COMPARATIVE ANALYSIS OF SQL INJECTION ATTACK CLASIFICATION USING NAÏVE BAYES METHOD AND VECTOR MACHINE (SVM)

Pramono[1]*, Ridwan Dwi Irawan[2], Aprilisa Arum Sari[3]
Universitas Duta Bangsa Surakarta[1, 2, 3]
*Correspondence Email : pramono@udb.ac.id[1], ridwan_dwiirawan@udb.ac.id[2], aprilisa_arumsari@udb.ac.id[3]

ABSTRACT

SQL Injection is an attack that attempts to gain unauthorized access to a database by injecting code and exploiting SQL queries. SQL injection is an attack that is easy to execute but difficult to detect and classify because of the many types. The SQLI vulnerability is the result of incorrect validation of user input, enabling attackers to manipulate programmer queries by adding new SQL operators. Therefore, this study compares the use of the Naïve Bayes algorithm with the Support Vector Machine (SVM). The dataset that will be used in this study comes from a website called Kaggle. This study analyzes the comparison of methods resulting from the classification process based on the value of accuracy of confusion matrix, precision, recall. Naive Bayes, 95.594% accuracy quality while Support Vector Machine (SVM) 96.093% accuracy quality. The highest percentage of accuracy is obtained by the Support Vector Machine (SVM) while the Naïve Bayes accuracy score is slightly lower.

KEYWORDS

Sql Injection, Classification, Naïve Bayes, Support Vector Machine

INTRODUCTION

Information security has become a critical issue in today's increasingly interconnected digital world. Various forms of cyber threats have emerged, including SQL Injection attacks, which have evolved alongside the growing reliance on computer systems and the Internet(Jony & Hamim, 2024). SQL Injection is one of the most common and dangerous attacks on web-based applications, where attackers inject malicious SQL commands into queries executed by a database. These attacks can lead to unauthorized access to sensitive data, alteration, or even complete deletion of data, resulting in significant losses for organizations(Johny et al., 2021).

To counter these threats, the development of effective detection systems is essential. Various machine learning-based approaches have been employed to automatically detect and classify SQL Injection attacks(Alghawazi et al., 2022). Two techniques commonly used for this classification task are Naïve Bayes and Support Vector Machine (SVM). Since each method has its own strengths and weaknesses from a classification perspective, a comparative analysis of these two methods is important to gain deeper insights into their effectiveness(Syahputra et al., 2022).

Naïve Bayes is a classification method based on Bayes' Theorem, which assumes that the features used for prediction are independent of each other. Despite its simplicity, Naïve Bayes often yields very good results in various applications, such as cyberattack detection. However, the assumption of feature independence in Naïve Bayes is often unrealistic, particularly in the context of complex data like SQL Injection, and this can affect the model's accuracy(Rimal, 2019).

Support Vector Machine (SVM), on the other hand, is a more sophisticated classification method known for its superior performance in high-dimensional feature spaces. SVM works by finding an optimal hyperplane that can separate data from two classes with the maximum margin. The main strength of SVM lies in its ability to handle non-linear data using the kernel trick, enabling it to manage complexities that simpler methods like Naïve Bayes might overlook(Arum Sari, n.d.).

This article aims to provide a detailed comparative analysis of the Naïve Bayes and SVM techniques in classifying SQL Injection attacks. The primary goal of this analysis is to identify which method is more effective and efficient in detecting these attacks and to make recommendations for developing more robust security systems in the future. The findings of this research are expected to serve as a valuable reference for cybersecurity experts and researchers in the field of intrusion detection(Khraisat et al., 2019).

RESEARCH METHOD

The approach used in this research is quantitative, with an experimental research type(Barella et al., 2024). This study conducts a testing scenario by comparing the Naïve Bayes and Support Vector Machine (SVM) algorithms in detecting SQL Injection attacks from both attack and non-attack forms. The experimental scenario will then be evaluated to assess the accuracy in detecting and classifying attacks and non-attacks from each scenario to reveal the facts from the research results. The research conducted by the author is descriptive, where the data is explained in descriptive numbers and tables.

