resulting in a long-term comparative solution that has been used to date. The experimental computation performance

of the suggested Valid Security tool justifies the formulation in this study with a 70-80% rate of success in dataset protection from 'SQLI' injected by hackers from outside of the cloud, who focus on getting more information effectiveness than other approaches. The method increases the firewall's security by detecting and preventing attacks, resulting in an IUR range of 80-90 percent, which is considered a balanced range in terms of several technological elements [8].

Abikoye, O, et al (2020) used the Knuth-Morris-Pratt string matching method to show how to identify and avoid these risks, The technique (KMP) was used to identify any malicious code by comparing the user's input string to the injection string stored pattern. In order to complete the project, the PHP programming language and the Apache XAMPP server were used, Several test of XSS , SQL injection, and encoded injection attacks were used to evaluate the technique's security. The five stages of the methodology used in this study are: creating a SQL injection string pattern, designing a parse tree for different types of attacks, detecting SQL injection and cross-site scripting attacks, and applying the KMP algorithm to block SQL injection and XSS attacks. The findings revealed that the proposed technique was capable of identifying and preventing assaults, as well as logging the attack record in a database, blocking the system based on its mac address, and creating a warning message. As a result, the suggested method was most efficient in detecting and preventing SQL injection and cross site scripting attacks[25].

Samarin, S et al. (2019) proposed an unique anomaly-based solution for SQLIA detection and/or prevention that does not require any changes to the source code of vulnerable apps (implemented as a proxy between the application server and its database server), During the detection phase, lexical analysis of the queries is used to identify the majority of assaults that result in a change in the syntax of application requests. Other kinds of attacks, such as second-order attacks and attacks that cause data type mismatch issues are avoided during the preventive step, in which each query is independently changed to a structured query (prior to getting submitted to the database) using a lexical analysis. (There are three stages to this method: learning, detection, and prevention). In the learning phase, the profiles of legal queries are generated by the lexical analysis and syntactic structure of them. These profiles are used to detect attacks in the detection phase and to create parameterized queries in the prevention phase. Executing parameterized queries (instead of the raw ones) prevents completing the attacks that do not directly change the structure of the queries, such as the second-order attacks. In preparing the parameterized queries, they can also easily detect the malicious queries resulting in data-type mismatch errors without needing to execute the queries). The main contribution of this paper is proposing an automated method to create parameterized queries dynamically by semantically analyzing the raw queries sent from a vulnerable application. By creating parameterized queries, they could prevent second-order and unknown attacks, and detect/prevent attacks causing data-type mismatch errors without the need to execute the query. In the proposed method, they did not need to change the application source code. Evaluation of the proposed approach in practice showed that the performance overhead imposed by this method was minimal and, in some cases, it even led to improvement in the application's performance by parameterizing the queries [26].

Durai, K et al. (2021) employed an ontology-based methodology for detecting and preventing SQLIA through ontology (SQLIO), which includes an ontology generation and vulnerability prediction rule-based method. The suggested approach avoids and identifies SQLIA online vulnerabilities to a greater extent in the cloud. Ontology emphasizes vulnerabilities, threats, and regulations that aid in the more efficient projection of complex attacks. Using SWRL principles and surmising processes, the suggested technique anticipates potential attack flaws. The deduction motor method ensures a rundown of probable attacks by taking into consideration Network vulnerability information. These attacks were made in conjunction with the security goals, In addition, the suggested model offers recommendations for effectively limiting counter-attack actions, The information obtained is critical for analysts and designers to manage threats by putting together a well-thought-out secure application procedure. The proposed ontology framework could be used to detect security flaws at the testing level in the future. For tracking SQLI assaults on web pages, the proposed approach is crucial [27].

Cahyadi, N et al. (2021) The approach proposed in this implementation is to generate detectors using transfer learning techniques. The development process is carried out by preparing the dataset used; because the model will function as a filter, it is necessary to prepare data balanced between normal SQL queries and injection SQL queries. Then the dataset is obtained in such a way as the need so that

it can be included in the model to be trained. The model used in this study is SWIVEL. SWIVEL is a model used in embedding text, especially the identification of an input string. The word embedding neural network used is a text embedding model from TensorFlow called gnews swivel 20 dims with oov. After that, it is necessary to configure hyper parameters such as adding batch normalization or regularization in the model to be trained. The results obtained from this system achieve a string detection accuracy that is rated as SQL injection of 99.77%, with a loss under 0.0218. Although the results obtained in this implementation can exceed the state-of-the-art SQL injection detector from previous studies, it is necessary to note that statistically, there should be overfitting due to the unbalanced dataset used. The main challenge in carrying out the lesson for the SQL injection detector is setting up the dataset. The variety of distributed formats (although still limited) and the different training data structures between one researcher and another, make the SQL injection detector development need to be further developed in the proper time frame. However, as a starting point for research on SQL injection detectors, the results obtained are sufficient to meet the requirements, where the final results can be used as initial expectations as a potential topic in general [28].

