مجلة الحاسوب والتقانة العلمية
Scientific Journal of Computer and Technology

Design of a System Model Based on Machine Learning Technique for SQL Injection Detection

Abdalla Adel Abdalla Hadabi
Computer Science Department
Al-Neelain university
Khartoum, Sudan
aboadel30@gmail.com

Eltyeb Elsamani AbdElgabar Elsamani
Computer Science Department
Al-Neelain university
Khartoum, Sudan
tayebsamani@gmail.com

المستخلص

تمثل هجمات حقن الاستعلام الهيكلى ثلثي هجمات تطبيقات الويب. ما يقدر بنحو 25٪ من الخروقات العام الماضي بدأت بهجوم حقن الاستعلام الهيكلى. يعد حقن الاستعلام الهيكلى هجومًا شائعًا فى الويب و يمثل تحديًا لأمن الشبكة ؛ تتسبب هجمات حقن الاستعلام الهيكلى في خسائر مالية في جميع أنحاء العالم بالإضافة إلى كونها تخترق خصوصية بيانات المستخدم. أصبح اكتشاف حقن الاستعلام الهيكلى موضوعًا ساخنًا مؤخرًا. جذبت كيفية الدفاع ضد هجمات حقن الاستعلام الهيكلى بشكل فعال انتباه المتخصصين والباحثين في أمن الويب. الهدف من هذه الورقة هو تقديم نموذج يمكنه تحديد هجمات حقن الاستعلام الهيكلى بشكل فعال بناءً على بيانات الإدخال. لقد أنشأنا نموذجًا للتعلم الآلي يعتمد على خوارزمية الانحدار اللوجستي لاكتشاف هجمات حقن الاستعلام الهيكلى استنادًا إلى بيانات سجل الويب التاريخية ، وتم جمع مجموعة البيانات من موقع مستودع على الإنترنت يحتوي على 4201 مدخلا. حقق النموذج دقة 0.93 ، حساسية 0.78 ، خصوصية 0.81 ، ضبط 0.98. بما يتجاوز الدقة ، تم النظر في مقاييس الأداء الأخرى لتصميم النموذج الأمثل. يعد استخدام تقنيات التعلم الآلي لاكتشاف هجوم حقن الاستعلام الهيكلى مفيدًا جدًا ويمكن استخدامه حتى في تطبيقات الوقت الفعلي.

الكلمات المفتاحية: حقن الاستعلام الهيكلى، تعلم الالة، الانحدار اللوجستى.

Abstract

SQL Injection Attacks Represent Two-Third of All Web App Attacks. An estimated 25% of breaches last year started with an SQL Injection attack. SQL injection is a popular web attack and has been a challenging matter for network security; SQL causes financial losses worldwide as well as user data offensive. SQL injection detection has become a hot topic recently. How to defense against SQL injection attacks effectively has drawn the attention of web security professionals and researchers. The objective of this paper was to introduce a model that could identify SQL injection attacks effectively based on entry data. We built a machine learning model based on a logistic regression algorithm to detect SQL injection attacks based on historical web log data, the dataset was collected from an online repository website, containing 4201 entries. The model achieved an accuracy of 0.93, sensitivity 0.78, specificity 0.81, and precision of 0.98. Therefore, beyond accuracy, other performance metrics were considered for optimal model design. Using machine learning techniques for SQL attack detection is very useful and can be used even in real-time applications.

Keywords: SQL injection, machine learning, logistic regression.

1. Introduction

web applications are broadly used in many sectors of life due to the availability and accessibility they offer. Therefore, web applications have become a appropriate goal for attackers, and then it’s required to keep it safe. But, these types of applications have other sorts of attacks one of the most dangerous attack. Injection Attack (SQLIA) is used to attack Web applications. Moreover, SQL Injection is a weakness that happens when the attacker has the facility to change the Structured Query Language (SQL) that an application permits to a database. The ability to change what is passed to the database, the attacker can alter the syntax of SQL itself, in addition to the control of supporting database and operating system functionality accessible to the database. SQL injection effects go further beyond Web applications, due to the reality that any code takes input from an unauthorized source and used as SQL statements are exposed to SQL injection attack.

