International Journal of Artificial Intelligence Research
Vol 6, No 2, December 2022
ISSN: 2579-7298

Detection of SQL Injection Attack Using Machine Learning Based on Natural Language Processing

Joko Triloka[a,1a], Hartono[a,2] Sutedi[a,3]

[a] *Institut Informatika dan Bisnis Darmajaya, Indonesia*
[1]*joko.triloka@darmajaya.ac.id*;* [2] *harton.2021210011@mail.darmajaya.ac.id**[3] *sutedi@darmajaya.ac.id*

ARTICLE INFO

Article history:
Received 12 June 2022
Revised 16 August 2022
Accepted 25 August 2022

Keywords:
Cyber Security
Machine Learning
Support Vector Machine
SVM Based NLP
Multi-Layer Security

ABSTRACT

There has been a significant increase in the number of cyberattacks. This is not only happening in Indonesia but also in many countries. Thus, the issue of cyber attacks should receive attention and be interesting to study. The Open Web Application Security Project has published the Top-10 website vulnerabilities regarding the explored security vulnerabilities. SQL Injection is still one of the website vulnerabilities that attackers often exploit. This research has implemented and tested five algorithms. They are Naïve Bayes, Logistic Regression, Gradient Boosting, K-Nearest Neighbor, and Support Vector Machine. In addition, this study also uses natural language processing to increase detection accuracy as a part of text processing. Therefore, the primary dataset was converted to the corpus to make it easier to be analyzed. This process was carried out in the feature engineering stage. This study used two datasets of SQL Injection. The first dataset was used to train the classifier, and the second was used to test the classifier's performance. Based on the tests, the Support Vector Machine gets the highest level of accurate detection. The detection accuracy is 0.9977 with 0,00100 microseconds per query time of the process. The Support Vector Machine classifier can detect 99,37% of the second dataset in performance testing. Not only Support Vector Machine, but the study has also revealed the detection accuracy level of further tested algorithms: K-Nearest Neighbor (0,9970), Logistic Regression (0,9960), Gradient Boosting (0,99477), and Naïve Bayes (0,9754).

I. Introduction

Based on PacketLabs, there will be a significant increase in cyberattacks in 2021, and these attacks will occur in many countries worldwide [1]. Google records 18 million malware attack attempts per day [2]. Illegal website hacking also reaches 30,000 times daily, and as many as 64% of companies in the world experience at least one attack attempt. As of March 2021, 20 million records were accessed illegally, and cyberattacks continue to occur every 39 seconds [3]. According to a report from Purplesec, around 18 million websites are infected with malware every week [4]. In the case of Indonesia, the BSSN (State Cyber and Code Agency) of Indonesia states that 5,934,058 techniques have not been identified by 2020 [5]. The diversity of technologies, processes, and attack strategies affects the attack identification level. In addition, detection of attack symptoms is also not easy because it depends on the technology (software and hardware) available or used by the server computer. Without reliable attack detection technology, the mitigation process automatically becomes challenging. Therefore, the two components have to work in synergy. Thus, it could develop a robust, stable, reliable cybersecurity system [6].

Due to the diverse and complex types of cyber attack methods, the attack methods tested refer to the OWASP Top-10 Common Web Application Vulnerabilities published by the Open Web Application Security Project (OWASP) [7]–[9]. OWASP Top-10 vulnerabilities list the most common security vulnerabilities found in most web-based applications (See Figure 1). In other words, the security gap is the component most often used as a target for attacks. The attack methods

used to exploit these common vulnerabilities may vary in this regard. As an illustration, the attack method used in the injection security gap is SQL Injection (SQLi). In this case, related to OWASP Top-10 Vulnerabilities, SQL Injection is a part of injection vulnerabilities. The most challenging part of cyber attack detection is detecting insider attacks, which are usually seen after the attack is successfully carried out [10]. Based on some data, 70% of illegal hacking is committed from inside rather than outside, but 90% of security controls and oversight are focused on external threats [11]. To perform anomaly detection, machine learning-based systems can be used. The detection system would trigger an alarm when an object or component behaves differently from a predetermined regular pattern. Therefore, the use of machine learning is highly recommended. The machine learning, Dua defines machine learning as a computational process that infers and generalizes a learning model from a given dataset or sample [12].

