How to Cite:
Mondal, B., Banerjee, A., & Gupta, S. (2022). A review of SQLI detection strategies using machine learning. *International Journal of Health Sciences*, *6*(S2), 9664–9677. https://doi.org/10.53730/ijhs.v6nS2.7519

A review of SQLI detection strategies using machine learning

Biswajit Mondal
Computer Science and Engineering, Dr. B C Roy Engineering College, Durgapur, 713206, West Bengal, India

Abhijit Banerjee
Electronics and Communication Engineering, Dr. B C Roy Engineering College, Durgapur, 713206, West Bengal, India

Subir Gupta
Computer Science and Engineering, Dr. B C Roy Engineering College, Durgapur, 713206, West Bengal, India.
Corresponding author email: subir2276@gmail.com

***Abstract*---**Various platforms and web apps to deliver material via the Internet are becoming more widespread. Web-based technologies accept and store sensitive information from users. Because of their Internet connectivity, these systems and the databases they link to are vulnerable to various information security vulnerabilities. The most dangerous threats are denial of service (DoS) and SQL injection assaults. SQL Injection attacks are at the top of the list for web-based systems. In this type of attack, the perpetrator will take sensitive and classified information that might hurt a firm or enterprise. Depending on the conditions, the corporation may incur financial losses, have private information disclosed, and have its stock market value drop. This work uses machine learning-based classifiers such as MLP, Support Vector Machine, Logistic Regression, Naive Bayes, and Decision Tree to identify and detect SQL Injection attacks. For the SQLI dataset learning strategies, we examined all five algorithms using the Confusion Matrix, F1 Score, and Log Loss. We discuss the benefits and drawbacks of the proposed AI-based SQLI techniques. Finally, we talk about making SQLI reach its full potential through more research in the coming years.

***Keywords*---**SQL injection, MLP, support vector machine, logistic regression, naive Bayes, decision tree.

Manuscript submitted: 27 March 2022, Manuscript revised: 18 April 2022, Accepted for publication: 9 May 2022

Introduction

SQL, or Structured Query Language, is a query language used to interact with relational databases. "SQL Injection Assault" is a web hacking strategy that entails using SQL to launch assaults on databases and exploit them to achieve what the user wants[1][2]. A SQL injection is an attack that involves adding code and modifying a SQL query to gain unauthorized access to a database. SQL injection attacks are becoming more of a concern for cyber security professionals. SQL-based attacks accounted for over half of all web-based assaults discovered in the preceding year. SQL injection is a frequent type of cyber-attack. Several strategies have been devised to counter such assaults. On the other hand, cybercriminals keep coming up with new ways to get around the many security measures to protect against SQL Injection attacks. The SQL Injection attacks are classified into three categories:

a) Union Based SQLI
   The UNION operator is used when SQL requires two or more SQL statements or queries to be combined. A union-based SQL injection yields this performance, along with these values. The simple text does not appear anywhere in the query, which results in another type of insertion into the expression.
b) Error Based SQLI
   The SQL Injection error-based solution operates by sending an invalid query input and causing a failure in the database. The database is forced to do an operation that leads to a mistake. The customer then searches for database errors and uses those errors to learn how the database can be further manipulated using the SQL query.
c) Blind Based SQLI
   A Blind SQL Injecting attack is a technology in which the malicious attacker asks the database questions and chooses a further plan of action depending on the replies. This is the worst kind of SQL attack since details in the database are not understood. This solution method is used when basic errors such as "Syntax Error" are returned. Injection Blind SQL is a part of SQL Injection, DDOS (distributed denial-of-service) attacks, and Time-Based SQL Attacks[3][4][5].

Figure 1 shows all types of SQLI. Criminology researchers have challenged the utility of machine learning algorithms, claiming that they have difficulty identifying and avoiding cyber threats. Machine learning methods have been employed to identify SQL injections, and much time and effort have been invested in developing them. If a machine learning problem is to be solved, it can be done with one form, but there is no one-size-fits-all algorithm to fit all machine learning problems.