A. Data Collection Method

The research was conducted using a dataset consisting of plain SQL and SQLI. The dataset was then given three labels: label 0 indicates Non-SQL attacks, label 1 indicates SQLI Union attacks, label 2 indicates SQLI Tautology attacks, and the last label, label 3, indicates SQLI Blind attacks. Labeling the data helps the model learn the types of attacks and non-attacks. An example of the dataset can be seen in Table 1.

Table 1. Table of Example Dataset Usage

Sentence	Label
To believe thought, believe true private heart true men, — genius	0
1))) union all select null --	1
or 1 = 1 or "" =	2
1" Waitfor delay '0:0:5'	3

B. Data Analysis Method

The data analysis method used in this research is quantitative analysis using the Naïve Bayes and Support Vector Machine (SVM) algorithms. The obtained dataset will undergo preprocessing, a stage aimed at processing raw data into data ready for model development. The preprocessing process in this study includes three steps: Case Folding, Stopwords Removal, and Tokenization. The experiments in this research involve two main scenarios: using Naïve Bayes and Support Vector

Machine (SVM). These experiments were conducted using the Python programming language with the help of the sklearn library to build the Naïve Bayes and SVM architectures.

C. Modeling the Data Analysis Process

The Data Analysis Process is conducted to provide an overview of how this research will be carried out. This model illustrates the flow of how the dataset taken from the Kaggle website must go through several stages, including preprocessing, tokenization, feature extraction, and then the classification process using Naïve Bayes and Support Vector Machine (SVM) algorithms. The classification model is tested, resulting in a confusion matrix output. The accuracy, precision, and recall values of the two classification methods are then compared(Pusean et al., 2023). The test results from the dataset classification process produce a confusion matrix output. The design of the model can be seen in the accompanying diagram.

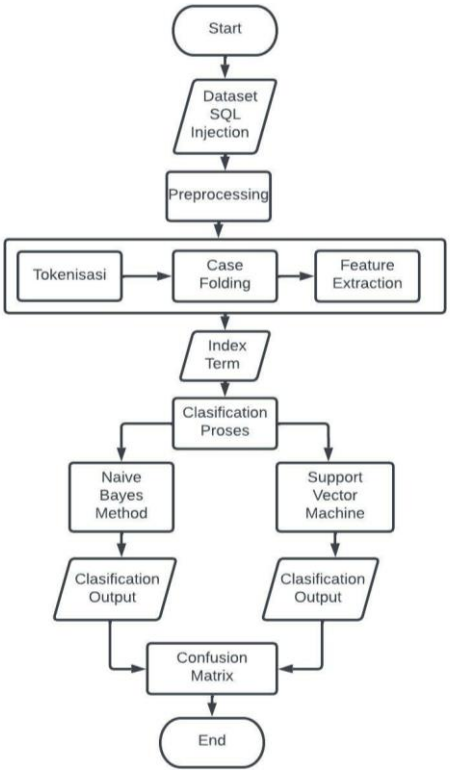

Figure 1. Data Analysis Process

RESULT AND DISCUSSION

A. Data Collection

1. Dataset

In this research, the dataset used is sourced from a website called Kaggle, which is a platform for analyzing and predicting datasets. The dataset is downloaded manually via the link https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset. The reason for selecting this dataset from the website is that it consists of a combination of URLs, special characters, textual data, and numerical data. The plain dataset contains ordinary text sentences classified

as Non-attacks or SQL, with 3060 rows. The second dataset, SQLI or the attack dataset, contains approximately 1127 rows. The dataset is stored on Google Drive because the classification process is carried out using Google Colab.

2. Preprocessing

At this stage, the process begins by loading data from Google Drive into Google Colab. The purpose of this stage is to process raw data into data that is ready for model development. The preprocessing process in this research involves three steps: Case Folding, Stopwords Removal, and Tokenization.

a) The first preprocessing stage is Case Folding. Case folding is the process of standardizing all words to be the same, or converting all uppercase letters to lowercase.
b) The next stage is Stopword Removal. Stopword removal is the process of eliminating words that are considered irrelevant or less important in a sentence. The words removed include: conjunctions, such as: and, or, as well as , prepositions, such as: in, to, at, unwanted words
c) The final stage is Tokenizing, also known as parsing. Tokenizing is the process of splitting a document into individual words, called tokens. Spaces are used to separate these words. Unnecessary words are removed through a filtering process after tokenizing.