Mishra, S et al. (2019) classified and detected SQL Injection attacks using a method called Gradient Boosting Classifier from ensemble machine learning techniques. Incoming traffic is classified as either SQL Injection or plain text using a classification algorithm, The challenge is solved using two machine learning classification techniques: the Nave Bayes Classifier and the Gradient Boosting Classifier. The NB algorithm produces results that are 92.8 percent accurate. Because they use multiple fundamental classifiers to reduce error and improve accuracy, ensemble learning algorithms are thought to produce more accurate results. As a result, the Gradient Boosting Classifier was chosen to solve the SQL Injection classification problem, with a 97.4 percent accuracy. This study shows that ML algorithms may be used to identify SQL Injection, and that the Gradient Boosting classifier algorithm outperforms the Nave Bayes model, The efficiency and usability of this project may be improved in the future. Other SQL Injection detection methods, like web application firewalls and static code analysis, may be implemented in conjunction with the machine learning methods for identifying SQL Injections. Additionally, the machine learning techniques may be further modified via improved feature extraction, Tokenization is applied in this project to generate features for the machine learning algorithms, Other methods may be employed to extract features and train the model more effectively [29].

Kranthikumar, B et al. (2020) compare machine learning classifications such as SVM (Support Vector Machine), Gradient Boosting Algorithm, and Naive Bayes classifier to detect SQL injection threats using pattern-based classification called REGEX, By performing classification on collected dataset with 20474 queries and it is shown that the REGEX classifier gives 97% accuracy along with 3.98 sec computation time in execution which more efficient than above machine learning techniques. This paper is a comparative study of SQL injection detection. Three machine learning techniques, namely SVM, Naive Bayes, Gradient boosting Algorithm, and another REGEX classifier was implemented and tested. Experiments showed that REGEX method took 3.98 sec computational time to detect SQL injection attack using pattern scanning method. REGEX gives a very good performance with 97% accuracy and also 100% precision From the another algorithms, they observed that REGEX performance is better than other classifiers with respect to comparison metrics as following : Computation time is very less; hence it is faster, accuracy is more so it is less erroneous, The higher precision rate indicates positive prediction rate and Sensitivity is high, so the ability to detect injected queries is more [30].

Latchoumi, T et al. (2020) studied and trained the SVM algorithm on all possible malicious expressions before generating the model. In order to forecast if a particular query includes malicious expressions or not, SVM is applied to the model when a query is provided by the user. If the user creates a new method, SVM can identify any malicious expression by matching it with a small set of syntax. These researchers also demonstrated predictive models with good performance, as evidenced by the mixing matrix and ROC chart, which can be used to identify and avoid SQLIA in massive data, the technique provided here operates in a huge quantity of data, which is insufficient, according to SQLIA's current information. A multiclass classifier will be used in future research to describe and classify different types of SQLIA [7].

The goal of Farooq, U (2021) was to identify this attack by tokenizing the queries and then implementing the algorithms to the tokens. They

employed AdaBoost, Gradient Boosting Machine (GBM), Light Gradient Boosting Machine (LGBM), and Extended Gradient Boosting Machine (XGBM) as Ensemble Machine Learning models. Their models provide near-perfect outcomes, with a rate of error that is practically insignificant. With 0.993371 accuracy, 0.993373 precision, 0.993371 recall, and 0.993370 f1, LGBM produces the best results. The LGBM also produced a lower error rate, with the RMSE and FPR being 0.007 and 0.120761, respectively. AdaBoost produces the poorest results, with 0.991098 accuracy, 0.990733 precision, 0.989175 recall, and 0.989942 f1. AdaBoost also generated a high FPR of 0.009 [12].

Zheng, J et al. (2021) In order to defend against noise-exploiting attacks on privacy-preserving systems, this work introduces a new technique for detecting malicious queries, in which The original framework has been updated with a filtering layer that inhibits the execution of risky and malicious query expressions. They use the association rule mining method to attack patterns in order to greatly reduce the risk of personal identification. For model testing and training, four machine learning classifiers are often used to the training data to avoid fitting. All other classifiers are outperformed by the random forest classifier, which has an accuracy of 0.91, In this study, the malicious SQL query patterns are exploited, but they can be immediately applied to existing noise exploitation and associated anonymization system threats. Numerous new attack types, as well as some new features, may emerge. In this case, the detecting method is fairly limited. Their method, on the other hand, can quickly exploit new attack features and update the existing model, making it flexible to a variety of application scenarios [15].

Chen, D et al. (2021) proposed a SQL injection detection technique that doesn't dependent on background rules and instead uses a deep learning system and a natural language processing approach, based on considerable local and worldwide research, This method could improve accuracy while lowering false alarm rates. In this study, they employed multilayer perceptron (MLP) and convolutional neural network (CNN). The model's accuracy is confirmed by testing the set of data after it has been trained, and the training process is exhibited using the tensor flow framework board feature to examine the model's ease of use, Experiments show that the approach is more accurate and efficient at detecting first-order SQL injection attacks, The following research will concentrate on modern SQL injection attack techniques like hybrid attacks and second order injection. [31].