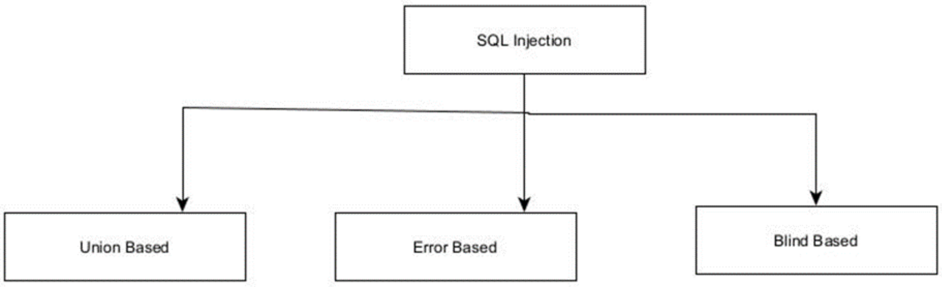

Figure 1: Types of SQLI

The primary aspects of this study:

- We have studied different AI strategies and analyzed their appropriateness and limits in tackling SQLI difficulties and needs.
- In addition, we comprehensively address possible future research avenues for integrating machine learning into SQLI.
-

The remainder of the paper is organized as follows. Section 2 examines the literature review. Significant advancements to date have resulted in a plethora of knowledge for future upgrades. Section 3 goes over technique, while Section 4 goes over results analysis. Finally, in Section 5, we will look at SQL Injection.

Literature Review

The following section summarises efforts to detect and prevent SQL injection attacks. Detection of injections is now divided into two subfields of research. One approach is to write SQL queries that validate input. The delivery of SQL queries from web apps to databases is also checked in other ways. The combination of these two techniques is crucial for detecting injection attacks. This section will go through inquiry methods. A General Dynamics Navy researcher created SQLLite[6]. They put together an attack submarine damage-control system with HP-UX and an IBM Informix database. SQLite functionality has now been added to TCL. In August of 2000, gdbm was the sole provider of SQLite 1.0. In version 2.0, transactions were handled via a custom B-tree implementation.

Waves is a well-recommended security testing approach. Exploring the code of a website for SQLIA injection locations Machine learning produces assaults that target these weak points precisely. A variety of ways may be used to identify and prevent SQL injection tools. An attacker can access a web application's database via a vulnerability. As a result, no manual modifications are required. Malicious online programs may compromise a company's systems. An extensive web application may have hundreds of data input locations. The loss of critical data will probably reduce the company's value. They analyze, identify, and prevent SQL injection threats. Application of statistical analysis to determine errors Java Static Tainted data flows are intended to detect tainted data. This approach has a limitation in that it only recognizes well-known patterns. Sure researchers presented SAFELY, a dynamic testing tool for SQLIA vulnerabilities. By handling the MSIL byte code in ASP.NET, SAFELY uses symbolic execution. SAFELY may be able to detect flaws in source code that black-box scanners cannot see. This approach can only be used with Microsoft applications. Using system dynamics analysis, illegal requests may be recognized and refused. Both require the rewriting of a piece of code. Researchers compared the actual questions asked to the predicted questions (programmer intended) in one study. However, parse trees are employed in this method to get the desired outcomes. It also developed a dictionary-based search mechanism for generating application questions. All conceivable searches need a significant amount of storage. This tool may be unable to anticipate database expressions. According to an examination, candidates for DD's "Candidate Assessment" have a separate SQL query setting. By following these methods, SQL injection attacks may be recognized and avoided. When a programmer employs a tool, it generates a query for them. The developer is responsible for detecting and preventing incorrect SQL.

As a consequence of the research, a rule-based SQL injection solution was developed. This is why AMNES was established. This application creates every SQL query. Study At the same time, complex SQL queries are discovered. The SQL queries in risky requests don't match the feedback. There were several false alarms.

Only a few researchers used ontology to define network attacks for distributed intrusion detection systems. In the proposed ontology, victim parameters are allocated to classes. Over 4,000 bugs, exploits, and use cases were examined. They say that ontology provides IDS with a broad range of information. A few individuals suggested creating a set of things with specific security information requirements to identify them. SQLI attacks have previously been simulated, but there is no proof that these models can be used to stop or detect the assaults successfully. Following that is a quick SQLI attack scenario. These ontologies were created using OWL-DL, OWL GUI, and Methodology. The Known Vulnerabilities section covers all attacks. It keeps track of the attackers' methods, the system damage they inflict, the holes they exploit, and the extent to which they are subject to policy control. According to the survey, just a few researchers used Uml-based Web Engineering (UWE). It is an SDLC ontology based on UML for the software development process. Designers may find it helpful in establishing design security guidelines. Using prepared statements makes it feasible to avoid the need for database query injection. It demonstrates a sense of belonging to, utilizing, and depending on something. They altered people's perceptions of the Internet of Things. Recognize and prepare for network and application attacks using NSI (National Security Identification) (NSSA). This flaw was used to identify a subset of SQLI attacks. This technique fails to identify zero-day assaults because it relies on well-known threats.