3. Feature Extraction

After the Preprocessing stage, the next step is Feature Extraction. Feature extraction is useful for recognizing characteristics in the query within the dataset and using them as features. In machine learning, features are used to achieve good performance, making feature extraction an important step. In this research, feature extraction is performed using TF-IDF.

After completing the Tokenizing stage in preprocessing, weighting is applied to the tokens in the document using the TF-IDF method(Siino et al., 2024). TF-IDF consists of Term Frequency (TF) and Inverse Document Frequency (IDF). A high term value in a document will influence the resulting value. Inverse Document Frequency (IDF) is the calculation of a term's occurrence distributed across related documents. To obtain a large IDF value, the number of documents containing the term must be small.

After all the data has been standardized in size, the data and data labels are stored in variables in the form of arrays. In other words, the final result of this process is a matrix in the form of a multidimensional array, so it can be used as input for the Naïve Bayes and Support Vector Machine algorithms, stored in the variables `x` and `y`.

B. Data Grouping

After performing preprocessing and feature extraction on the dataset, the next step is data grouping by splitting the dataset into two parts: training data and test data, with a ratio of 50%:30%:10%. The `train_test_split` function provided by Sklearn, a Python library, is used to randomly divide the data.

With the process already completed, the data was divided into the following ratios: At a 50% ratio: the training data consists of 2003 entries, and the test data consists of 2004 entries. At a 30% ratio: the training data consists of 2804 entries, and the test data consists of 1203 entries. At a 10% ratio: the training data consists of 3606 entries, and the test data consists of 401 entries.

C. Data Analysis

The dataset used in this research consists of plain queries and SQL Injection queries, totaling 4187 data points, which include 3060 plain queries and 1127 SQL Injection queries. The data is then processed through a preprocessing stage to prepare it for classification. This is followed by a feature extraction process, which aims to assign weights to specific features or characteristics in the document. Afterward, the data and data labels are stored in variables in the form of multidimensional arrays to serve as input for the Naïve Bayes and Support Vector Machine algorithms. Once the classification results for each data point are obtained, the accuracy is calculated using test data of varying sizes, with the results evaluated using a confusion matrix.

1. Analysis of Results Based on the Naïve Bayes Algorithm.

Table 2. Scenario Analysis of Test Results Based on the Naïve Bayes Algorithm

Scenario	Number of Training Data	Number of Test Data	Accuracy	Precision	Recall
S1	2003	2004	0,9531	0,9695	0,9088
S2	2804	1203	0,9667	0,9786	0,9349
S3	3606	401	0,9651	0,9776	0,9320

a) The first scenario.

The following are the results of testing 2004 test data against 2003 training data. As shown in the image, the accuracy rate for testing with 2004 test data is 96.1077%.

b) The Second Scenario

The following are the results of testing 1203 test data against 2804 training data. As shown in the image, the accuracy rate for testing with 1203 test data is 95.2618%.

c) The Third Scenario

The following are the results of testing 401 test data against 3606 training data. As shown in the image, the accuracy rate for testing with 401 test data is 96.2593%.