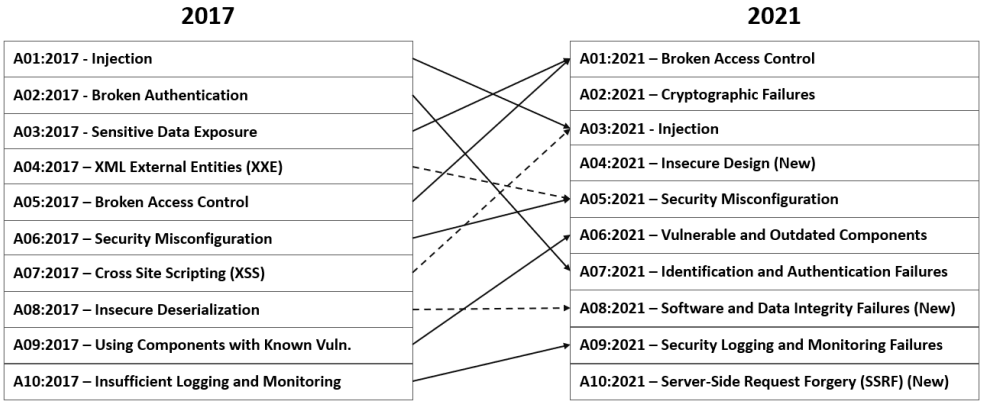

Fig. 1.OWASP Top-10 Web Vulnerabilities 2017 to 2021

Cherry states that SQL Injection is an attack carried out via modified SQL queries and sent via browser queries. This type of attack can occur on website-based applications that use Active Server Pages (ASP) or Hypertext Preprocessor (PHP) and use SQL-based data [13], [14]. Kavitha defines SQL Injection as an action when an attacker sends malicious SQL codes to a website application [15], [16]. The attacker can access the database server when SQL malicious codes are successfully executed. Ahmad and Karim explained SQL Injection as a method to steal or access databases illegally [16]. They also stated that SQLi is an act of sending malicious codes to web applications, and it uses the database to execute specific commands. Ogundijo said the SQLi attack uses website form or input to send SQL commands. After successfully attacking, the attacker can access the database server [17]–[19].

Regarding the risk, Hirani *et al.* stated that an SQLi attack could be hazardous [20]. In some classification, SQLi attack becomes web vulnerabilities with high severity. Roy *et al.* used Naïve Bayes to detect SQLi attacks with 98,33% accuracy [21].

Related to the previous studies that have been carried out, several researchers use machine learning to detect SQLi attacks. In the case of those earlier studies, the accuracy level obtained by each researcher and algorithms differs. Hashem [22] simultaneously implemented the detection of SQLi attacks. Jemal [23], in his research, has tested nine algorithms to detect SQLi attacks. The algorithms are Naive Bayes, Back Propagation Neural Network, Neural Network Based Model, Neural Network, Decision Tree, Multi-Layer Neural Network, Support Vector Machine, K-Nearest Neighbor, and TBD-NNBr. The accuracy level obtained for each algorithm tested by Jemal can be seen in table 1. In addition, Hasan et al. [24] used Heuristic Algorithm with a 93.8 accuracy level. In quite a different algorithm, Kranthikumar [25] used a REGEX classifier with a 97% of detection accuracy rate.

Table 1. Jemal's Research SQLi Detection Accuracy Level

	Algorithm	***Accuracy Level of Detection***
1	Naïve Bayes	97,6%
2	Back Propagation Neural Network	95,8%
3	Neural Network Based Model	95%

	Algorithm	*Accuracy Level of Detection*
4	Neural Network	98,6%
5	Decision Tree	83,7%
6	Neural Network Multi-Layer	66,67%
7	Support Vector Machine	96,23%
8	K-Nearest Neighbor	97%
9	TbD-NNbR	99,23%

II. Methods

A. Procedures and Optimization

There are five stages carried out in producing the detection method. First, the dataset is prepared according to needs at the preparation stage, starting from crawling, merging, and so on. The second stage is data preprocessing and data modeling. At this stage, there are several general actions taken. They are (a) eliminating empty or incomplete dataset rows, (b) eliminating duplicate data, and (c) performing data conversion as needed. The third stage is to perform feature engineering, data training, and data testing. At the feature engineering stage, the actions taken are (a) corpus construction and (b) feature construction (see figure 2). The detection method uses the help of the Python NLTK library so that each row of the dataset is converted into a corpus before training and testing. After the classifier is generated, performing performance optimization is next.