The authors presented a new framework (HIPS). Threat signatures connect a firewall inspection engine with a machine learning technique. Despite its 97 percent accuracy, HIPS has false positives and false negatives devised a pair of tests. IDS proposed two neural networks. A backpropagation neural network was used to identify seven forms of SQLI truisms and alternate encoding. The authors employed a linear kernel of 13,000 URLs to categorize and assess SQLIA. Almost every time, their model was correct. Their second investigation used the neural network model, URL categorization, and a URL generator. To train and test their model, they employed a superior dataset.

The findings were right 95% of the time. It may produce a new set of network firewall rules. These rules distinguish between legitimate and malicious network traffic. On has presented a method for building and evaluating data sets. Tree parsers should be written in programming syntax. Dynamic queries cannot be conducted without a suitable storage dataset. Two more parse trees are made to ensure that the user's input is reasonable and used correctly. SQL injection attacks may be detected using sophisticated Bayesian machine learning. It organizes unusually well. In this study, we're searching for a brand-new Injection Attack template. A different algorithm would be necessary to identify SQL injection. An ANN firewall is designed to prevent web applications from seeking SKL injections. The system's accuracy ranges from 76.67 percent to 100 percent. However, the relative order of the SQL keywords causes many false warnings. Detecting web-based attacks with unbiased HMM-Web The approach identifies fraudulent requests using patterns. The final evaluation contrasts each HMM's group performance. The device detects 96 percent of viruses, with only 1 percent of false positives. Heuristics (HMMs) are used for the inspection of the HTTP payload. Another difficulty with HMM is the time required to train the model.

A, N, and N are utilized in the query string for the HMM-based method to identify and stop malicious traffic. It exploits a fusion of inferences rather than a threshold. The judging module, which was handcrafted, has a significant problem. HIPS improves web server firewall detection. The naive Bayesian classification approach does design matching. The classifier uses 45 features to classify incoming HTTP requests. There are several false positives and negatives. A system based on ANNs (ANN) A URL generator was employed to produce this to educate the ANN. The method has a success rate of 96 percent and a failure rate of 4 percent. Nothing to do with strategy or overall success. SQL injections in type fields are possible since the solution only evaluates HTTP URLs.

Background Knowledge

In this work, ML-based models such as MLP, SVM, Logistic Regression, Naive Bayes, and Decision Trees are used to assess the performance of SQLI detection[11][12]. First, we'll go through all of the details of the algorithm in this section[13][14]. The Confusion Matrix, the F1 Score, and the Log Loss were displayed as performance matrices. Finally, describe the dataset[15][16][17].

Machine Learning Algorithms

Multi-layer Perceptron

For the most beneficial neural network, artificial neural networks are frequently referred to as multi-layer perceptrons. A perceptron is a solitary neuron that existed before developing more extensive neural networks. Computer science investigates how basic brain models may be used to solve complex computer problems, such as statistical modeling in machine learning. Instead of developing functional brain models, the objective is to build scaling algorithms and statistical frameworks. The ability of neural networks to learn how to apply their training data to the performance variable is powerful. Neural networks learn to route in this scenario. They are a general reasoning approach that can mathematically train any mapping function. Because of their systemic or multi-layered architecture may anticipate that the data structure can detect and integrate features of varying sizes and resolutions. Lines, string arrays, and forms are a few examples. To replicate input-output connections, the multiplayer perceptron is based on input-output pairings. The model's parameters or beginning weights are modified during training to decrease errors. Backpropagation is used to change the weight or distortion of an error (RMSE)[18][19].

Support Vector Machine (SVM)

SVM is a machine learning approach that seeks to lower the overall risk of the system. Performance has improved as a result of these additional features. SVM was created to recognize patterns in binary situations. If you can't separate the two problems, SVM is the kernel space principle you're looking for. SVMs can perform multiclass classification using several binary classifiers[20].