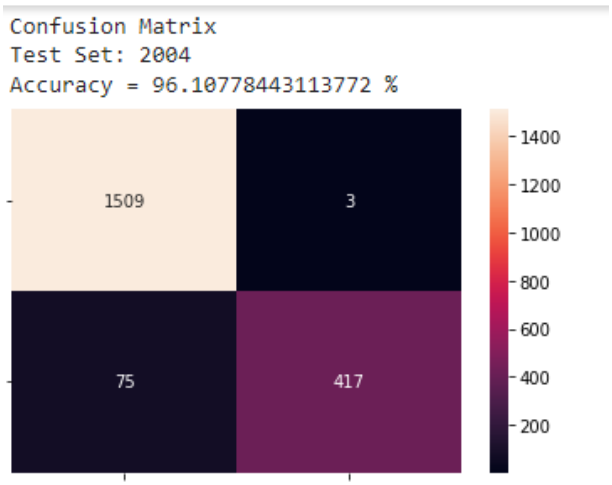

Figure 2. First Scenario

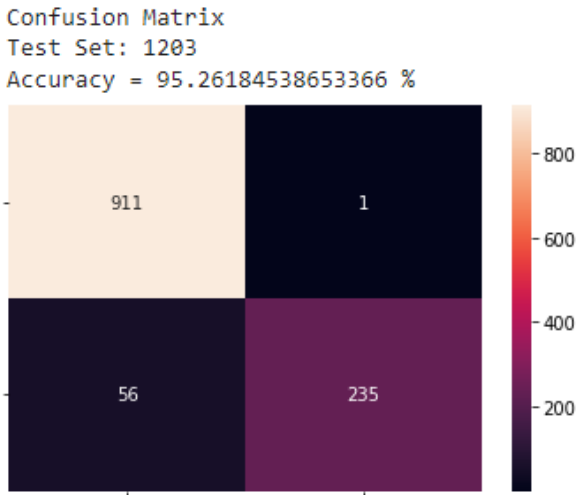

Figure 3. Second Scenario

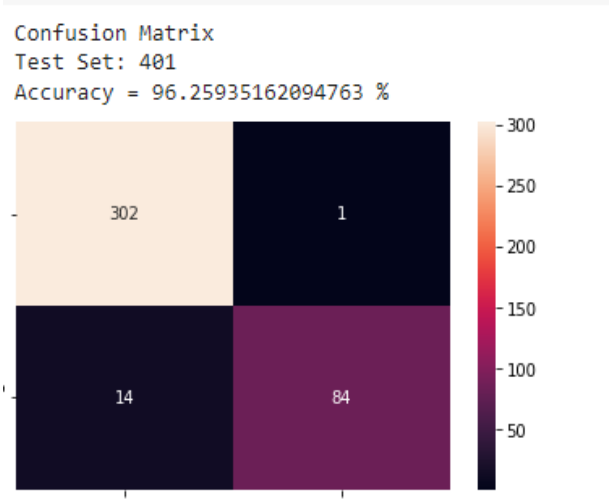

Figure 4. Third Scenario

2. Analysis of Results Based on the Support Vector Machine Algorithm

Table 3. Scenario Analysis of Test Results Based on the Support Vector Machine Algorithm

Scenario	Number of Training Data	Number of Test Data	Accuracy	Precision	Recall
S1	2003	2004	0,9646	0,9757	0,9302
S2	2804	1203	0,9485	0,9657	0,9050
S3	3606	401	0,9426	0,9627	0,9009

d) The first scenario.

The following are the results of testing 2004 test data against 2003 training data. As shown in the image, the accuracy rate for testing with 2004 test data is 96.1077%.

e) The Second Scenario

The following are the results of testing 1203 test data against 2804 training data. As shown in the image, the accuracy rate for testing with 1203 test data is 96.8412%.

f) The Third Scenario

The following are the results of testing 401 test data against 3606 training data. As shown in the image, the accuracy rate for testing with 401 test data is 95.7605%.