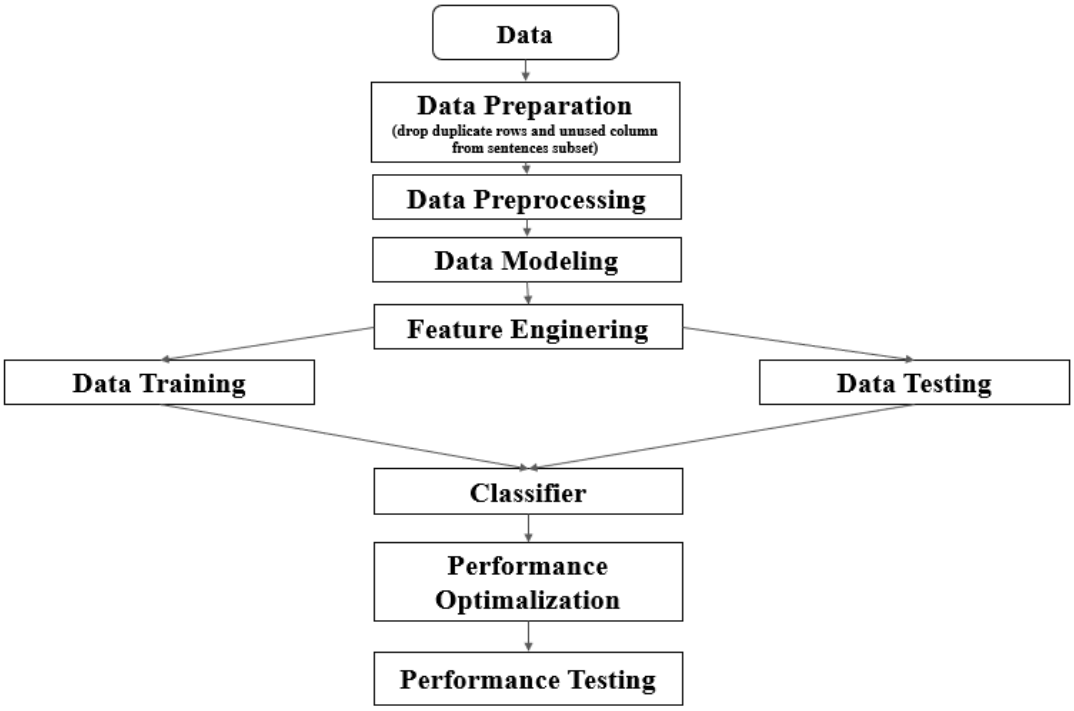

Fig. 2. Schematic of Stages of Detection Method Implementation

Each algorithm is tested for getting detection accuracy levels. There are five detection algorithms tested, namely (a) Naïve Bayes; (b) Logistic Regression; (c) Gradient Boosting; (d) Support Vector Machine; and (d) K-Nearest Neighbor. Researchers choose a detection algorithm based on the highest level of accuracy. Then, the researcher also conducted performance testing. That is, the algorithm with the highest level of accuracy is tested to detect new or different dataset rows from the dataset used in the training and testing process. As stated in the previous explanation, feature engineering uses corpus rules. In this case, there are three corpus parameters used they are (a) lower case conversion, (b) alphanumeric filter, and (c) punctuation removal.

B. Machine Learning Algorithms

This study uses NLP-based machine learning. Text recognition or attack patterns are carried out in this machine learning using Python NLTK. After each algorithm is tested, the detection method will choose the algorithm that can achieve the highest accuracy level. This experiment tested five classification algorithms to get the detection method with the highest level of accuracy. The following are the formula used in algorithms. The following formula is Naïve Bayes.

$$\mathbf{P(label|feature)} = \frac{\mathrm{P}(label) * \mathrm{P}(\mathrm{f1}|label) * \ldots * \mathrm{P}(\mathrm{fn}|label)}{\mathrm{P}(features)}$$

This algorithm will make nave assumptions, where all the features that have been determined are seen as independent. In that context, each feature is associated or correlated with the label that appears. The algorithm then implicitly calculates P(feature) so that this algorithm will calculate the numerator for all brands (payload or non-payload). In addition, the following formula used in this study is Logistic Regression.

$$\mathbf{Y} = \frac{\exp\ (\mathrm{B}_0 + \mathrm{B}_1\mathrm{X})}{(1 + \exp\ (\mathrm{B}_0 + \mathrm{B}_1\mathrm{X}))}$$

Logistic regression classification models discrete target variables as a function of multiple feature variables. This classification uses a discrete y variable. For each observation, the probability that y = 1 is modeled as a logistic function over a linear combination of feature values. A label yi will follow the set of features xi. Logistic regression will interpret the probability that the label is in one class as a logistic function of the combination of features. In addition, Gradient Boosting has several types of bases. This experiment uses a decision tree-based Gradient Boosting. This algorithm processes the dataset sequentially because it adds the previous predictor to the ensemble data. With this pattern of work, previous prediction errors are corrected. In this case, the ensemble is defined as a list of predictive decisions generated by machine learning. The dominant class predicts each row of data. The following formula is Gradient Boosting.

$$-\log L1 = -\sum_{i=1}^{N} y_i \log(odds) + \log(1 + e^{\log(odds)})$$

Support Vector Machine is an algorithm used to determine the decision boundary. The decision boundary determines the classification of this algorithm. The Support Vector Machine utilizes a linear model as a decision boundary. The general form of this process is as follows.

$$\boldsymbol{y(x) = w^T \emptyset(x) + b}$$

Based on formula 8, w is a weight parameter, (x) is a primary function, and b is a bias. The simplest linear model for the decision boundary is y(x) = wtx + w0, where x is a vector, w is a weight vector, and w0 is a bias. Thus, the decision bounty is defined as y(x) = 0, a dimensionless hyperplane (D-1). Meanwhile, the last algorithm tested is K-Nearest Neighbor. This algorithm uses the Euclidean Distance method to see the closest predicted distance on the defined labels. In general, the Euclidean Distance formula can be described as follows.

$$\mathrm{dis}\ (\mathrm{x}_1, \mathrm{x}_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})2}$$

C. Evaluation of Detection Method

In addition to using the confusion matrix to further improve the quality of algorithm testing in detecting attack patterns or vectors, researchers take two actions: performance optimization and performance testing. In the performance optimization stage, the researcher tested the five algorithms

with different margins, corpus rules, and configuration parameters to obtain a high detection accuracy. Meanwhile, in the performance testing stage, the selected detection algorithm with the highest level of accuracy is tested for its detection quality on different datasets. Unlike most machine learning patterns which generally only rely on the level of accuracy in the training and data testing process, this study applies performance testing to test whether the detection method can detect "foreign" attack vectors from the datasets used in the training process. and testing. The dataset used at this stage is called the challenge dataset.

III. Results and Discussion

A. Dataset Characteristics

The total number of SQLi datasets used is 30,904-row datasets. After preprocessing data, there are 295 data lines that are eliminated, so the number of eligible data used in the training and testing data process is 30,609. Similar to the dataset in the XSS attack technique, two labels are used in the SQLi dataset, namely payload, and non-payload. Of the total 30,906 rows of the dataset, the number of rows of data with payload labels is 11,341, and data with non-payload labels is 19,268.

B. Configuration of Detection Parameters

Optimization and evaluation of the performance of detection methods on SQLi attack techniques are carried out in 20 stages. Each stage tested five algorithms. In other words, 100 algorithm testing times have been carried out with different parameter configurations. This process is carried out by setting the margin parameter configuration and corpus rules.

Table 2. Optimal Parameter Configuration on SQLi Attack

	Algorithm	***Accuracy Level of Detection***
1	Margin	6.5
2	Test Size	0.2
3	Lowercase Conversion	True
4	Alphanumeric Filter	False
5	Punctuation Remover	False

C. Feature Set Injection

In an SQLi attack, the maximum number of vectors with a margin of 6.5 is 104. In the feature set injection stage, as many as 104 character vectors on the payload and non-payload labels are collected into a feature set. The feature set injection used to detect SQLi attacks can be seen in the following figure.

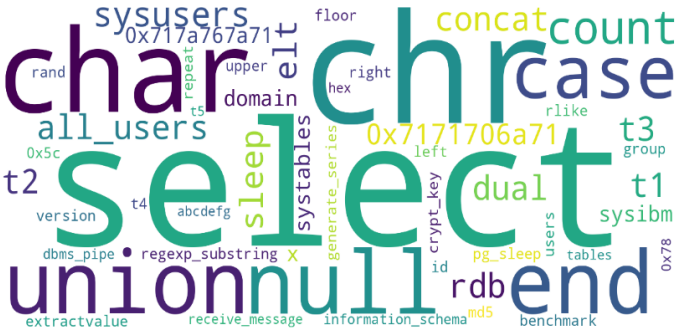

Fig. 3. SQLi Label Payload Vector Visualization Using Wordcloud

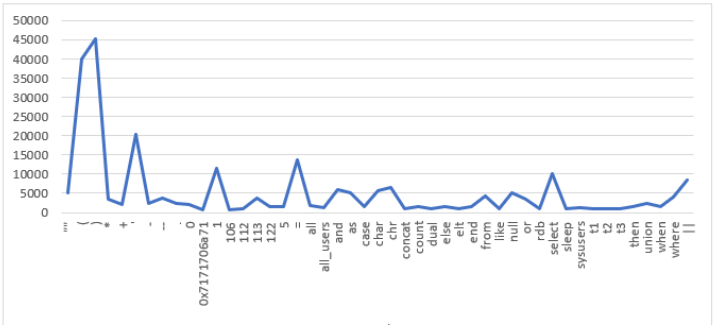

Fig. 4. SQLi Vector Distribution Graph on Payload Labels

Figure 3 shows a list of vectors used to perform SQLi attacks and still related to the previous image, which offers 50 vectors accompanied by the number of occurrences in the row of the dataset. Meanwhile, the vector used in the payload label can be seen in the following figure.

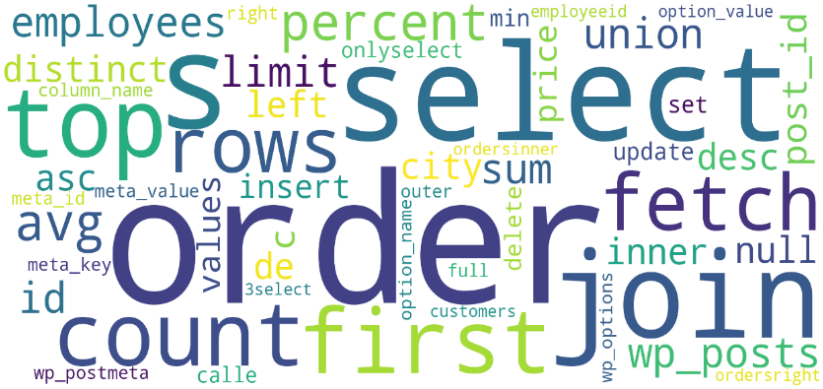

Fig. 5. SQLi Label Non-Payload Vector Visualization Using Wordcloud

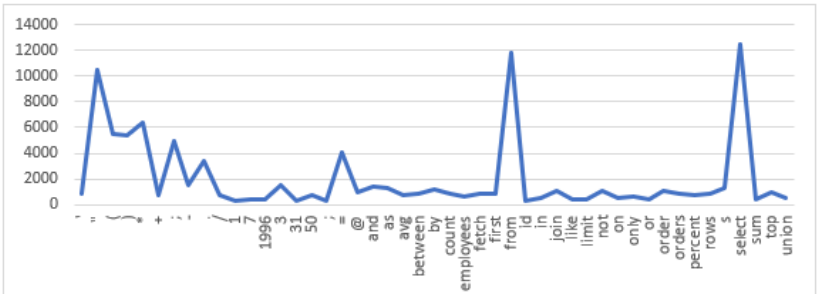

Fig. 6. SQLi Vector Distribution Graph on Non-Payload Labels

D. Feature Set Injection

Based on the experiments, SVM becomes an algorithm that can detect with the highest level of accuracy, which is 0.99771. In addition, the top generated is also speedy because it is still under microseconds, which is 0.00100 microseconds per query. With this accuracy and ToP, SVM has proven reliable in detecting SQLi attacks.

Table 3. Level of Accuracy and Top Classifier in SQLi Attacks

	Classifier Name	*Accuracy*	*Top of Learning*	*Top/Query*
1	Support Vector Machine	0,997713166	0:00:10.713473	0,001008
2	K-Nearest Neighbors	0,997059784	0:01:23.933334	0,055445
3	Logistic Regression	0,996079713	0:00:04.485225	0,000444
4	Gradient Boosting	0,99477295	0:00:13.379335	0,000652
5	Naive Bayes	0,975498203	0:00:04.138092	0,000415

E. Confusion Matrix, Accuracy Visualization, and ToP

SVM can achieve the highest accuracy level compared to other algorithms. With an accuracy rate of 0.997713, SVM can detect SQLi attacks more accurately. In addition, the top algorithm is also reasonably fast because it only requires 0.00100 microseconds. The following is an SQLi attack data confusion matrix using SVM.

Table 4. Confusion Matrix SQLi Attack Detection Using SVM

	Nama Classifier	***Non-Payload***	***Payload***
1	Non-Payload	<3890>	1
2	Payload	13	<2218>

Table 5. SQLi Detection Precision, Recall, and F-Measure Details

	Nama Classifier	***Non-Payload***	***Payload***
1	Non-Payload	<3890>	1
2	Payload	13	<2218>

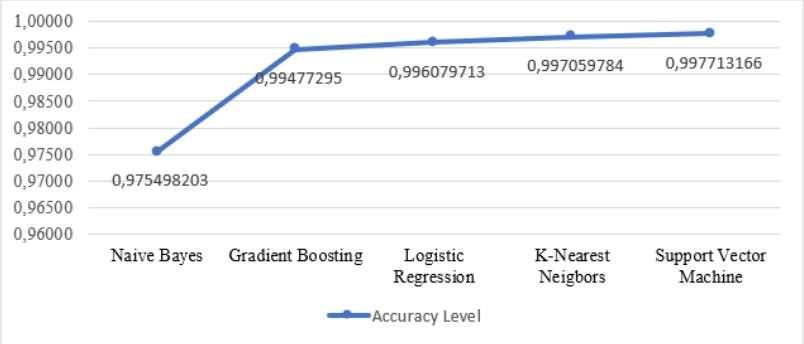

Fig. 7. The Accuracy Level of Each Algorithm

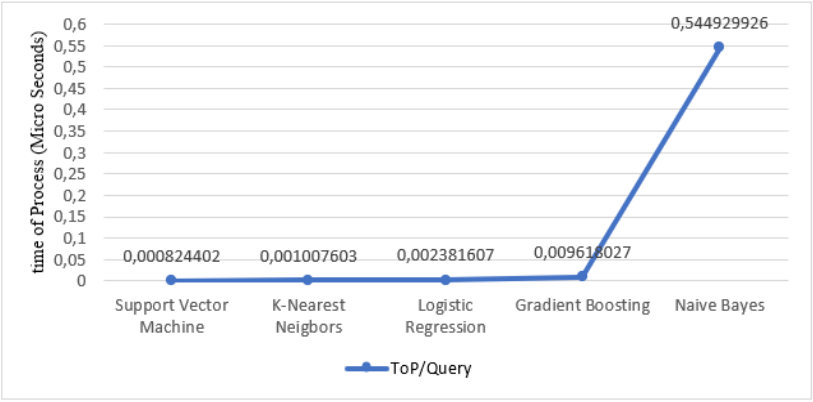

Fig. 8. Time of Process of Each Algorithm

F. Performance Optimization

SVM is the algorithm of choice in the SQLi attack detection method. The choice of this algorithm is, of course, because the basis is clear and firm. Based on the research stages that have been carried out, SVM is proven reliable and achieves a very high level of accuracy. The lowest accuracy rate of SVM during the optimization stage was 0.89. After several optimization steps were carried out, together with other algorithms, SVM achieved the highest level of accuracy, which was 0.9977 with a margin of 6.5. With a margin of 6.5, the required ToP is 0.001. Accuracy optimization and SVM Topp stages can be seen in the following figure.

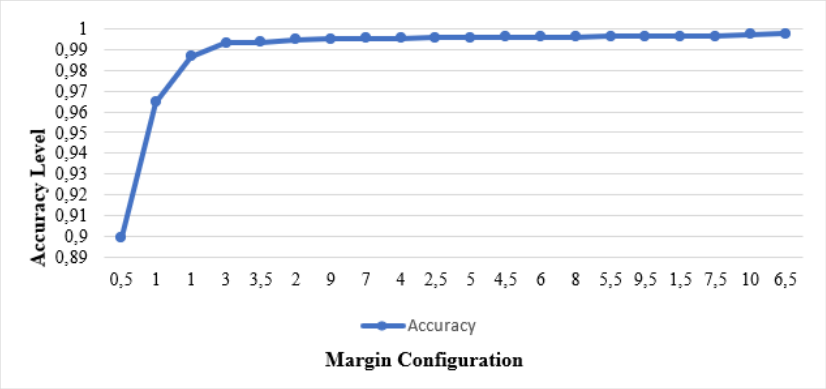

Fig. 9. Stages of Optimizing SQLi Detection Accuracy Using SVM

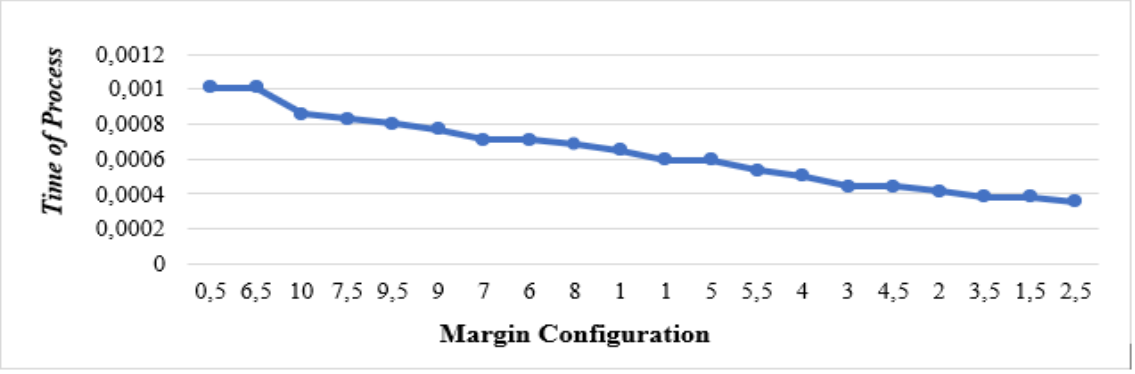

Fig. 10. Stages of Optimizing SQLi SVM Detection Method Accuracy

G. Performance Testing

Given the reality of today's cyberattacks, the patterns and forms of SQLi attacks can vary widely. SQLi attack detection methods based on machine learning must continue to be tested for their performance to produce a detection method that is stable and consistent with its level of accuracy. Therefore, in addition to strengthening the learning process through the stages of comprehensive data training and testing and parameter switcher configuration, this study also applies the stages of performance optimization, which are referred to as challenges. At this stage, the detection method that has been produced is tested for its performance in detecting different datasets.

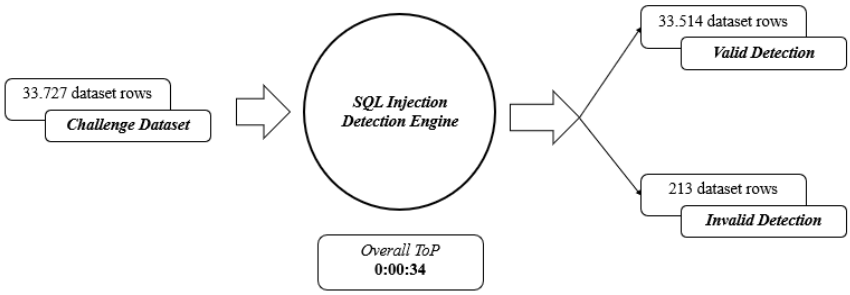

Fig. 11. Implementation of SQLi Detection Method on Challenge Dataset

The number of dataset challenges tested on the selected SQLi attack detection method amounted to 33,727 dataset rows. Data preprocessing stages are also carried out on the difficulties dataset. In addition to trying the chosen detection method, this stage also tests other algorithms to obtain performance comparisons and ToP data for each algorithm. The selected SVM algorithm accuracy is

99.37%. The detection method is further integrated with multi-layer security mitigation methods to increase website applications' security from SQLi attacks.

IV. Conclusion

Based on the results of tests that have been carried out on five algorithms: (a) Naive Bayes; (b) Logistic Regression; (c) Gradient Boosting; (d) Support Vector Machine; and (e) K-Nearest Neighbor, SVM can produce the highest level of accuracy. SVM can achieve an accuracy of 0.9977 with a ToP of 0.00100 microseconds. In addition, this study has also revealed the detection accuracy level of further tested algorithms: K-Nearest Neighbor (0.9970), Logistic Regression (0.9960), Gradient Boosting (0.99477), and Naïve Bayes (0.9754). Based on ToP per query, K-Nearest Neighbor becomes an algorithm with the slowest processing time, 0,5449 seconds. About the previous research, the highest accuracy level of SQLi detection by Jemal, which uses TbD-NNbR, is 99,23% (see table 1) [23]. For researchers who want to do the same research, I recommend optimizing training and testing data selection. The selection is expected to increase the level of accuracy further.

References

[1] "Cybersecurity Statistics for 2021," *Packetlabs*, Aug. 03, 2021. https://www.packetlabs.net/cybersecurity-statistics-2021/ (accessed Nov. 05, 2021).

[2] "Protecting against cyber threats during COVID-19 and beyond," *Google Cloud Blog*. https://cloud.google.com/blog/products/identity-security/protecting-against-cyber-threats-during-covid-19-and-beyond/ (accessed Nov. 05, 2021).

[3] "How Many Cyber Attacks Happen Per Day? [2021 Stats and Facts]," *TechJury*, Jul. 15, 2020. https://techjury.net/blog/how-many-cyber-attacks-per-day/ (accessed Nov. 05, 2021).

[4] "2021 Cyber Security Statistics Trends & Data," *PurpleSec*, Nov. 08, 2020. https://purplesec.us/resources/cyber-security-statistics/ (accessed Nov. 05, 2021).

[5] A. Yusuf, *Laporan Tahunan 2020 Honeynet Project BSSN - IHP*. Badan Siber dan Sandi Negara, 2020.

[6] B. Akhgar, A. Staniforth, and F. Bosco, "Cyber Crime and Cyber Terrorism Investigator's Handbook," p. 399.

[7] OWASP, "A04 Insecure Design - OWASP Top 10:2021." https://owasp.org/Top10/A04_2021-Insecure_Design/ (accessed Nov. 18, 2021).

[8] M. Akbar and M. A. F. Ridha, "SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall," p. 7.

[9] A. Marchand-Melsom and D. B. Nguyen Mai, "Automatic repair of OWASP Top 10 security vulnerabilities: A survey," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, Seoul Republic of Korea, Jun. 2020, pp. 23–30. doi: 10.1145/3387940.3392200.

[10] K. Odayan, "Artificial Intelligence controlling Cyber Security," p. 190.

[11] F. Cleary and M. Felici, Eds., *Cyber Security and Privacy*, vol. 530. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-25360-2.

[12] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. 2016. Accessed: Nov. 16, 2021. [Online]. Available: https://go.oreilly.com/university-of-alberta/library/view/-/9781439839430/?ar

[13] D. Cherry, "Securing SQL Server: Protecting Your Database from Attacker 3rd Edition," in *Securing SQL Server*, Elsevier, 2015, p. iii. doi: 10.1016/B978-0-12-801275-8.00016-6.

[14] N. Y. Xuan, J. Juremi, and N. H. M. Saad, "Securing e-commerce against SQL injection, cross site scripting and broken authentication," vol. 5, no. 2, p. 5, 2021.

[15] M. N. Kavitha, V. Vennila, G. Padmapriya, and A. R. Kannan, "Prevention Of Sql Injection Attack Using Unsupervised Machine Learning Approach," vol. 12, no. 03, p. 12, 2021.

[16] K. Ahmad and M. Karim, "A Method to Prevent SQL Injection Attack using an Improved Parameterized Stored Procedure," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120636.

[17] A. F. B. Dr, "INTERNATIONAL ADVISORY BOARD," vol. 10, p. 95, 2022.

[18] F. Deriba, "Development of a Compressive Framework Using Machine Learning Approaches for SQL Injection Attacks," *PRZEGLĄD ELEKTROTECHNICZNY*, vol. 1, no. 7, pp. 183–189, Jul. 2022, doi: 10.15199/48.2022.07.30.

[19] S. S. A. Krishnan, A. N. Sabu, P. P. Sajan, and A. L. Sreedeep, "SQL Injection Detection Using Machine Learning," vol. 11, no. 3, p. 11, 2021.

[20] A. Falor, M. Hirani, H. Vedant, P. Mehta, and D. Krishnan, "A Deep Learning Approach for Detection of SQL Injection Attacks Using Convolutional Neural Networks," in *Proceedings of Data Analytics and Management*, vol. 91, D. Gupta, Z. Polkowski, A. Khanna, S. Bhattacharyya, and O. Castillo, Eds. Singapore: Springer Singapore, 2022, pp. 293–304. doi: 10.1007/978-981-16-6285-0_24.

[21] P. Roy, R. Kumar, and P. Rani, "SQL Injection Attack Detection by Machine Learning Classifier," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, May 2022, pp. 394–400. doi: 10.1109/ICAAIC53929.2022.9792964.

[22] I. Hashem, M. Islam, S. M. Haque, Z. I. Jabed, and N. Sakib, "A Proposed Technique for Simultaneously Detecting DDoS and SQL Injection Attacks," *Int. J. Comput. Appl.*, vol. 183, no. 11, pp. 50–57, Jun. 2021, doi: 10.5120/ijca2021921428.

[23] I. Jemal, O. Cheikhrouhou, H. Hamam, and A. Mahfoudhi, "SQL Injection Attack Detection and Prevention Techniques Using Machine Learning," vol. 15, no. 6, p. 12, 2020.

[24] M. Hasan, Z. Balbahaith, and M. Tarique, "Detection of SQL Injection Attacks: A Machine Learning Approach," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, United Arab Emirates, Nov. 2019, pp. 1–6. doi: 10.1109/ICECTA48151.2019.8959617.

[25] B. Kranthikumar and R. L. Velusamy, "SQL injection detection using REGEX classifier," p. 10, 2020.