In the paper [32], SQLIA was classified into 18 predictive analytic classes by Idowu, S et al (2020) in order to identify and prohibit suspicious database access in order to fraudulently get confidential information. The creation of diverse attack techniques is aided by the research of SQLIA protection and detection via taxonomy and classification into classes according on their importance. As a result, the capacity of attackers to create new injection sites into the database is hampered. Database Fingerprinting, Time-Based Error, Buffer Overflow, Stored Procedure, Deep Blind, Second Order, Alternate Encoding, Out of Band, Union, Double-Blind, Conditional Error, Illegal / Invalid / Logical Incorrect, Conditional Response, Error Based (blind), Piggyback, Database Mapping, Literal, and Tautology are among the 18 SQLIA types, with Benign referring to the non-malicious class. The taxonomies were used to explore the various attack strategies in order to produce a viable SQL injection attack prevention solution for all taxonomies discovered. All attack methods used against online applications to steal database secret information necessitate the classification of SQLIA into eighteen types. In order to train the emerging web application security categorization model in real-world scenarios, data must be learned. The development of data of the predicted legal and illegal input query signature strings, establishing SQLIA signatures using SQLIA knowledge shown in the taxonomy, and developing SQL token pattern that includes delimiter identifiers, constants symbols, and query keywords has been a significant barrier to SQLIA studies, and this has been addressed in this work using the following strategies: Determining the type of SQLIA is critical for effective prevention and detection in predictive analytics [32].

Lin, P et al. (2020) investigated the GreenSQL database firewall workflow and operation mode, Based on a study of the characteristics and SQL injection attack command patterns, the GreenSQL learning input model is improved by creating input patterns and optimizing the whitelist. They can, on the other hand, use the GreenSQL open-source database firewall's multiple operating modes to generate a large number of random input data with a specified distribution and analyze when there is a potential attack procedure from of the IPS intercepted samples to advance their research on SQL injection attack, Modular attack input is achieved in this study by studying each GreenSQL

database working mode characteristic in conjunction with common SQL attack means, and establishing a set of abstractions for seven attack kinds, all in accordance with the combination rule and guidelines of these seven sets. When compared to the traditional random input command, the input efficiency is improved, GreenSQL's learning performance may be significantly improved, and its IPS intercepting samples are more plentiful, all of which contribute to GreenSQL's adoption. The research technique may improve GreenSQL learning performance and intercept samples in IPS mode, ensuring database security in the background [33].

Arock, M et al. (2021) concentrated on extracting SQL injection patterns using existing tagging and parsing approaches. Multi-layer Perceptron is used to model and train pattern-based tags, which surpasses conventional query classification algorithms by 94.4 percent. The goal of this research was to find tautology-based SQL injection, a more advanced SQL injection technique. The conditional operator "OR" is used in this attachment to add malicious commands into the SQL queries "WHERE" clause, and to ensure that the query returns "TRUE" frequently, the initial condition is turned into a tautology, Its goal is to retrieve all database records without the need for user verification, Current methods focus on extracting letters, special symbols, and other information from queries, and the amount of keywords, and then classifying the injected requests based on those qualities. It is, however, limited to the symbols, operators, and keywords they offer. On the other hand, this pattern-based technique successfully classifies and detects the injection, but Due to the lack of a direct dataset, the model is trained and tested with a smaller number of human-labeled legitimate and injected queries, yielding a 94.4 percent accuracy rate. The study's major flaw is that it solely uses tautologies to diagnose SQLIA [34].

In Table 1, we summarized and compared some studies of SQL attack detection using machine language in terms of methodology and results.

Table.1. Summarizes the related works in our field

Author	Objective	Approach	Tools	Results
Kranthi kumar, B et al. [30]	detection of SQL injection attack using pattern based classification called REGEX. -To improve the efficiency of the system. –To improve the accuracy (True Positive and True Negative).	-The REGEX is a pattern based Classifier. -The proposed model and the Machine Learning models were implemented using Python, and tested using the synthesized dataset obtained from GITHUB. -compared SVM (Support Vector Machine), Gradient boosting Algorithm and Naïve Bayes classifier	SVM (Support Vector Machine) -Gradient boosting Algorithm -Naive Bayes classifier -REGEX classifier	REGEX gives a very good performance with 97% accuracy and also 100% precision From the another algorithm
Latchoumi, T et al. [7]	-SQLI attacks are detecting and preventing, When each applications home page is switched to a test page.	- The model will be created when the SVM algorithm has been trained with all possible harmful expressions	(SVM) Support Vector Machine	good findings that can be examined in the mixing matrix and ROC chart to detect and avoid SQLIA. .

Farooq, U [12]	- to detect SQL Injection attack.	-detect SQL attacks by tokenizing queries and splitting them into their corresponding tokens. -evaluate the performance of their proposed model, they used the algorithms to the tokenized dataset.	-GBM -AdaBoost -XGBM -LGBM	-LGBM produces the best results, with an accuracy=0.993371 AdaBoost produces the lowest results in accuracy=0.991098
Mishra, S et al. [29]	- to classify and detect SQL Injection attacks.	Implementation the Naive Bayes algorithm and compare the results to Gradient Boosting.	-Gradient Boosting Classifier -Naïve Bayes classifier	Naïve Bayes accuracy is 92.8 - Gradient Boosting Classifier accuracy is 97.4

3. RESEARCH METHODOLOGY

We focused on detecting SQL injection attacks in cloud computing platform using machine learning rather than traditional methods because artificial intelligence has a greater ability to obtain detection methods for these attacks than traditional approaches, which have limitations in this regard. The suggested model's main goal is to detect SQL Injection attacks, The technique is divided into four steps: The initial step involves gathering a dataset where we have collected a dataset containing SQL queries and SQL injection attack queries, The second stage involves data preprocessing, in the third stage, feature extraction is performed for the data set, The model is trained in the fourth stage with four machine learning algorithms which are KNN,MNB, DT and SVM algorithms , In this phase, 80% of the dataset is used to train the model, and the fifth stage focuses on testing and evaluating the proposed model with 20% of dataset that we separated from the collected dataset.

3.1 System Model

Because of the high ability of machine learning to obtain detection methods for these attacks that exceed the ability of traditional methods, which had limitations in this aspect, we focus on detection of SQL injection attacks in cloud computing using machine learning instead of traditional methods in this work.

Machine learning is an AI application that uses pre-programmed algorithms to execute a certain function on data and improves over time, The algorithm must be presented several samples of data and the appropriate forecasts for those samples in order to learn, it can be utilized to make predictions.

Supervised, unsupervised, and semi-supervised are the three types of machine learning algorithms [23] [35], Figure 3 shows the our system model from the dataset stage to evaluation stage

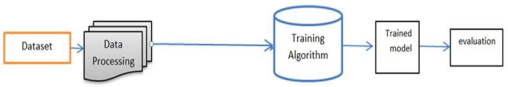

Figure 3: Our System Model

3.1.1 Dataset

Gathering a useful dataset that contains both SQL injection attacks and normal SQL queries is the essential step in detecting a SQLIA. In our research, we used a SQL injection set of data, and we employed preprocessing techniques to prepare the data for training. We then extracted certain features. After that, we used the machine learning algorithm to train the model with the training data and extract features. After that, some performance metrics are used. We decided to divide the dataset into two parts: training and testing. The training set contains 80% of the data and is used to train the models, while the remaining 20% is used to evaluate the trained models. We collected datasets from publicly available repositories. We used the Kaggle website to collect data on SQL injection attacks and benign traffic.

3.1.2 Data Preprocessing

Data preprocessing is critical before entering it into our model because the quality of the data and the meaningful information that can be extracted directly influences our model's ability to learn [36], removing all nulls, for example.

3.1.3 Feature extraction

The process of translating raw data into numerical features that can be processed while keeping the information in the original data set is known as feature extraction. It produces better outcomes than applying machine learning to raw data directly[36].
because our dataset contains a queries that is a text value, it must be changed to an INT value before the training can begin, In order to accomplish this, we'll use countVectorizer class to convert the text value to its numeric equivalent.
The countVectorizer is a basic tool for tokenizing a set of text documents and creating a vocabulary of known words, as well as encoding new documents using that vocabulary. The result is an encoded vector with the whole vocabulary's length and an integer count of how many times each word appears in the document [37].

3.1.4 Training Algorithms

For training, we'll utilize K-Nearest Neighbor algorithm, Multinomial Naive Bayes algorithm, Decision tree algorithm and Support Vector Machine algorithm, which are all machine learning algorithms.

3.1.4.1 K-Nearest Neighbors Algorithm

K-nearest Neighbors is a supervised ML algorithm for classifying and predicting data, In classification cases, it assigns new data to a specific class based on its similarity to classified data in that class, which can be determined by computing the shortest distance between the updated data and the classified data. A common method for computing distance in KNN is to use the distance, which calculates a straight line between two points. To calculate the distance in KNN, use the following equation: The distance between points A(x1,y1) and B(x2,y2) is:

$$= \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

After computing the distance between the updated data and the data obtained, we can choose the K number of nearest neighbors to see which of these K neighbors is the most comparable. K is a positive number which is determined by the scenario's requirements [37].

3.1.4.2 Multinomial Naïve Bayes Algorithm

MNB is a Naive Bayes variant developed to perform text document classification, MNB uses a multinomial distribution as a classification feature, with the number of words appearing or the word weight, MNB counts the number of times each word appears in the document [39].

3.1.4.3 Decision Tree Algorithm

Decision trees are a form of prediction model that has been used to address classification issues, a decision tree generates models that allow object categorization by generating a set of decision rules, These rules are derived from the attributes of the training data, Each child node in the tree, including its branches, represents a set of characteristics that contribute to decision tree classification, As a result, classifying an object will begin by looking at the root node's value, then progressing down the tree branches that correspond to those values, This is repeated for each node until it reaches the leaf node, at which point it can no longer go any further, While developing the decision tree for the best model, the Information Gain (IG) are utilized to choose the root node and sub-node [24] [40].

3.1.4.4 Support Vector Machine Algorithm

SVM is a collection of supervisory learning algorithms based on statistical learning theory that can be utilized for classification and regression applications. SVM is a global model of classification that generates non-overlapping portions and employs all attributes in general as a classification system. SVM is based on maximum margin and linear discriminant, comparable to a probabilistic technique, but without taking into account inter-quality relationships. The SVM classifier's core principle is to select this method, the max bridge plane[7].

An SVM is a linear classifier, It is common practice to employ machine learning (ML) techniques to categorize SQL injection attacks, and one of the most prominent ML algorithms for classification [30].

3.1.5 Evaluation

We evaluated the performance of all classification models using a dataset, which is 20% of the total data set. Once we evaluate the four models that emerged from the machine learning algorithms for training, we compare the results of each model.

3.1.5.1 Confusion Matrix

The Confusion Matrix is a measuring performance table that includes both actual and predicted data. The rows reflect the true value data,

while the columns indicate the predicted data, as shown in Figure 4 When a prediction comes correct, it is referred to as a true positive (TP), True negative (TN) refers to when a prediction was made that turned out to be correct, When a prediction is falsely negative, it is referred to as a false negative (FN), While false positive (FP) refers to a positive prediction that turns out to be false [41].

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 4: confusion matrix

3.1.5.2 ROC-AUC Curve

The Area Under the Curve (AUC) ROC curve is a performance statistic for classification issues at various threshold levels. AUC represents the degree or measure of separability, whereas ROC is a probability curve. It indicates how well the model can distinguish between classes. The AUC indicates how well the model predicts 0 classes as 0 and 1 classes as 1. The higher the AUC, the better the model predicts 0 classes as 0 and 1 classes as 1.
The ROC curve is plotted with TPR on the y-axis and FPR on the x-axis, with TPR on the y-axis and FPR on the x-axis, The TPR and FPR equations [42].

3.2 Tools

3.2.1 python

Python is an object-oriented, interpreted, mid-level programming language that is easy to learn and use while being versatile enough to tackle a number of problems, and we use it to implement all tasks in this project, Since its first introduction in 1991, its open-source nature has catapulted its popularity, and it is now rated one of the top programming languages to learn [41], and we used the Google Colab platform to do so.

3.2.2 Google Colab

Google Colab is a cloud-based collaboration tool provided by Google. This feature enables you to run computational simulations with Python and some libraries already installed, save files to Drive, and quickly and easily complete the setup process [43].

4. EXPERIMENTATION AND EVALUATION

4.1 KNN Performance

In KNN the nearest neighbor value utilized for training our dataset was 5, After that, we used the training set to train our model, We then put our model to the test by feeding it the test dataset, Following that, the model's accuracy was determined to be 92.45 % For this algorithm, we've also created a classification report and the confusion matrix heat map of the KNN model.

	precision	recall	f1-score	support
0	0.99	0.89	0.94	3893
1	0.84	0.99	0.91	2291
accuracy			0.92	6184
macro avg	0.91	0.94	0.92	6184
weighted avg	0.93	0.92	0.93	6184

Figure 5: Result of KNN Model

4.1.1 Confusion Matrix

Figure 6 shows the confusion matrix heat map of the KNN model. Actual values are shown by rows, whereas predicted values are represented by columns, The True Negative value is represented by the [0] [0] cell, the False Positive value is represented by the [0] [1] cell, the False Negative value is represented by the [1] [0], and the True Positive value is displayed by the [1] [1] cell. In the heat map of the confusion matrix of the KNN model, it turns out that when the model was trained using the KNN algorithm, the query is detected and classified as normal or attack. The TP of this classification was 2258, and TN was 3459, as was the FP, FN were 33 and 434, respectively.

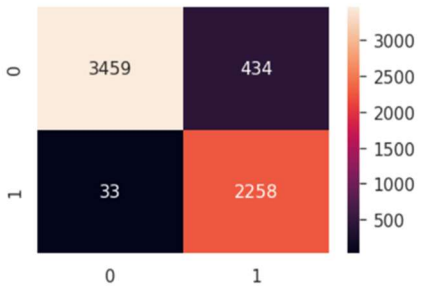

Figure 6: Confusion matrix with heat map of KNN model

4.2 MNB Performance

In the MNB algorithm, the model was also trained on the same set of data, and then we tested the model, and the model got an accuracy of about

Journal of Theoretical and Applied Information Technology
15th August 2022. Vol.100. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

97.08%, In the figure below, we display a classification report for the MNB model

	precision	recall	f1-score	support
0	0.96	0.99	0.98	3893
1	0.98	0.94	0.96	2291
accuracy			0.97	6184
macro avg	0.97	0.96	0.97	6184
weighted avg	0.97	0.97	0.97	6184

Figure 7: Result of MNB Model.

4.2.1 Confusion Matrix

In Figure 8 The query is identified and classed as normal or assault when the model was trained using the MNB algorithm, according to the heat map of the confusion matrix of the MNB model. The TP and TN for this category were 2144 and 3860, respectively, whereas the FP and FN were 147 and 33.

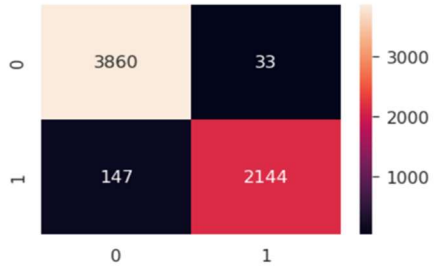

Figure 8: Confusion matrix with heat map of MNB model

4.3 Decision Tree Performance

The accuracy value of the Decision Tree model is 99.4%, In the figure below, we display a classification report for the Decision Tree model

	precision	recall	f1-score	support
0	0.99	1.00	1.00	3893
1	0.99	0.99	0.99	2291
accuracy			0.99	6184
macro avg	0.99	0.99	0.99	6184
weighted avg	0.99	0.99	0.99	6184

Figure 9: Result of Decision Tree Model

4.3.1 Confusion Matrix

In Figure 10 the heat map of the confusion matrix of the Decision Tree model, it turns out that when the model was trained using the Decision Tree algorithm, The TP of this classification was 2266, and TN was 3882, as was the FP, FN were 25 and 11, respectively.

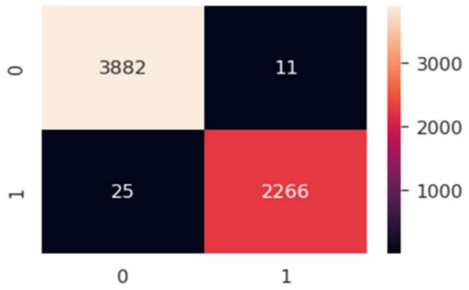

Figure 10: Confusion matrix with heat map of Decision Tree model.

4.4 SVM Performance

The SVM model was trained on the data set, and the performance of the model was tested, and it got an accuracy of 99.42%, we display a classification report for the SVM model

	precision	recall	f1-score	support
0	0.99	1.00	1.00	3893
1	1.00	0.98	0.99	2291
accuracy			0.99	6184
macro avg	1.00	0.99	0.99	6184
weighted avg	0.99	0.99	0.99	6184

Figure 11: Result of SVM Model

4.4.1 Confusion Matrix

Figure 12 show the heat map of the confusion matrix of the SVM model, it turns out that when the model was trained using the SVM algorithm, The TP of this classification was 2256, and TN was 3892, as was the FP, FN were 35 and 1, respectively.

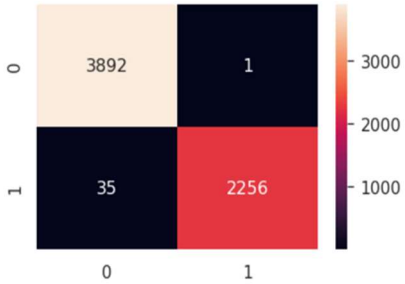

Figure 12: Confusion matrix with heat map of SVM model.

4.5 ROC-AUC Curve

The ROC-AUC values for all trained classifiers are shown in Figure 13 Which demonstrates that all of the classifiers performed well in the classification, but the SVM and Decision tree classifier performed the best, followed by the MNB classifier and finally the KNN classifier.

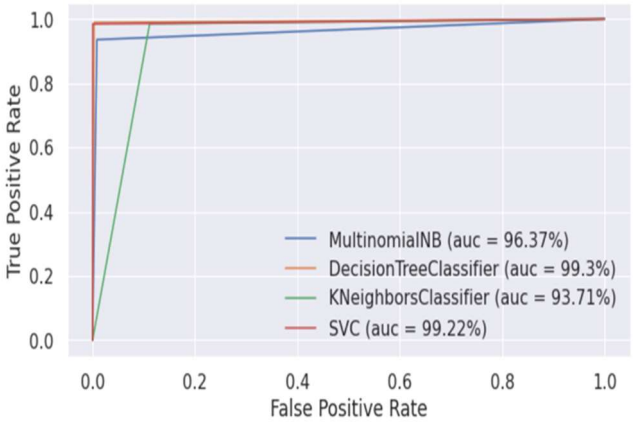

Figure 13: ROC –AUC of all models.

In this Table, we compare the accuracy, confusion matrix, F-value and AUC of all of the algorithms we used in this research:

Table 2: The performance values for the four classification models are summarized.

performance measurement/ classification model	KNN model	MNB model	Decision Tree model	SVM model
Accuracy	92.45%	97.09%	99.4%	99.42%
Confusion Matrix	TP= 2258 TN=3459 FP=33 FN= 434	TP= 2144 TN= 3860 FP= 147 FN= 33	TP= 2266 TN=3882 FP= 25 FN= 11	TP= 2256 TN= 3892 FP= 35 FN= 1
F-value	Attack= 0.91 Legit=0.94	Attack= 0.96 Legit= 0.98	Attack = 0.99 Legit= 1	Attack = 0.99 Legit= 1
AUC	93.71%	96.37%	99.3%	99.22%

5. CONCLUSION

We have outlined the SQL attack in our research because it is a serious threat that ranks first on the OWASP list of the most dangerous web attacks, SQL injection attacks on web applications are a serious problem, and it is critical to find a workable solution to this problem, As a result, we present a SQL injection detection tools based on machine learning algorithms, After training the model with four machine learning algorithms (KNN, MNB,DT and SVM), When applied to the dataset, the suggested model obtains an average accuracy of more than 99 percent with a very low error rate, indicating that the feature set chosen is extremely effective at distinguishing SQL injection attack requests from conventional SQL queries and plain text, For real-world detection systems, the results show that our suggested method, which is based on machine learning and includes the selected attributes may be used to detect SQL injection attacks. The Support Vector Machine algorithms produced the best accuracy of 99.42%, and the Decision Tree algorithm with accuracy of 99.4% and The accuracy of the other two algorithms we utilized, MNB and KNN algorithms was 97.09 % and 92.45 %, respectively. Finally, we have found Our models yield near-perfect results with a very low error rate.

REFERENCES

[1] A. K. Hegde, and P. N. Jayanthi, "A Survey on SQL Injection Attacks and Prevention Methods",2020.

[2] U. A. Butt, M. Mehmood, S. B. H. Shah, R. Amin, M. W. Shaukat, S. M. Raza, and M. Piran, "A review of machine learning algorithms for cloud computing security. Electronics," vol. 9, 1379, 2020.

[3] B. I. Mukhtar, and M. A. Azer, "Evaluating the Modsecurity Web Application Firewall Against SQL Injection Attacks," *In 2020 15th International Conference on Computer Engineering and Systems (ICCES),* December 2020, pp. 1-6.

[4] K. Abouelmehdi, H. Khaloufi, and A. Beni-hssane, "Risk assessment of SQL injection: An experimental study," *In 2021 7th International Conference on Optimization and Applications (ICOA),* May 2021, pp. 1-4.

[5] S. Leelavathy, "A Secure Methodology to Detect and Prevent Ddos and Sql Injection Attacks," *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* vol.12, 2021, pp.341-346.

[6] I. Hashem, M. Islam, S. M. Haque, Z. I. Jabed, and N. Sakib, "A Proposed Technique for Simultaneously Detecting DDoS and SQL Injection Attacks," *International Journal of Computer Applications,* vol. 975, pp. 8887.

[7] T. P. Latchoumi, M. S. Reddy, and K. Balamurugan, "Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention," *European Journal of Molecular & Clinical Medicine,* 2020, vol.7.

[8] N. Singh, and A. K. Singh, "SQL-Injection Vulnerabilities Resolving using Valid Security Tool in Cloud," Pertanika *Journal of Science & Technology,* 2019, vol.27.

[9] I. Tasevski, and K. Jakimoski, "Overview of SQL Injection Defense Mechanisms," *In 2020 28th Telecommunications Forum (TELFOR),* November 2020, pp. 1-4.

[10] A. Z. Ablahd, and S. A. Dawwod, "Using Flask for SQLIA Detection and Protection," *Tikrit Journal of Engineering Sciences*, 2020, vol.27.

[11] S. Ibarra-Fiallos, J. B. Higuera, M. Intriago-Pazmiño, J. R. B. Higuera, J. A. S. Montalvo, and J. Cubo, "Effective Filter for Common Injection Attacks in Online Web Applications," IEEE Access, vol.9, 2021, pp.10378-10391.

[12] U. Farooq, "Ensemble Machine Learning Approaches for Detection of SQL Injection Attack," Tehnički glasnik, vol.15, , 2021, pp. 112-120.

[13] D. Tripathy, R. Gohil, and T. Halabi, "Detecting SQL injection attacks in cloud SaaS using machine learning ," *In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS),* May 2020, pp. 145-150.

[14] I. Jacob, and M. Pirnau, "Sql Injection Attacks And Vulnerabilities," *Journal of Information Systems & Operations Management,* pp.68-81.

[15] J. Zheng, and X. Shen, " Pattern Mining and Detection of Malicious SQL Queries on Anonymization Mechanism," *IEEE Access,* vol.9, 2021, pp.15015-15027.

[16] Y. Abdulmalik, "An Improved SQL Injection Attack Detection Model Using Machine Learning Techniques," *International Journal of Innovative Computing,* vol.11, 2021, pp.53-57.

[17] W. H. Rankothge, M. Randeniya, and V. Samaranayaka, "Identification and Mitigation Tool for Sql Injection Attacks (SQLIA)," *In 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS),* November 2020, pp. 591-595.

[18] A. K. Shwaish, M. A. Hussain, and H. A. Al-Kashoash, "Encoding Query Based Lightweight Algorithm for Preventing SQL injection attack," *Journal of Basrah Researches ((Sciences)),* 2020, vol.46.

[19] R. Shobana, and M. Suriakala, "A Thorough Study On Sql Injection Attack-Detection And Prevention Techniques And Research Issues,".

[20] A. S. S. Ahmed, A. A. Ahmed, M. Ahmed, N. S. Shourav, and N. Sakib, "An Approach to Detect Cyber Attack on Server-side Application by using Data Mining Techniques and Evolutionary Algorithms," 2021.

[21] Z. C. S. S. Hlaing, and M. Khaing, "A detection and prevention technique on sql injection attacks," In 2020 IEEE Conference on Computer Applications (ICCA) , February 2020, pp. 1-6.

[22] F. Q. Kareem, S. Y. Ameen, A. A. Salih, D. M. Ahmed, S. F. Kak, H. M. Yasin, and N. Omar, "SQL injection attacks prevention system technology," *Asian Journal of Research in Computer Science,* vol. 13, 2021, pp.32.

[23] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review. Sensors,", vol.18, 2018, pp.2674.

[24] X. D. Hoang, "Detecting Common Web Attacks Based on Machine Learning Using Web Log," *In International Conference on Engineering Research and Applications* , December 2020, pp.311-318.

[25] 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 Journal on Information Security,* vol.20, 2020, pp.1-14.

[26] S. D. Samarin, and M. Amini, "Preventing SQL injection attacks by automatic parameterizing of raw queries using lexical and semantic analysis methods," *Scientia Iranica. Transaction D, Computer Science & Engineering, Electrical,* vol.26, 2019, pp. 3469-3484.

[27] K. N. Durai, R. Subha, and A. Haldorai, "A Novel Method to Detect and Prevent SQLIA Using Ontology to Cloud Web Security," Wireless Personal Communications, vol.117, 2021, pp. 2995-3014.

[28] N. Cahyadi, "SQL injection Detection Using Deep Learning A Project Report,".

[29] S. Mishra, "SQL injection detection using machine learning," , 2019.

[30] B. Kranthikumar, and R. L. Velusamy, "SQL injection detection using REGEX classifier. Journal of Xi'an University of Architecture & Technology," vol.12, 2020, pp. 800-909.

[31] D. Chen, Q. Yan, C. Wu, and J. Zhao, " Sql injection attack detection and prevention techniques using deep learning," *In Journal of Physics: Conference Series,* Vol. 1757, No. 1, 2021, p. 012055.

[32] S. A. Idowu, O. Awodele, S. O. Kuyoro, and J. E. Akinsola, "Taxonomy and Characterization of Structured Query Language Injection Attacks for Predictive Analytics," 2020.

[33] P. Lin, W. Jinshuang, C. Ping, and Y. Lanjuan, "SQL Injection Attack and Detection Based on GreenSQL Pattern Input Whitelist," *In 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)* , September 2020, pp. 187-190.

[34] M. Arock, "Efficient Detection Of SQL Injection Attack (SQLIA) Using Pattern-based Neural Network Model," In 2021 *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* , February 2021, pp. 343-347.

[35] S. A. Repalle, and V. R. Kolluru, "Intrusion detection system using ai and machine learning algorithm," *International Research Journal of Engineering and Technology (IRJET),* vol.4, 2017,pp.1709-1715.

[36] J. Burdack, F. Horst, S. Giesselbach, I. Hassan, S. Daffner, and W. I. Schöllhorn, "Systematic comparison of the influence of different data preprocessing methods on the performance of gait classifications using machine learning," *Frontiers in bioengineering and biotechnology*, 2020, vol. 8, pp.260.

[37] B. M. Ajose-Ismail, O. V. Abimbola, and S. A. Oloruntoba, "Performance Analysis of Different Word Embedding Models for Text Classification," *International Journal of Scientific Research and Engineering Development*, vol.6, 2020, pp.1016-1020.

[38] A. Alam, M. Tahreen, M. M. Alam, S. A. Mohammad, and S. Rana, "SCAMM: detection and prevention of SQL injection attacks using a machine learning approach,", 2021.

[40] F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT*),vol. 48, 2017, pp.128-138.

[41] D. Chicco, N. Tötsch, and G. Jurman," The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData mining*, vol.14, 2021, pp.1-22

[42] R. Kannan, and V. Vasanthi, "Machine Learning Algorithms With Roc Curve For Predicting And Diagnosing The Heart Disease," *In Soft Computing And Medical Bioinformatics*, 2019, pp. 63-72

[43] R. P. M. Vieira, F. R. V. Alves, and, P. M. M. C. Catarino, " Alternative views of some extensions of the padovan sequence with the Google Colab," vol.2, 2019, pp.266-273.