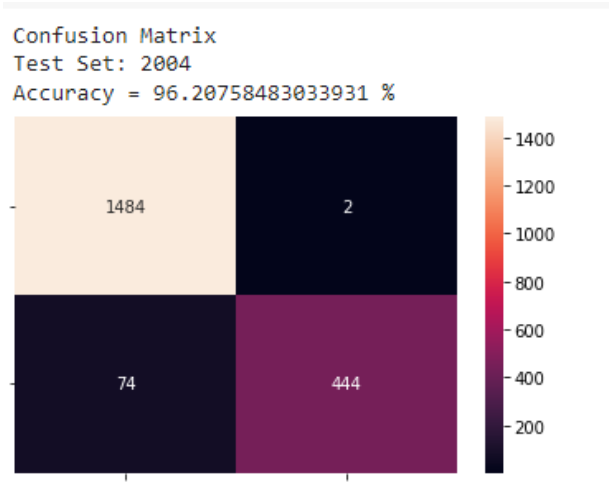

Figure 5. First Scenario

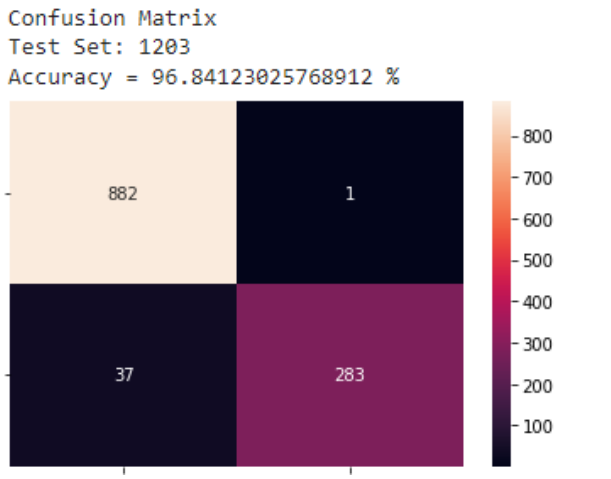

Figure 5. Second Scenario

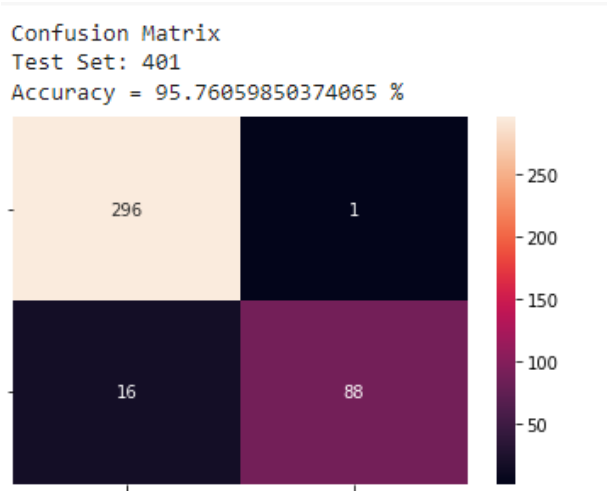

Figure 7. Third Scenari

D. Analysis of Research Results

After processing the data and training the model for each experimental scenario using the training data, several classification models were obtained, corresponding to the number of scenarios used. These models were then tested using test data to determine the accuracy and performance of each model. The results of the training and testing of the classification models can be seen in the following table.

Table 4. Accuracy Table for Naïve Bayes and SVM Algorithms

Skenario	Number of Training Data	Number of Test Data	Accuracy		Precision		Recall	
			SVM	Naïve Bayes	SVM	Naïve Bayes	SVM	Naïve Bayes
S1	2003	2004	0,9646	0,9531	0,9757	0,9695	0,9302	0,9088
S2	2804	1203	0,9485	0,9667	0,9657	0,9786	0,9050	0,9349
S3	3606	401	0,9426	0,9651	0,9627	0,9776	0,9009	0,9320

From the table of scenarios above, it is known that the Precision value of the Support Vector Machine is higher with a value of 0.9757 in Scenario 1. However, in Scenarios 2 and 3, Naïve Bayes has a higher Precision value with 0.9786 in S2 and 0.9776 in S3. Meanwhile, Recall achieved by Naïve Bayes is higher in Scenarios 2 and 3 with values of 0.9349 in S2 and 0.9320 in S3, whereas Support Vector Machine has a relatively high value in Scenario 1 with 0.9302. In terms of Accuracy, Support Vector Machine achieves the highest value in Scenario 1 with an accuracy of 0.9646, while Naïve Bayes has the highest Accuracy in Scenario 2 with a value of 0.9667.

CONCLUSION

Based on the analysis of the results from the experiments conducted on both architectures with 3 experimental scenarios, it can be concluded that:

1. The dataset processing, downloaded from the Kaggle website, was carried out in several stages, including preprocessing through Case Folding, Feature Extraction, and then storing the data and labels in variables in the form of arrays until the data was ready for classification.