Logistic regression

Non-binary problems can be solved using the logistic regression approach. The non-linear transformation of the linear regression approach predicts categorization with the target group interpreting the conclusion. Using the logistic sigmoid function, you may convert the logistic regression findings into two or more groups [21].

Naïve Bayes

The Naive Bayes theorem and the strong function independence assumptions are used to construct a straightforward probability system. What is the process by which a Nave Bayes classifier determines the conditional probability of each independent factor in an anticipated class? It looks at the values of the autonomous function for each class that has been chosen[22].

Decision Tree

The picture below shows that most decision trees divide data along coordinate axes. As the tree grows, the input vector is likely to be divided into smaller and smaller bits, allowing pattern recognition to occur. On the other hand, overgrown branches result in overfitting[23].

Evaluation Matrix

Confusion Matrix

An N x N matrix represents the number of groups (or divisions) created. If N = 2 and the matrix is 2 X 2, shown in Figure 2, we have a problem with binary grouping. The categorization samples may only be classified as yes or no. As a result, we developed a classifier that can predict the input class of a new dataset[24].

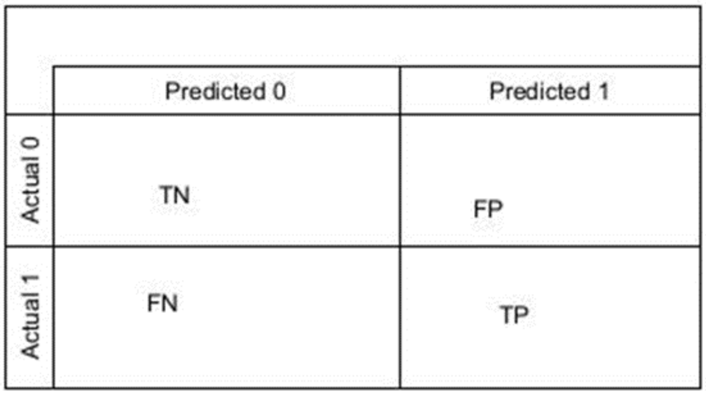

Figure 1: 2 x 2 Matrixes

F1 score

Harmonic the F-recall measures and accuracy are evaluated. It evaluates a classifier's ability to balance recall and accuracy. That is its tensile strength and precision. In general, if the F-measure is close to one, the classifier has become more accurate and more reliable. The primary emphasis of this program is on the recall and f-measure components. As a result, the recall score is used to evaluate its sensitivity. Churner detection would be resistant to high recall rate classifiers. Precision is a measure of how accurate a classifier is. It determines whether or not the predicted positive percentage is correct[25].

Logarithmic loss

The essential propagandistic strategy is to punish inaccurate (false positive) classifications as much as possible. It typically works well when it comes to multiclass grouping. When utilizing log failure, the classifier may return a probability for each data class throughout the whole dataset. While classification accuracy is excellent, viewing it through the lens of high accuracy is wrong. It is a problem because there is a good chance that small sample misclassifications will happen again.

Datasets

The dataset comprises four files: sql.csv, sqli.csv, username.csv, and password.csv.

- SQL.CSV: The SQL dataset is made up of SQL queries with only non-malicious data. This dataset will be used to detect accurate, non-fraudulent data.
- SQLI.CV: This dataset provides a list of all SQL queries that contain harmful data.
- USERNAME.CSV: A list of the most frequently used SQL injection usernames is contained in this dataset.

- PASSWORD.CSV: A list of the most commonly used SQL injection passwords is contained in this dataset.

Methodology

In Figure 2, We begin by detecting and restoring all missing values across all datasets. Pandas (a Python library) are being used to do this. We add a "length" field in the username and password collection. The "length" column displays the length of the login and password in characters. The size of the query column is also considered. SQL Injection seldom employs an enormous query. SQL injection is improbable since lengthy usernames and passwords are infrequent. In all datasets, rename "Label" columns to their column headings. This generates a single pivot point for combining datasets. Unnecessary columns. The attack column is no longer in use. As a result, we deleted it. Then combine the datasets. We merged all of the attributes into a single dataset. The index for this dataset has been reset. This dataset will be used for all future categorized reports. After the combination, the final dataset is shown in Table 1: Combined dataset t. Then combine the datasets. We merged all of the attributes into a single dataset. The index for this dataset has been reset. This dataset will be used for all future categorized reports. Punctuation in Questions and AnswersThe dataset is now free of punctuation. Punctuation is unnecessary and does not help in classifying. As soon as a phrase is tokenized, all punctuation disappears. The process is repeated for each term. The phrase ['Think and ponder, wonder and think.' tokenized and punctuation marks removed] is ['Think', 'and', 'wonder', 'wonder', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and', 'and'. The punctuation marks being dropped are:['!', "," ,"\'" ,";" ,"\"", ".", "-" ,"?","[","]",")","("]. The dataset after punctuations are is removed shown in Table 2.

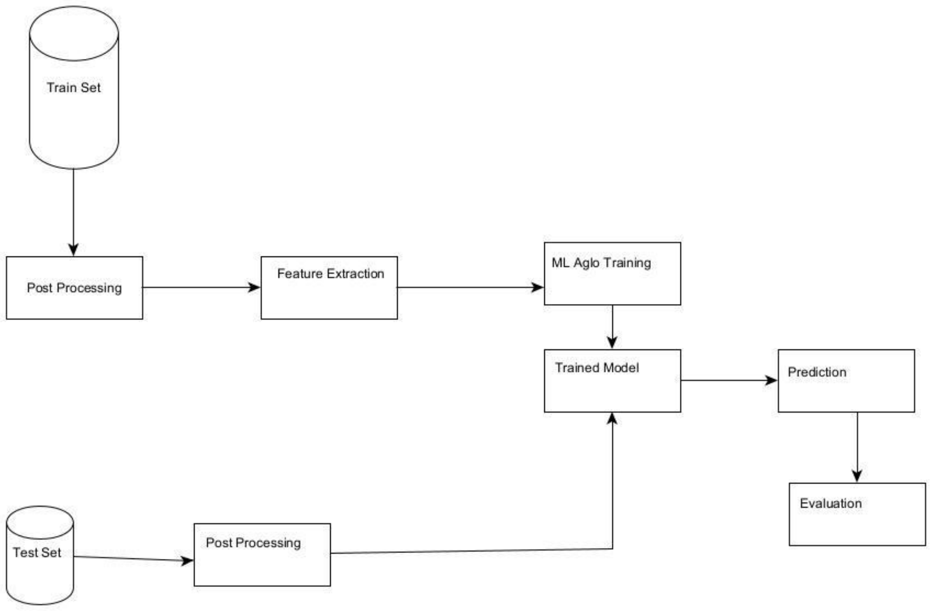

Figure 2: Methodology

Table 1: Combined dataset

Query	Length	Label
0. 1* Where 6406=6406;select count(*) from rdb$fi....	115.00	sqli
1 1) and 8514=(select count(*) from domain.domai....	111.00	sqli
2 -3136%*) or 3400=6002	21.00	sqli
3 1) where 7956=7956 or sleep(5)#	31.00	sqli
4 -7387*))) order by 1--	22.00	sqli

Table 2: Dataset with punctuations removed

Query	Length	Label	Punctuation
0. 1 Where 6406=6406;select count(*) from rdb$fi....	115.00	sqli	9
1 1 and 8514=select count * from domain.domai....	111.00	sqli	11
2 3136% or 3400=6002	21.00	sqli	3
3 1 where 7956=7956 or sleep 5 #	31.00	sqli	3
4 7387 order by 1	22.00	sqli	7

The keywords are then counted. All SQL keywords detected in the query field are: select, update, create, drop, alter, rename, exec, order, group, sleep, and count. This feature will aid in the detection of SQL injection requests. The dataset after calculating the keywords is shown in Table 3. Then we classify the "label" vector. Identifying SQL injection queries vs. non-injection queries A non-sqli label does not contain a SQL injection query, while a sqli label has a SQL injection query. The dataset after label encoding is shown in **Error! Reference source not found.**. Standardization of Encoding and Coding (FEATURE SCALING). In machine learning, feature scaling is a fundamental data preparation strategy. If the data is not scaled, employing equations that provide distance values close to one results in a bias toward small numbers. Feature-driven methods are not scale-dependent. Deep learning and machine learning systems benefit from feature scaling as well. Standardization and normalization are used in function scaling. Z-Score This adjustment removes characteristics before calculating the distribution based on the deviation from the mean. The creator of this story coined the term "Z-Score." You're The result of standardization is a Gaussian distribution.

Table 3: Dataset with punctuations removed

Query	Length	Label	Punctuation	Keyword
0. 1 Where 6406=6406;select count(*) from rdb$fi....	115.00	sqli	9	3
1 1 and 8514=select count * from domain.domai....	111.00	sqli	11	2
2 3136% or 3400=6002	21.00	sqli	3	0

3 1 where 7956=7956 or sleep 5 #	31.00	sqli	3	2
4 7387 order by 1	22.00	sqli	7	1

However, it is not essential. Usually, it enlarges or compresses features and alters the meaning of the origin. We've essentially converted the means and standard deviations to a natural distribution. There has been no modification to the distribution's parameters. Outliers should be ignored. In machine learning, we handle datasets that involve multiple discrete classes in one or more columns. Term definitions or numerical symbolsTraining data is frequently given in easily understood ways. The encoding consists in converting the marks to a machine-readable representation. Algorithms that employ learning algorithms will decide how to use the labels. It's a crucial step before autonomous learning. The dataset after standardizing and label encoding are shown in Table 4.

After standardization (FEATURE SCALING) and encoding, the text is vectorized. Text vectorization converts text into a numerical representation. A few typical text vectorization methods, namely Frequency of Binary Term, Words in a Bag (BoW), (L1) Frequency of Term, (L1) TF-IDF Normalized, and Word2Vec. The "Query" field will now be vectorized using NLTK, and the mean of the vectors will be calculated. The size of the vector is 4096. The dataset after vectorization is shown in Table 4. Train Test Split follows vectorization. Sets 1 and 2 are called "Train and Test." 80% of the dataset is for training and 20% for testing. Creating Tensorflow dataset Using slices, we generate a dataset appropriate for Tensorflow. We then shuffle the dataset with batch size = 64. Last, use Text Embedding. Many of the TFHub pre-trained text embeddings can be used. Word embedding is a recognized format for texts having similar phrases in words. This approach is one of the most significant deep learning advancements in natural language processing. Individual words in a preset vector context are interpreted as real-world vectors. Each phrase is converted to a vector, and the vector value is learned similarly to a neural network. For this, we utilized the "gnews-swivel-20dim" text embedding.

Table 4: Dataset after vectorization

Query	Length	Label	Punctuation	Keyword
0 0.005340	115.00	1	9	3
1 0.004673	111.00	1	11	2
2 0.000334	21.00	1	3	0
3 0.000334	31.00	1	3	2
4 0.000334	22.00	1	7	1

Result Analysis

The subjects of this inquiry are MLP, SVM, Logistic Regression, Naive Bayes, and Decision Trees shown in Figure 3. Each of these five machine learning algorithms uses a distinct set of cutoff criteria for analyzing data. The Confusion Matrix, F1 Score, and Log Loss are all different types of cutoffs. Table 5 shows the F1 outcome as well as the log loss. In this situation, the SVM surpasses both the F1

Score and the Log Loss, as seen in Table 5. Table 5 shows that SVM has a higher F1 score of 0.9986 and a log loss value of 0.08, the least of all potential losses. An F1 score is one of the most visible constraints. The confusion matrix for each of the five algorithms is depicted in Figure 3. According to Confusion, the Class-I Matrix SVM has a true positive rate of 2187.

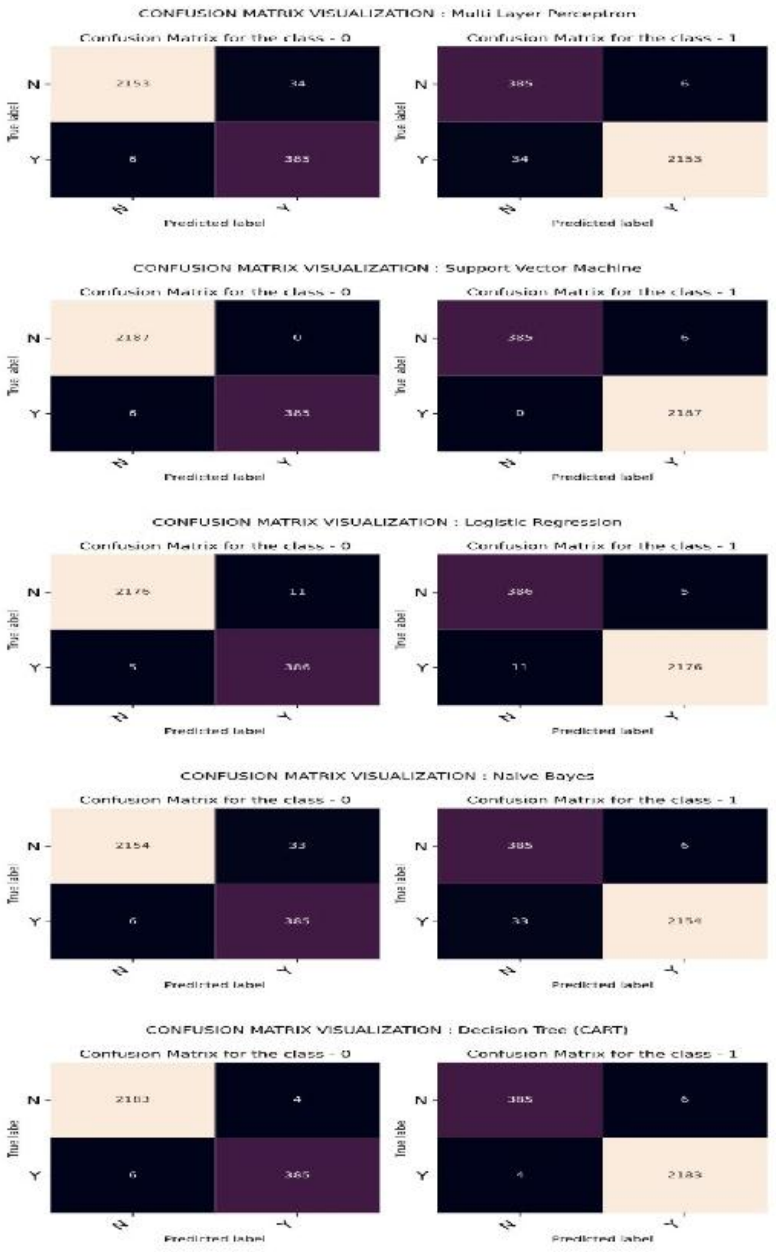

Figure 3: Confusion Matrix Visualization

Table 5: LOG LOSS

Model	Log Loss	F-Score
MLP	0.53	0.9907
Support Vector Machine	0.08	0.9986
Logistic Regression	0.21	0.9963
Naïve Bayes	0.52	0.991
7Decision Tree	0.13	0.9977

Conclusion

SQL injection attacks are a significant cause of worry among computer security specialists. Due to attackers' ongoing advancement of SQL injections, signature-based SQL injection detection solutions have become obsolete. SQL Injection detection technologies that are capable of detecting new threats are required. A substantial number of cyber-security researchers are considering machine learning. Because machine learning is still in its infancy in cyber-security, a few machine learning libraries and open-source resources may be used to minimize risks. Machine learning algorithms are employed in this study to detect SQL Injection attacks. The system categorizes incoming traffic as either SQL Injection or plain text. The following machine learning methods are used to handle the topic: MLP, SVM, LR, Naive Bayes, and DT. The most commonly utilized approach is MLP, which SVM and LR follow. SVM obtains the highest F1 Score, 0.9986, while having the lowest log-loss, 0.08. The SQL injection is detected using the SVM Classifier. According to the findings of this study, machine learning approaches such as SVM outperform MLP in seeing SQL injection. This project's usefulness and general quality should be increased in the future. SQL injections may be identified using static code analysis with web browser firewalls, which can then be combined with machine learning. Improved function extraction can boost the machine learning model's performance. Tokenization is employed in the development of this project's machine learning model. There are also alternative methods for obtaining information and developing the model.

References

1. H. Hanif, M. H. N. Md Nasir, M. F. Ab Razak, A. Firdaus, and N. B. Anuar, "The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches," J. Netw. Comput. Appl., vol. 179, no. August 2020, p. 103009, 2021, doi: 10.1016/j.jnca.2021.103009.
2. M. Baş Seyyar, F. Ö. Çatak, and E. Gül, "Detection of attack-targeted scans from the Apache HTTP Server access logs," Appl. Comput. Informatics, vol. 14, no. 1, pp. 28–36, 2018, doi: 10.1016/j.aci.2017.04.002.
3. T. He, Y. Zheng, and Z. Ma, "Study of network time synchronisation security strategy based on polar coding," Comput. Secur., vol. 104, May 2021, doi: 10.1016/j.cose.2021.102214.
4. P. Nimbalkar and D. Kshirsagar, "Feature selection for intrusion detection system in Internet-of-Things (IoT)," ICT Express, vol. 7, no. 2, pp. 177–181, 2021, doi: 10.1016/j.icte.2021.04.012.

5. R. T. Kokila, S. Thamarai Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in 6th International Conference on Advanced Computing, ICoAC 2014, Aug. 2015, pp. 205–210, doi: 10.1109/ICoAC.2014.7229711.
6. K. Natarajan and S. Subramani, "Generation of Sql-injection Free Secure Algorithm to Detect and Prevent Sql-Injection Attacks," Procedia Technol., vol. 4, pp. 790–796, 2012, doi: 10.1016/j.protcy.2012.05.129.
7. S. Gupta, J. Sarkar, M. Kundu, N. R. Bandyopadhyay, and S. Ganguly, "Automatic recognition of SEM microstructure and phases of steel using LBP and random decision forest operator," Measurement, vol. 151, p. 107224, Feb. 2020, doi: 10.1016/j.measurement.2019.107224.
8. K. Stokes et al., "A machine learning model for supporting symptom-based referral and diagnosis of bronchitis and pneumonia in limited resource settings," Biocybern. Biomed. Eng., vol. 41, no. 4, pp. 1288–1302, 2021, doi: 10.1016/j.bbe.2021.09.002.
9. S. M. Darwish, "Machine learning approach to detect intruders in database based on hexplet data structure," J. Electr. Syst. Inf. Technol., vol. 3, no. 2, pp. 261–269, 2016, doi: 10.1016/j.jesit.2015.12.001.
10. A. R. Panhalkar and D. D. Doye, "Optimization of decision trees using modified African buffalo algorithm," J. King Saud Univ. - Comput. Inf. Sci., no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.01.011.
11. B. Mondal, "Artificial Intelligence: State of the Art," in Intelligent Systems Reference Library, vol. 172, 2020, pp. 389–425.
12. T. Marwala, "Multi-layer Perceptron," in Handbook of Machine Learning, WORLD SCIENTIFIC, 2018, pp. 23–42.
13. S. Gupta, J. Sarkar, A. Banerjee, N. R. Bandyopadhyay, and S. Ganguly, "Grain Boundary Detection and Phase Segmentation of SEM Ferrite–Pearlite Microstructure Using SLIC and Skeletonization," J. Inst. Eng. Ser. D, vol. 100, no. 2, pp. 203–210, Oct. 2019, doi: 10.1007/s40033-019-00194-1.
14. S. Gupta et al., "Modelling the steel microstructure knowledge for in-silico recognition of phases using machine learning," Mater. Chem. Phys., vol. 252, no. May, p. 123286, Sep. 2020, doi: 10.1016/j.matchemphys.2020.123286.
15. B. Turkoglu and E. Kaya, "Training multi-layer perceptron with artificial algae algorithm," Eng. Sci. Technol. an Int. J., vol. 23, no. 6, pp. 1342–1350, 2020, doi: 10.1016/j.jestch.2020.07.001.
16. K. Goswami and G. L. Samuel, "Support vector machine regression for predicting dimensional features of die-sinking electrical discharge machined components," Procedia CIRP, vol. 99, pp. 508–513, 2021, doi: 10.1016/j.procir.2021.03.109.
17. I. H. Sarker, "CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks," Internet of Things, vol. 14, p. 100393, Jun. 2021, doi: 10.1016/j.iot.2021.100393.
18. E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," Heliyon, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.
19. Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," Sensors (Switzerland), vol. 20, no. 16, pp. 1–15, Aug. 2020, doi: 10.3390/s20164372.

20. Y. Sasaki, "The truth of the F-measure," Teach Tutor mater, pp. 1–5, 2007.
21. R. Guns, C. Lioma, and B. Larsen, "The tipping point: F-score as a function of the number of retrieved items," Inf. Process. Manag., vol. 48, no. 6, pp. 1171–1180, 2012, doi: 10.1016/j.ipm.2012.02.009.