The attacker can extract confidential information using SQL injection vulnerability, or even get the privilege of the database admin [1]. OWASP announced the top 10 in 2017 (The Open Web Application Security Project

(OWASP) is an open community that allows organizations to develop, purchase, and sustain applications that can be reliable); the highest frequency among web application attacks is SQL injection attack. OWASP reported that web applications need more security, and how these attacks happen, also, shows the top ten security risks that cause Web Application attacks [2]. Besides, positive technologies organization testers found more than 70 types of weaknesses in web applications as shown in the figure below.

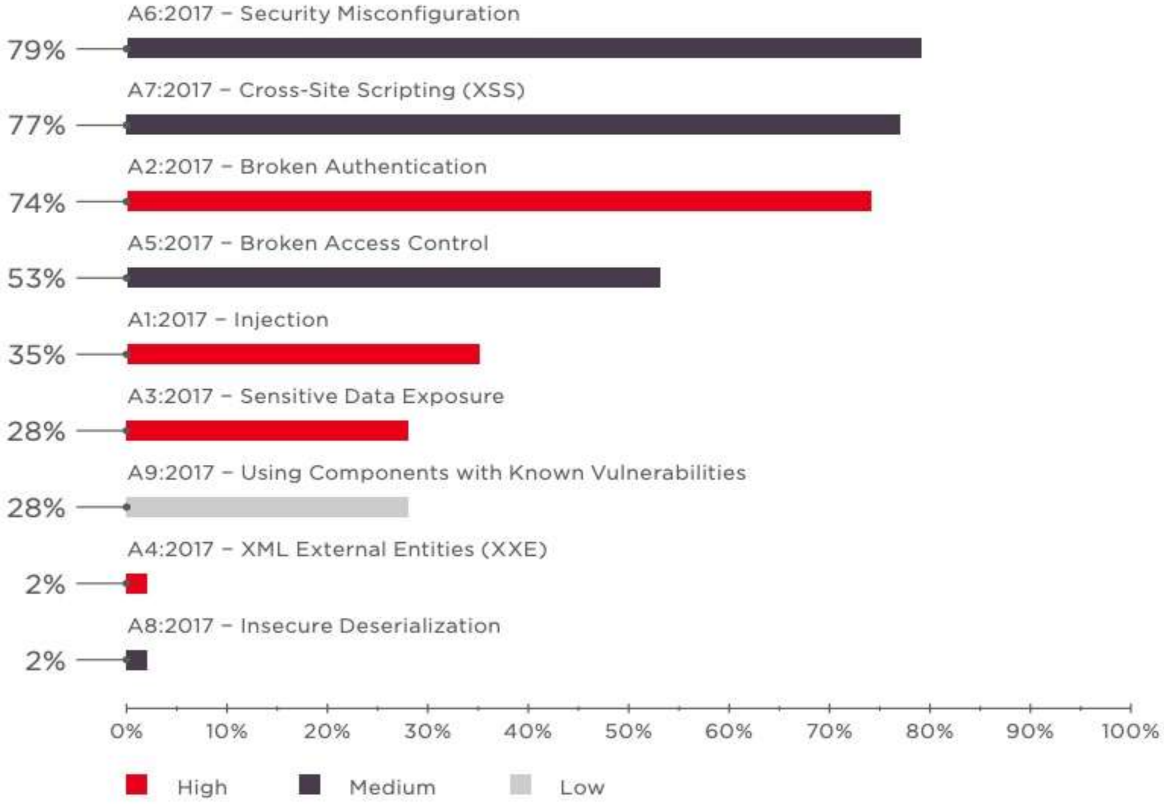

Fig 1: **OWASP Top 10‒2017 vulnerabilities (percentage of web applications according to positive technologies[2]).**

There is inadequate input validation because when developers code their web application they give emphasis to functionality more than security, as a result, the SQL injection attack happens and gives the attacker unlimited access to the database [3]. Therefore, businesses deliver many services to users via web applications by getting their requests with the back-end database and return appropriate data for users.

Web browser take the inserted code and react with the back end of the Database on the presentation tier[4]. Regularly, the back end of the Database holds sensitive and private data as an example financial data which is become an attractive target for many hackers.

SQL injection attacks were divided into three categories by researchers static, dynamic, and hybrid [1]. The static analysis tests the precision of the produced SQL queries to find any mismatch on the queries [5]. While the dynamic analysis allows the system to identify the legitimacy of SQL in the queries that are valid [1]. Hybrid techniques associates the pros of static and dynamic analysis. But, the combination is done by using static analysis first to build and train models of detection after that arises the need to take the correct decision by using the dynamic analysis by inspecting these models[1]. Machine learning is applied in both hybrid and dynamic analysis. False negatives and false positives happen due to the used classifier [5]. SQL injection detection can be improved once using up to date datasets even using the same classifier [1].

In this paper, a model has been proposed by using machine learning to classify SQL injection attacks. We built a machine learning model based on a logistic regression algorithm to detect SQL injection attacks based on historical weblog data.

1.1 SQL Injection Attack Methods

The attacker performs the attack using one of the SQL injection attack method.

i. **Retrieve Hidden Information**
Hackers change the SQL request in a way that it brings more results from the targeted database.

ii. **Subvert Logic of the App**
Hackers execute it on the web app’s logic by changing the SQL queries.

iii. **UNION Attack**
The union operator found in SQL is used to join several tables. Thus, union query-based attack aims to compromise data privacy. When an attacker adds code that has the union operator, then the attacker is attempting to return extra data than the query projected[6].

iv. **Scouting a Database**

With this kind, attackers use commands to return confidential data about the database like its version and structure.

v. Blind SQL injection

Web application usually does not send response with details of a query or database errors. This type is the most challenging because it can be used to manipulate data[7].

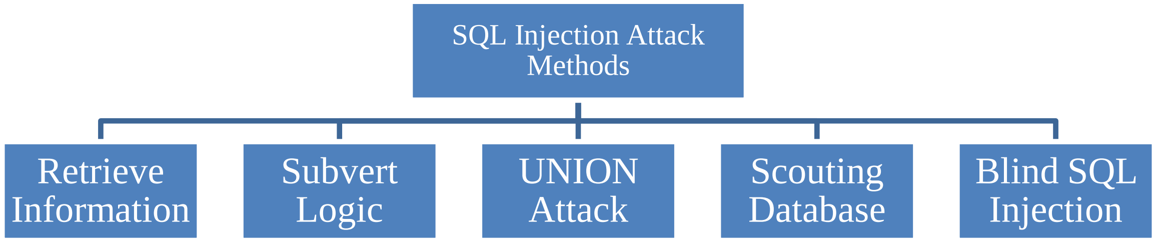

Fig 2. SQL injection attack Methods.

1.2 Securing Web Applications from SQL Injection

Scholars suggested two methods to protect web applications from SQL Injection attack. The first approach involves writing code for the web application to guarantee enough user input validation. In production web applications need improvement to include security mechanisms. The cost of modifying the software during development is much less, compared to after development. It is a better technique to protect a web application from SQL injection when the software is under development process [8].

The second approach involves the deployment of additional system intended to verify the legitimacy of produced queries by a web application before they run on the database. Still these methods have a downside for not being an inclusive solution to the problem. On the other hand machine learning models are exposed to the false negative and false positive when the classifier perceives valid queries as malicious code and prevents them or allowing harmful queries to pass causing a security breach[8].

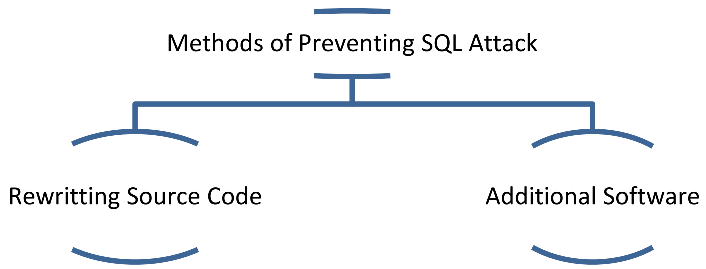

Fig 3. Methods of preventing SQL attack.

2. Machine learning Models to Detect SQL Attack

In recent years machine learning techniques have shown a great success in learning complicated patterns that permit them to make predictions about new data[9][10]. Machine learning has witness technological advances in recent years and been used widely in a range of applications especially for security [11]. Machine learning offers smart algorithms to identify vulnerabilities in Web Application Firewalls against SQL injection attacks[12] [13] [14]. Another study used a protective coding method for SQL attack detection and prevention [15].

2.1 System Model

Our proposed solution contains four components client, proxy server, classifier, and database server. The next scenario will present how the system functions. Frist the client makes a request which received by the proxy server the role of the proxy is to add a security layer to our solution by having a list that contains the attacker's information to prevent them from connecting to our database server, later on, new tuple added to the list when the classifier receives the request from the proxy server and classify the request as a malicious request vice versa it classifies it as a normal request and passes the request to database server. Thus database server will execute the query within the backend database and send the results to the client. The next figure illustrates the above-mentioned scenario.

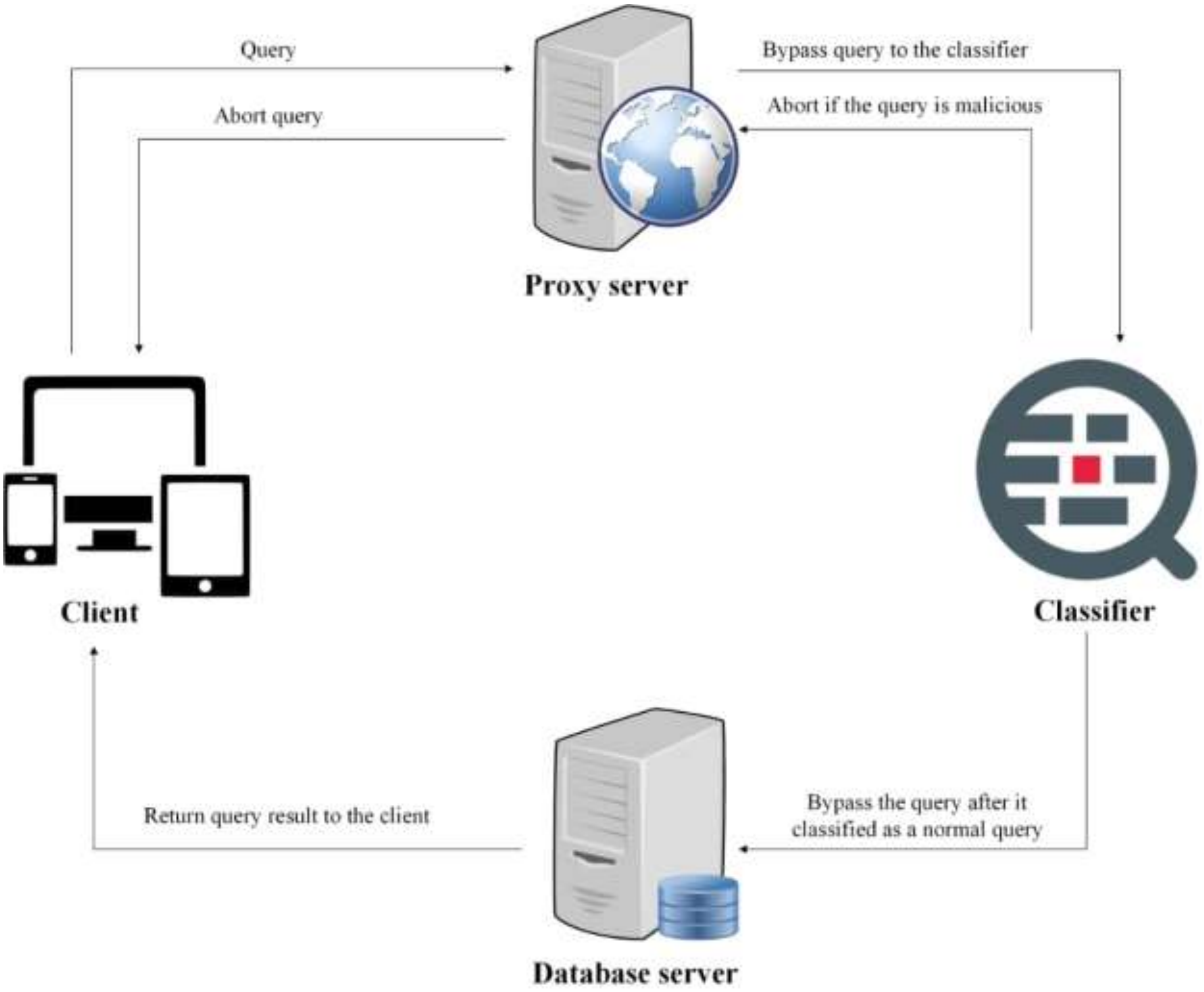
Query
Abort query
Proxy server
Bypass query to the classifier
Abort if the query is malicious
Client
Classifier
Return query result to the client
Bypass the query after it classified as a normal query
Database server

Fig 4. Illustrates system model design.

3. Methodology

The methodology consists of several steps starting with data collection, then data preprocessing, and followed by model training and testing last evaluation using performance metrics for model improvement.

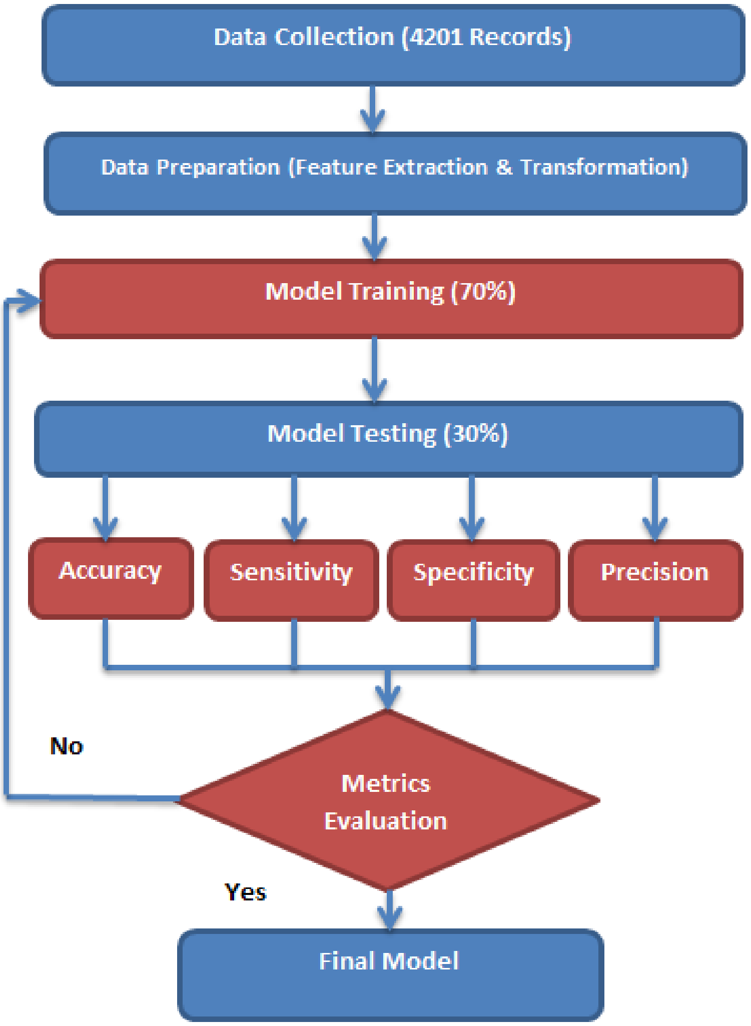

Figure 5: illustrates the steps for model design.

Logistic regression method is a kind of linear model that is used for datasets when the dependent variable is categorical[16]. Logistic regression is an efficient prediction technique for many classifications types of problems. Logistic regression is used when the dependent variable is categorical and produces output in terms of probabilities. To estimate the logistic regression model using the probability of the target variable based on one or more predictor variables. It is effective when the dependent variable of a dataset is binary[17] [18].

4. Model Design

developing techniques for reasoning under uncertainty has become one of the most interesting field of machine learning. Machine learning has been used for many years to address a wide variety of real life problems in many fields [9]. Web security attacks are analyzed to discover hidden patterns and insight from user input queries by machine learning. It can also discover unknown and new patterns [19]. The machine learning (ML) models are created based on the historical data using a logistic regression algorithm; we evaluated the model performance and measured the model accuracy on the testing data. We divided the dataset into two parts: training and testing set consisting of 0.70 and 0.30 percent respectively. The model achieved an accuracy of 0.93, sensitivity 0.78, specificity 0.81, and precision of 0.98.

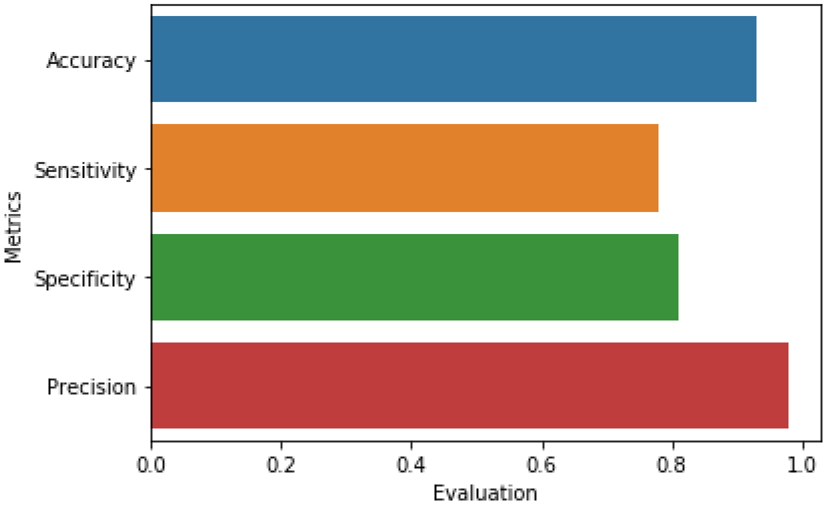

Figure 6: shows the performance metrics evaluation.

from to the figure above the model achieved an accuracy and precision of 0.93 and 0.98 respectively. But, the model achieved less for sensitivity 0.78 and specificity 0.81. Thus, sensitivity and specificity need more improvements.

5. Model Evaluation

We evaluated the best algorithm which was the logistic regression model in terms of accuracy, recall, specificity, and precision as shown below. The number of correct and incorrect classifications in each potential value of the classified variables to evaluate the outcomes gained. The following formulas are used to calculate the accuracy, sensitivity, specificity, and precision of the designed model [20].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2)$$

Recall or also widely identified as sensitivity

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (3)$$

Specificity is stated as the proportion of actual negatives.

$$\text{Specificity} = \frac{TN}{TN+FP} \qquad (4)$$

The precision of all the records we predicted positive.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5)$$

6. Results and Discussion

We introduce a model that can identify SQL injection attacks effectively based on user input patterns. We built a machine-learning model based on a logistic regression algorithm to detect SQL injection attacks based on historical weblog data. The model achieved an accuracy of 0.93, sensitivity 0.78, specificity 0.81, and precision of 0.98. We used other performance metrics for the best model design.

7. Conclusion

In conclusion, this paper employs machine-learning models against security attacks. The proposed mod was evaluated using four performance metrics, using the four metrics allows us to see a bigger picture of our model and how it is expected to behave in different scenarios. This model detects SQL injection only; researchers should apply the proposed model to other types of cyber-attacks.

References

[1] D. G. Kumar and M. Chatterjee, "Detection Block Model for SQL Injection Attacks," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 11, pp. 56–63, 2014, doi: 10.5815/ijcnis.2014.11.08.

[2] "OWASP releases the Top 10 2017 security risks - SD Times." [Online]. Available: https://sdtimes.com/app-development/owasp-releases-top-10-2017-security-risks/. [Accessed: 29-Apr-2021].

[3] Z. Lashkaripour and A. Ghaemi Bafghi, "A simple and fast technique for detection and prevention of SQL injection attacks (SQLIAs)," *Int. J. Secur. its Appl.*, vol. 7, no. 5, pp. 53–66, 2013, doi: 10.14257/ijsia.2013.7.5.05.

[4] From百度文库, *済無No Title No Title*, vol. 53, no. 9. 2013.

[5] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," *Proc. 19th Int. Conf. World Wide Web, WWW '10*, pp. 281–290, 2010, doi: 10.1145/1772690.1772720.

[6] R. Dharam and S. G. Shiva, "Runtime Monitors to Detect and Prevent Union Query based SQL Injection Attacks," 2013, doi: 10.1109/ITNG.2013.57.

[7] "Here's Everything You Should Know About the Deadly SQL Injection in 2021 | My IT Guy." [Online]. Available: https://www.gomyitguy.com/blog-news-updates/sql-injection. [Accessed: 11-Mar-2021].

[8] P. R. Mcwhirter, K. Kifayat, Q. Shi, and B. Askwith, "Journal of Information Security and Applications SQL Injection Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence Kernel," *J. Inf. Secur. Appl.*, vol. 40, pp. 199–216, 2018, doi: 10.1016/j.jisa.2018.04.001.

[9] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-asl, and B. Yu, "Definitions , methods , and applications in interpretable machine learning," vol. 116, no. 44, 2019, doi: 10.1073/pnas.1900654116.

[10] Q. I. Li, W. Li, and J. Wang, "A SQL Injection Detection Method Based on Adaptive Deep Forest," pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[11] M. Rahouti, K. Xiong, and N. Ghani, “Bitcoin Concepts, Threats, and Machine-Learning Security Solutions,” *IEEE Access*, vol. 6, pp. 67189–67205, 2018, doi: 10.1109/ACCESS.2018.2874539.

[12] D. Appelt, C. D. Nguyen, A. Panichella, and L. C. Briand, “A Machine-Learning-Driven Evolutionary Approach for Testing Web Application Firewalls,” *IEEE Trans. Reliab.*, vol. PP, pp. 1–25, 2018, doi: 10.1109/TR.2018.2805763.

[13] M. Venkata, S. Soma, and R. K. Megalingam, “Applying and Evaluating Supervised Learning Classification Techniques to Detect Attacks on Web Applications,” no. 10, pp. 2222–2225, 2019, doi: 10.35940/ijitee.J9434.0881019.

[14] M. Kempanna, “Web Security Aware by using Naive Baye ’ s ML Technique,” no. 4, pp. 3222–3230, 2020, doi: 10.35940/ijitee.D1325.029420.

[15] E. T. Jide and A. Sunday, “SQL Injection Attacks Predictive Analytics Using Supervised Machine Learning Techniques,” vol. 9, no. 4, pp. 139–149, 2020.

[16] B. Heung, H. C. Ho, J. Zhang, A. Knudby, C. E. Bulmer, and M. G. Schmidt, “An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping,” *Geoderma*, vol. 265, pp. 62–77, 2016, doi: 10.1016/j.geoderma.2015.11.014.

[17] C. Zhu, C. U. Idemudia, and W. Feng, “Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques,” *Informatics Med. Unlocked*, vol. 17, no. January, p. 100179, 2019, doi: 10.1016/j.imu.2019.100179.

[18] M. Maniruzzaman *et al.*, “Accurate Diabetes Risk Stratification Using Machine Learning: Role of Missing Value and Outliers,” *J. Med. Syst.*, vol. 42, no. 5, pp. 1–17, 2018, doi: 10.1007/s10916-018-0940-7.

[19] J. H. Park, “Symmetry-adapted machine learning for information security,” *Symmetry (Basel).*, vol. 12, no. 6, pp. 1–4, 2020, doi: 10.3390/sym12061044.

[20] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu, “Knowledge-Based Systems Detection of SQL injection based on artificial neural network ☆,” *Knowledge-Based Syst.*, vol. 190, p. 105528, 2020, doi: 10.1016/j.knosys.2020.105528.