2. The processed dataset was then grouped by splitting it into two parts: training data and test data, with ratios of 50%:30%:10%. Using Sklearn, a Python library, the results were: for the 50% ratio, the training data amounted to 2003 and the test data to 2004; for the 30% ratio, the training data amounted to 2804 and the test data to 1203; and for the 10% ratio, the training data amounted to 3606 and the test data to 401.
3. The performance results from the confusion matrix for the Naïve Bayes classifier were: Scenario 1 with a 50% ratio and 2004 test data had an accuracy of 96.1077%; Scenario 2 with a 30% ratio and 1203 test data had an accuracy of 95.2618%; and Scenario 3 with a 10% ratio and 401 test data had an accuracy of 96.2593%.
4. The performance results from the confusion matrix for the Support Vector Machine classifier were: Scenario 1 with a 50% ratio and 2004 test data had an accuracy of 96.2075%; Scenario 2 with a 30% ratio and 1203 test data had an accuracy of 96.8412%; and Scenario 3 with a 10% ratio and 401 test data had an accuracy of 95.7605%.
5. Naïve Bayes is the algorithm with the highest accuracy compared to the Support Vector Machine algorithm. However, there is a significant difference in Scenario 2, where the Support Vector Machine outperforms the Naïve Bayes algorithm.

Here are some recommendations for further developing this research:

1. This research is limited to the dataset downloaded from Kaggle. Future research should consider using other datasets to obtain different results.
2. To improve the accuracy of a method, techniques such as bagging and boosting can be employed. This study is limited to a comparison between Naïve Bayes and Support Vector Machine (SVM) methods. Future research could incorporate bagging and boosting techniques to enhance accuracy.
3. It is advisable to add variation to experimental scenarios, such as using different ratios for splitting training and test data.
4. The training process should preferably be conducted using Python libraries run locally on a computer. Using Google Colab may result in less accurate computation times due to varying internet speeds during each experiment.

REFERENCES

Alghawazi, M., Alghazzawi, D., & Alarifi, S. (2022). Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. Journal of Cybersecurity and Privacy, 2(4), 764–777. https://doi.org/10.3390/jcp2040039

Arum Sari, A. (n.d.). Prediksi Serangan Sql Injection Pada Jaringan Komputer Menggunakan Metode Support Vector Machine (SVM). https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset

Barella, Y., Fergina, A., Mustami, M. K., Rahman, U., & Alajaili, H. M. A. (2024). Quantitative Methods in Scientific Research. Jurnal Pendidikan Sosiologi Dan Humaniora, 15(1), 281. https://doi.org/10.26418/j-psh.v15i1.71528

Johny, J. H. B., Nordin, W. A. F. B., Lahapi, N. M. B., & Leau, Y. B. (2021). SQL Injection Prevention in Web Application: A Review. Communications in Computer and Information Science, 1487 CCIS, 568–585. https://doi.org/10.1007/978-981-16-8059-5_35

Jony, A. I., & Hamim, S. A. (2024). Navigating the Cyber Threat Landscape: A Comprehensive Analysis of Attacks and Security in the Digital Age. Journal of

Information Technology and Cyber Security, 1(2), 53–67. https://doi.org/10.30996/jitcs.9715

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1). https://doi.org/10.1186/s42400-019-0038-7

Pusean, N. V., Charibaldi, N., & Santosa, B. (2023). Comparison of Scenario Pre-processing Performance on Support Vector Machine and Naïve Bayes Algorithms for Sentiment Analysis. Inform : Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi, 8(1), 57–63. https://doi.org/10.25139/inform.v8i1.5667

Rimal, Y. (2019). INTERNATIONAL JOURNAL ON ORANGE TECHNOLOGIES (IJOT) e-Naïve Bayes Machine Learning Classification with R Programming: A case study of binary data sets. www.researchparks.org

Siino, M., Tinnirello, I., & La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. Information Systems, 121. https://doi.org/10.1016/j.is.2023.102342

Syahputra, R., Yanris, G. J., & Irmayani, D. (2022). SVM and Naïve Bayes Algorithm Comparison for User Sentiment Analysis on Twitter. Sinkron, 7(2), 671–678. https://doi.org/10.33395/sinkron.v7i2.11430

https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset