Applied Information Technology And Computer Science Vol. 3 No. 2 (2022) 019-031
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

A Comparative Study of SQL Injection Detection Using Machine Learning Approach

Fabian Dass Michael Dass[1], **Cik Feresa Mohd Foozy**[1]*

[1]Faculty Computer Science & Information Technology,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Malaysia

DOI: https://doi.org/10.30880/aitcs.2022.03.02.002
Received 21 July 2022; Accepted 30 September 2022; Available online 30 November 2022

Abstract: Injection vulnerabilities are still the most common and deadly attacks against online applications. Therefore, a SQL Injection detection framework with suitable approaches has been proposed. In this paper, the Random Forest and Support Vector Machine algorithms and detection of SQL Injection are analyzed. The experiments were carried out and tested on HTTPParamsDataset. In this research, there are six (6) phases implemented in the research such as Raw Data, Data Preprocessing, Feature Extraction, Features Selection, Classification, and Result. The experiments evaluated in terms of Accuracy, True Positive, True Negative and Precision in identifying the best performances classifiers. Random Forest classifiers have 96.02 percent average accuracy compared to Support Vector Machine classifiers with 91.89 percent accuracy without Information Gain for SQL Injection detection. Random Forest with Information Gain got 96.02 percent accuracy while Support Vector Machine got 91.89 and 90.25 percent.

Keywords: Random Forest, SQL Injection, Support Vector Machine

1. Introduction

Humans are increasingly reliant on data in the twenty-first century, including usernames, passwords, and user information. For any organization or individual, data is a critical asset and identifier. Unfortunately, as a result of hacking organizations' databases, there has been an increase in cyber-attacks such as Structured Query Language (SQL) Injection. According to the Open Web Application Security Project's Top Ten Attacks list, injection vulnerabilities remain the most prevalent vulnerability in web applications [1]. Typically, black hat hackers are responsible for these types of cyber-attacks, which they perpetrate against vulnerable web applications. Additionally, there are numerous tools available for exploiting the vulnerabilities of web applications. hackers with increased opportunities to hack databases via SQL Injection attacks. It is a type of website manipulation attack that aims to expose sensitive data by injecting malicious SQL queries into the database, thereby compromising the Confidentiality, Integrity, and Availability (CIA) triad of security. As mentioned previously, SQL Injection attacks are getting out of hand. Moreover, there are some significant features were not being used to detect SQL Injection while some of the detection systems are giving lower rates due to the unsuitable techniques and classification [2]. Therefore, in this work, SQL Injection detection

*Corresponding author: feresa@uthm.edu.my framework using Machine Learning algorithms and techniques is initiated to detect attacks and analyze the results from the framework. Recent research has revealed that machine language algorithms can assist in detecting and preventing various types of cyber threats. The computational resources and time required by such complicated algorithms continue to be a major source of concern for the security community, despite the fact that their effectiveness in detecting security threats is unquestionable. Multiple machine learning algorithms can be used to detect SQL injection attacks. This project begins by introducing SQL injection attacks and then proposes and develops a detection framework using machine learning algorithms. Therefore, the objectives of this project are:

1. to design a detection for SQL Injection attacks using Machine Learning techniques.
2. to develop a detection for SQL Injection attacks using Support Vector Machine and Random Forest algorithm.
3. to evaluate the SQL Injection attack detection in terms of Accuracy, Precision, True-Positive and True-Negative.

The rest of the paper is organized as follows: Section 2 discussed about related work of SQL Injection detection. Then, Section 3 described about detection methodology which has been used in this paper. Section 4 explained about the result from the experiment and finally, Section 5 discussed about the conclusion and future work.

2. Related Work

This section discussed SQL injection, Machine Learning and its categories, and the existing research on SQL injection detection techniques using Machine Learning.

2.1 SQL Injection

SQL Injection Attack (SQLIA) is one of the most significant cyber threats to web database systems. Attackers take advantage of SQLIA to gain unauthorized access to and modify data [3]. User input data can be used to inject a SQL query into an application [1]. When a SQL injection vulnerability is successfully exploited, it is possible to retrieve sensitive data, update database data, perform database administration, recover data from the Database Management Systems (DBMS) file system, and run commands. SQL injection attacks alter the way prepared SQL commands are executed by injecting SQL commands into the data planes' input. UNION Based SQL Injection, Error Based SQL Injection, and Blind SQL Injection are the common types of SQL Injection.

UNION Based SQL injection is a type of injection that allows hackers or attackers to hack or retrieve information from a database. The UNION operator has been used by the attacks to insert a harmful query into the initial query in Union Based SQL Injection. By using the harmful query to link the results of the initial query, the attacker will be able to access the values of columns in subsequent tables [4]. Erroneous data is entered into a SQL statement which causes the database to crash in Error Based SQL Injection. These are commonly achieved by pressuring an error-prone database activity. The user can then search for database issues and use them to learn how to navigate the database with SQL queries [4]. Lastly, Blind SQL injection is a SQLI attack in which the attacker queries the database true or false questions to see which solution best supports the application's response. The attack is frequently used on the internet and is designed to highlight generic error alerts; however, it does not alleviate code that is exposed to SQL injection [4].

2.2 Machine Learning

Machine Learning is a type of artificial intelligence that applies statistical methods to identify patterns and relationships in data. Systems may be able to learn and grow without precise instructions. The aim of machine learning is to create computer programs that are able to learn and access information on their own. It is one of the most rapidly evolving technologies, with plenty of functions and datasets.

There are still demands for machine learning. Automatically, it discovers the most important trends in data. As a result, machine learning technologies focus on programming's ability to learn and adapt [5]. While it is frequently faster and more accurate in detecting lucrative opportunities or harmful dangers, adequate training may take more time and resources. Unsupervised and supervised machine learning are the two types of machine learning [5]. Figure 1 shows the example and types of Machine Learning techniques.

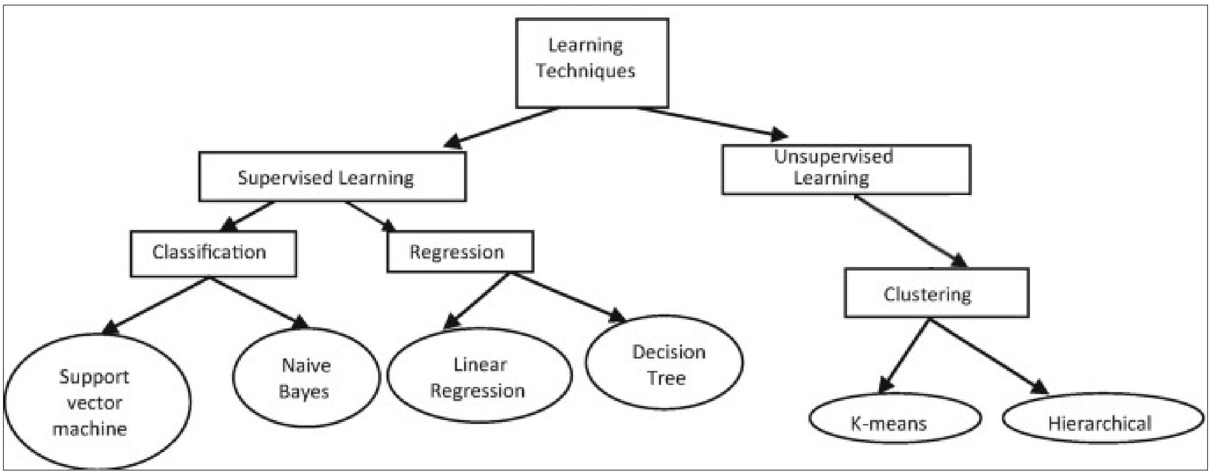


Figure 1: Machine Learning Techniques [5]

Unsupervised machine learning approaches are used when training data isn't labeled or classified. Computers can derive a function from unlabeled data to determine a dataset's hidden patterns using unsupervised learning. It analyses data and draws conclusions from datasets to characterize the hidden structures of unlabeled data instead of producing the correct result. Learner-independent data exploration and categorization are two of the main goals of unsupervised learning. Clustering and association are two types of unsupervised learning algorithms. In order to categorize items, clustering is used to group those that share the most similarities together, and those that share the fewest or no similarities. Based on whether or not they share commonalities, cluster analysis identifies and categorizes data objects. It is possible to discover correlations between variables in a large database using an association rule, an unsupervised learning strategy. In the collection, it's used to group together items that appear in the same order repeatedly. Effectiveness is increased by adhering to the rule of association [6].

Nevertheless, the term "supervised machine learning" refers to algorithms that require external assistance. The input dataset is separated from the training and testing datasets. To predict output values, the learning algorithm builds an inferred function from the research of a given training dataset. Any new input can be provided with a target after proper training. Additionally, the algorithm's results can be compared to the intended results, allowing the model to be improved. Subsequently, algorithms that learn behaviors and patterns from the training dataset and use them to predict or classify data are all part of the same cycle. Regression and classification problems are two of the most common types of supervised learning problems. For categorical data, classification algorithms are used, whereas regression is used for continuous data. Regression procedures are used if there is a correlation between the input variable and the output variable [6]. In this research, supervised learning will be used due to usage of classification algorithms like the Random Forest and Support Vector Machine.

2.3 Comparison on SQL Injection Detection Techniques using Machine Learning based on the existing research and the proposed research

Table 1 shows the comparative analysis existing work on SQL Injection Detection using Machine Learning. This work differs from other research in such a way that proposed the SQL Injection Detection approach through feature selection, feature extraction and two classifications techniques which are the Support Vector Machine (SVM) and Random Forest (RF) and tested on the dataset which is the HttpParamsDataset using WEKA tool. The sample data of SQL Injection attacks from the dataset are 10852 items labeled as "anom" while the benign values are 19304 items labeled as "norm". Furthermore, in this research, feature selection will be used which helps to decrease the number of attributes in the dataset that enables machine learning (SVM and Random Forest) algorithms to train faster. Information Gain Algorithm will be used for the feature selection. For the feature selection, the initial modeling will be using the information gain algorithm by using all features. After the modeling, the information gain algorithm ranked the feature according to the importance, will keep the top feature and remove the unrelated features. Once it is trained with machine learning algorithms, the result will be evaluated in terms of Accuracy, Precision, TP and TN. This work focuses on detecting SQL Injection similar to [7], [8], [2] and [9]. The dataset used in this research is from [10].

Table 1: Comparison Table

Work	Dataset	Features Techniques	Classification	Result
Abdulmalik [7]	Benign and Malicious SQL queries dataset	Feature Extraction (Extracting semantic features from dynamic and static analysis)	Random Forest (RF), Artificial Neural Network (ANN), Support Vector Machine (SVM) and Logistic Regression (LR)	Not mentioned

Chen et. al [8]	1000 samples of SQL injected in GitHub and exploit-db website, and 1000 samples of normal HTTP request as negative examples	Feature Extraction (The sample has been vectorized and one-hot code method, word2vec features have been used to express text features as vectors)	Support Vector Machine	Statistical Characteristics
				TPR = 0.920
				FPR = 0.058
				Recall = 0.923
				ROC curve area = 0.933
				Word2vec
				TPR = 0.954
				FPR = 0.041
				Recall = 0.946
				ROC curve area = 0.982

Azman et. al [2]	Damn Vulnerable Web Application (DVWA) and bWapp (Rotates the data sample until the fifth test set as the final data sample.)	Feature Extraction (Access log has been extracted and separated into a test set and training set)	String Matching and Boyer's Moore	Accuracy Result: First Test Set 93% Other 4 test set 100 %

Table 1: (cont.)

Work	Dataset	Features Techniques	Classification	Result

Krishnan et. al [9]	GitHub and Kaggle (The training and testing sets are randomly taken from the
	dataset with a conventional ratio of 80-20)
		Feature Extraction (Natural Language Processing and Word Level TF-IDF Vectors have been used throughout this extraction process)	Naïve Bayes (NB), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Logistic Regression (LR), Passive Aggressive (PA)	Accuracy Result:
				NB= 95 %
				LR= 92 %
				CNN = 97 %
				SVM = 79%
				PA = 79 %

3. Methodology

This section discusses the SQL Injection detection model using machine learning approaches. There are six phases involved in this SQL Injection detection model; Raw Data, pre-processing, features extraction, feature selection, classification using 10-fold cross validation and the evaluation of the classification result as shown in Figure 2. Random Forest and Support Vector Machine as classifiers. WEKA tool has been used to analyze the performance in term of Accuracy rate, True Positive, True Negative and Precision.

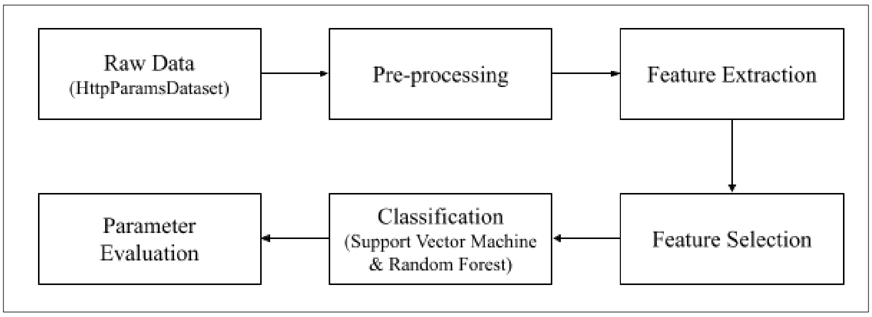

Raw Data (HttpParamsDataset)
Pre-processing
Feature Extraction
Feature Selection
Classification (Support Vector Machine & Random Forest)
Parameter Evaluation

Figure 2: Proposed SQL Injection Detection Methodology

3.1 Raw Data

The dataset that will be used in this research are HTTPParamsDataset[10]. The downloaded data from the GitHub [10] is raw because the data is not in the normalized form as well as the data contains 4 types of attacks such as SQL Injection attacks, Cross-Site Scripting, Command Injection, and Path Traversal attacks. This research is only focused on SQL injection attacks. Therefore, 13172 benign data labeled as “norm” and 8758 anomaly data labeled as “sqli” has been chosen from [11].

3.2 Pre-processing

Data preprocessing is a crucial phase in Machine Learning that enhances quality of the data and helps the extraction of significant insights from the data. Pre-processing of data is used to convert the raw data into clean and organized form, and to remove the irrelevant data which is not required by our research. Simply, data preprocessing is a data mining method that converts raw data into understandable format. In data preprocessing, this project will remove duplicate data and the 3 types of attack such as, Cross-Site Scripting, Command Injection, and Path Traversal attacks from the dataset [10] which were irrelevant to the research based on the data reduction process from pre-processing steps.

3.3 Feature Extraction

Feature extraction is process that identifies important and new features of the data. It yields better result and speed up in the training process as well Originally, the dataset provided a feature and a class attribute which are "length", and "attack_type" as shown in Table 2 based on the raw data. In this process, there are 5 features have been extracted out of the raw dataset based on the keywords that triggers or used in SQL Injection as shown in Table 3 including the provided feature as well.

Table 2: Features provided in the HTTPParamsDataset

Feature / Class Attribute	Descriptions
length	Length of the queries
attack_type (class attribute)	It is a class attribute to records in WEKA were attributed during the classification and indicate whether the selected queries are injected with SQL or not based on the raw data

Table 3: Features extracted from the HTTPParamsDataset

Features	Descriptions
count	Indicate whether the selected queries used "count" word to identify either benign or sqli
null	Indicate whether the selected queries used "null" word to identify either benign or sqli
select	Indicate whether the selected queries used "select" word to identify either benign or sqli
union	Indicate whether the selected queries used "union" word to identify either benign or sqli
where	Indicate whether the selected queries used "where" word to identify either benign or sqli

3.4 Feature Selection

Feature selection is a technique for removing superfluous and redundant information from a dataset. This approach to feature selection incorporates the Information Gain (IG) algorithm. Information Gain (IG) is a widely used entropy-based feature evaluation method in machine learning [11]. Eq. 1 is the equation for this information gain algorithm.

$$E = -\sum_{i}^{C} P_i \log_2 P_i \qquad \text{Eq. 1}$$

Eq. 1 is used to determine the entropy of a divide by subtracting the weighted entropies of each branch from the total entropy. Thus, when using these metrics to train, the optimal split is determined by maximizing Information Gain. The element E stands for entropy, while the element P_i denotes the probability of randomly selecting an element from a class [12]. A top 3, 4 and 5 features out of 6 features will be ranked for the experiment.

3.5 Classification

In this research, two Machine Learning algorithms will be used which are the Random Forest and Support Vector Machine. Random Forest algorithm is one of the supervised classification techniques. Random Forest was created by the merging of Decision Trees [13]. In this investigation, the model's prediction is based on the class with the highest number of random forest scores. Since the data is labeled therefore the random forest algorithm is suitable. It is a widely used predictive modelling and machine learning approach. This approach is applicable to classification and regression applications. The Random Forest constructs many decision trees and combines them to get a more accurate and consistent forecast. Random Forest has a significant benefit in that it can be used for both classification and regression issues. Additionally, it is easier to train and provides higher accuracy. Additionally, it manages unbalanced data. It includes built-in error balancing mechanisms and the capability to lower the total error rate. Eq. 2 is the formula of Random Forest classifier. The f_i is the frequency of label at i a node and C is the number of unique labels.

$$E = -\sum_{i=1}^{C} f_i(1 - f_i) \qquad \text{Eq. 2}$$

Support Vector Machine algorithm is a supervised and linear Machine Learning technique that is most frequently used to tackle classification problems. SVR stands for Support Vector Regression and is a subset of Support Vector Machine that uses the same ideas to tackle regression issues. The kernel approach, commonly known as the kernel Support Vector Machine, is another feature of Support Vector Machine that helps the research to deal with non-linearity [8]. The kernel function will be helpful for this research as the function can solve complex problems. Since the data is labeled therefore the random algorithm is suitable for this research. The aim of the algorithm is to discover a hyperplane in N-dimensional space that classifies the data points in a distinct manner [14]. This method aids in the accuracy of this research. Finally, it may help the research in situations where the dataset can be converted to high-dimensional data using the kernel function to eliminate the requirement for assumptions [14].

10-Fold Cross Validation has been employed in this research to train and test the data. This approach divides the dataset into ten pieces, referred to as "folds," in order to hold out each portion in turn and average the findings. As a result, each data point in the dataset is tested once and trained nine times. This statistical approach is used to measure machine learning model skill. This validation method was chosen to avoid data bias caused by issues and problems found through input features. The purposes of applying cross-validation in this research are to solve the issue of overfitting and make predictions broader. Furthermore, it aids in determining the detection quality, assuring optimal performance. This whole process will be run using WEKA.

3.6 Parameter Evaluation

In order to measure the effectiveness of the proposed SQL Injection Detection model. These performance metrics have been applied as shown in Table 4.

Table 4: Performance Evaluation and its formula

Performance Evaluation	Formula	Description
True Positive (TP)	$\frac{TP}{TP+FN}$	An outcome where the dataset properly predicts anomaly class (sqli)
True Negative (TN)	$\frac{TN}{TN+FP}$; FP = False Positive	An outcome where the dataset predicts benign class (norm)
Accuracy (A)	$\frac{TP+TN}{TP+FP+TN+FN}$; FN = False Negative	A measurement for evaluating classification models and fraction of correct predictions from the dataset
Precision (P)	$\frac{TP}{TP+FP} \times 100\%$	A measurement for evaluating the number of accurate positive class (sqli) predictions

3.7 Hardware and Software Requirement

Table 5 shows the hardware requirement to conduct the experiment. A HP Gaming Pavilion 15-dk0010tx laptop has been used to conduct this research. RapidMiner and WEKA will be utilized in this research. This software is a data mining program that employs a variety of machine learning methods. These algorithms may be applied on data directly or invoked from Java code. RapidMiner and WEKA are set of tools for Regression, Association, Clustering, Data Preparation, Visualization, and Classification In order to measure the effectiveness of the proposed SQL Injection Detection model. These performance metrics have been applied as shown in Table 5.

Table 5: Hardware Requirements

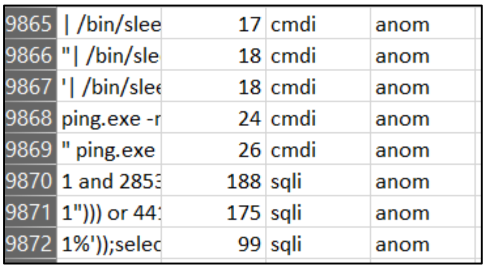
Hardware	Description			
HP Gaming Pavilion 15-dk0010tx	Processor	System Type	Installed RAM	Windows Edition
	Intel® Core™ i5-9300H CPU @ 2.40GHz	64-bit operating system, x64-based processor	32.0 GB	Windows 10 Home

4. Results and Discussion

This section presents the experimental setup. In this study, the classification was performed using WEKA. This section explains about the result that has been obtained from the experiment using HTTPParamsDataset [10].

4.1 Preprocessing

The dataset is raw since it is not normalized and because it contains four different types of attacks, including SQL Injection attacks as shown in Figure 3. However, this project is focused on SQL Injection attacks. The preprocessing was performed in RapidMiner as shown in Figure 4 and Waikato Environment for Knowledge Analysis (WEKA) software in Figure 6. During the preprocessing process, the dataset has been trimmed in various circumstances like removing any duplicates and filtering the dataset by keeping the benign and SQL query. Furthermore, it also converts numerical to binary and nominal filters as well to give a better accuracy. StringToWordVector filter has been applied to the dataset, so it converts string attributes into a set of numeric attributes representing word occurrence information from the text contained in the strings. The preprocessed dataset as shown in Figure 5 consists of 13172 benign queries which is known as the "norm" and 8758 anomaly SQL queries which is known as the "sqli". The dataset is then being saved into the CSV file to input in WEKA as shown in Figure 6 for experimental purposes.



9865	| /bin/slee	17	cmdi	anom
9866	"| /bin/sle	18	cmdi	anom
9867	'| /bin/slee	18	cmdi	anom
9868	ping.exe -r	24	cmdi	anom
9869	" ping.exe	26	cmdi	anom
9870	1 and 2853	188	sqli	anom
9871	1"))) or 44	175	sqli	anom
9872	1%'));selec	99	sqli	anom

Figure 3: Sample of the dataset before pre-processing process

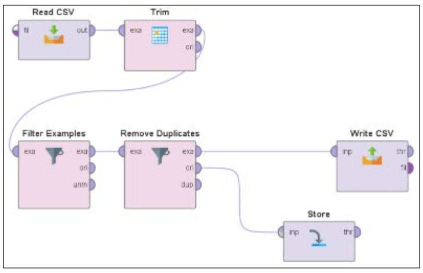
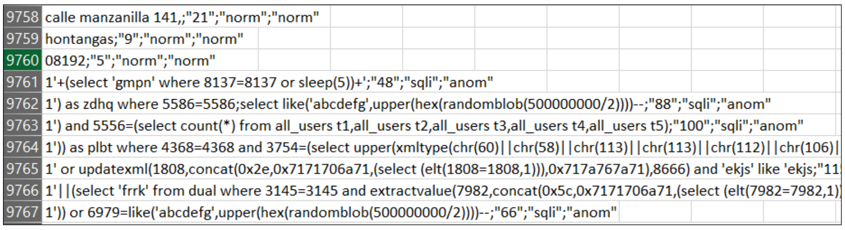
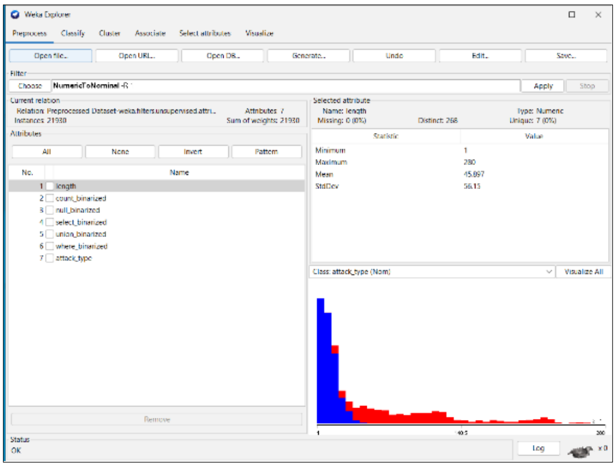


Figure 4: Process of Pre-processing in RapidMiner



9758	calle manzanilla 141,;"21";"norm";"norm"
9759	hontangas;"9";"norm";"norm"
9760	08192;"5";"norm";"norm"
9761	1'+(select 'gmpn' where 8137=8137 or sleep(5))+';"48";"sqli";"anom"
9762	1') as zdhq where 5586=5586;select like('abcdefg',upper(hex(randomblob(500000000/2))))--;"88";"sqli";"anom"
9763	1') and 5556=(select count(*) from all_users t1,all_users t2,all_users t3,all_users t4,all_users t5);"100";"sqli";"anom"
9764	1')) as plbt where 4368=4368 and 3754=(select upper(xmltype(chr(60)||chr(58)||chr(113)||chr(113)||chr(112)||chr(106)|
9765	1' or updatexml(1808,concat(0x2e,0x7171706a71,(select (elt(1808=1808,1))),0x717a767a71),8666) and 'ekjs' like 'ekjs;"11
9766	1'||(select 'frrk' from dual where 3145=3145 and extractvalue(7982,concat(0x5c,0x7171706a71,(select (elt(7982=7982,1))
9767	1')) or 6979=like('abcdefg',upper(hex(randomblob(500000000/2))))--;"66";"sqli";"anom"

Figure 5: Sample of the dataset after pre-processing process



Filter: NumericToNominal -R

Current relation
Relation: Preprocessed Dataset-weka.filters.unsupervised.attri... Attributes: 7
Instances: 21930 Sum of weights: 21930

Attributes

No.	Name
1	length
2	count_binarized
3	null_binarized
4	select_binarized
5	union_binarized
6	where_binarized
7	attack_type

Selected attribute
Name: length Type: Numeric
Missing: 0 (0%) Distinct: 268 Unique: 7 (0%)

Statistic	Value
Minimum	1
Maximum	280
Mean	45.097
StdDev	56.13

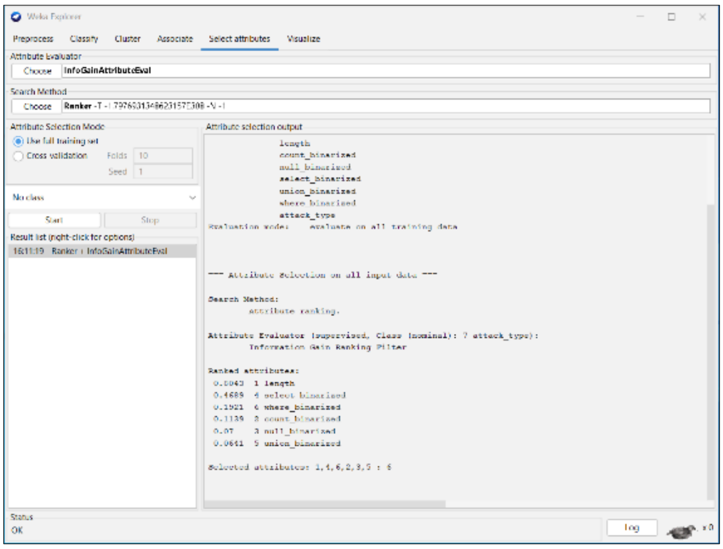
Class: attack_type (Nom)

Status
OK

Figure 6: Numerical to Binary and Numerical to Nominal filter have been applied in the dataset using WEKA

4.2 Feature Extraction and Selection

The dataset provides a feature which is length, and one class attribute (attack_type). However, five features have been extracted from the raw data which are the queries of the raw dataset based on the keywords that triggers or used in SQL Injection such as 1759 "count", 1110 "null", 5842 "select", 1029 "union", and 2828 "where" features for feature extraction process. These words are the most frequently used words that can be seen in SQL Injection queries. These new features and the originated features are saved in new comma separated values (csv) files for feature selection, classification, and result purpose. To select the best features for this study, a filter method has been used to get the subset of the features. In the feature selection phase, the filter method that has been used here is Information Gain (IG) algorithm.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode
● Use full training set
○ Cross validation Folds 10 Seed 1

No class

Result list (right-click for options)
16:11:19 - Ranker + InfoGainAttributeEval

Attribute selection output

```
              length
              count_binarized
              null_binarized
              select_binarized
              union_binarized
              where_binarized
              attack_type
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 attack_type):
        Information Gain Ranking Filter

Ranked attributes:
 0.5043  1 length
 0.4689  4 select_binarized
 0.1921  6 where_binarized
 0.1139  2 count_binarized
 0.07    3 null_binarized
 0.0641  5 union_binarized

Selected attributes: 1,4,6,2,3,5 : 6
```

Status
OK

Figure 7: Sample of Information Gain Ranked the Features

Table 6 shows the value of Information Gain from Feature Selection phase based on Figure 7. All the features have been listed and ranked in this phase. The features will be ranked from the most important features to the less important based on the score. The value has been arranged in descending order. The lowest value is Union with value 0.0641.

Table 6: IG value

Features	IG value
Length	0.807
Select	0.4689
Where	0.1921
Count	0.1139
Null	0.07
Union	0.0641

Table 7 shows the selected features for classification. there will be a top 3, 4, 5 and 6 features will be ranked as mentioned in Table 6. This process is very important to know where features are important and not important in detecting the SQL Injection attack. The lowest score from the feature list is "union" feature, the second lowest score is "null" feature, and the third lowest score is "count". Therefore, these features will be removed according to the low value of Table 6 as shown in list of Table 7.

Table 7: Selected Features for Classification

Features / Top	Features List
3	Length, select, where
4	Length, select, where, count
5	Length, select, where, count, null
6 (All features)	Length, select, where, count, null, union

4.3 Classification Result

Random Forest (RF) algorithm and Support Vector Machine (SVM) algorithm from Machine Learning classification and 10-fold cross validation have been used to obtain the results of Accuracy, True Positive Rate, True Negative Rate and Precision. The result of classification experiments based on Table 7 in section 4.2 will be shown in Table 8.

Table 8: Comparison of Result with Support Vector Machine and Random Forest Classifier based on IG using HTTPParamsDataset

Features / Top	Algorithm	Accuracy	TP Rate	TN Rate	Precision
All Features{length, count, select, null, where, union}	RF	96.02%	0.960	0.946	0.961
	SVM	91.89 %	0.919	0.878	0.929
Top 5: Features without word "union"{length, count, select, null, where}	RF	96.02%	0.960	0.946	0.961
	SVM	91.89 %	0.919	0.878	0.929
Top 4: Features without word "union" and "null"{length, count, select, where}	RF	96.02%	0.960	0.946	0.961
	SVM	90.25 %	0.903	0.853	0.916
Top 3: Feature without word "union", "null" and "count"{length, select, where}	RF	96.02%	0.960	0.946	0.961
	SVM	90.25 %	0.903	0.853	0.916

Random Forest achieved higher accuracy with 96.02 percent than Support Vector Machine algorithm with 91.89 percent in all features. This shows that Random Forest classifier give better results in detecting SQL Injection using all features. Random Forest scored 0.960 in True Positive rate, 0.946 in True Negative rate, and 0.916 in Precision rate while Support Vector Machine scored score of 0.919, 0.878, 0.929 in TP rate, TN rate and Precision rate, respectively using all features. TP value shows that the dataset has positively correct classified features that contribute to negatively incorrect classified while TN rate shows that the dataset correctly predicts negative class. Therefore, based on the result, Random Forest classifier has higher TP rate and TN rate than SVM. This result shows that the Random Forest outperformed the Support Vector Machine classifier.

Random Forest achieved higher accuracy with 96.02 percent than Support Vector Machine algorithm with 91.89 percent in Top 5 Features. This shows that Random Forest classifier give better results in detecting SQL Injection using the Top 5 Features. Random Forest scored 0.960 in True Positive rate, 0.946 in True Negative rate, and 0.916 in Precision rate while Support Vector Machine scored score of 0.919, 0.878, 0.929 in TP rate, TN rate and Precision rate, respectively. TP value shows that the dataset has positively correct classified features that contribute to negatively incorrect classified while TN rate shows that the dataset correctly predicts negative class. Therefore, based on the result, Random Forest classifier has higher TP rate and TN rate than SVM. This result shows that the Random Forest outperformed the Support Vector Machine classifier.

Random Forest achieved higher accuracy with 96.02 percent than Support Vector Machine algorithm with 90.25 percent in Top 4 Features. This shows that Random Forest classifier give better results in detecting SQL Injection using the Top 4 Features. Random Forest scored a perfect score of 0.960, 0.946 and 0.961 in TP rate, TN rate and Precision rate respectively while SVM scored score of 0.903, 0.853, 0.196 in TP rate, TN rate and Precision rate, respectively. TP value shows that the dataset has positively correct classified features that contribute to negatively incorrect classified while TN rate shows that the dataset correctly predicts negative class. Therefore, based on the result, Random Forest classifier has higher TP rate and TN rate than SVM. This result shows that the Random Forest outperformed the Support Vector Machine classifier.

Random Forest achieved higher accuracy with 96.02 percent than Support Vector Machine algorithm with 90.25 percent in Top 3 Features. This shows that Random Forest classifier give better results in detecting SQL Injection using the Top 3 Features. Random Forest scored a perfect score of 0.960, 0.946 and 0.961 in TP rate, TN rate and Precision rate while SVM scored score of 0.903, 0.853, 0.196 in TP, TN and Precision rate, respectively. TP value shows that the dataset has positively correct classified features that contribute to negatively incorrect classified while TN rate shows that the dataset correctly predicts negative class. Therefore, based on the result, Random Forest classifier has higher TP rate and TN rate than SVM. This result shows that the Random Forest outperformed the Support Vector Machine classifier.

Overall, the implication of Information Gain and feature extraction helped the study significantly in terms of accuracy as well. This is because it reduces overfitting which gives less chances of making decisions based on noise. Overall, this study shows that Random Forest outperformed Support Vector Machine algorithms in SQL Injection Detection using the dataset because Random Forest uses multiple trees and merges them together to get an accurate and stable prediction.

5. Conclusion

SQL Injection queries are a cyber security vulnerability that can collect valuable data from users. The attackers use numerous query crafting techniques to avoid detection. In order to achieve those objectives, this research uses machine learning to design and develop a detection model for SQL Injection, and then uses Support Vector Machine and Random Forest algorithms to evaluate the detection model's classification Accuracy, True Positive, True Negative and Precision. The dataset is run using Random Forest and SVM algorithms. The results show promising results using features with 96.02 percent and 91.89 percent accuracy value for Random Forest and Support Vector Machine algorithm respectively without implementing Information Gain feature selection while the accuracies of Random Forest with implementing Information Gain are 96.02 percent, 96.02 percent, and 96.02 percent, whereas the accuracies of the Support Vector Machine are 91.89, 90.25 and 90.25 percent using the Top 5, 4 and 3 features. As future works, this study can be used to investigate further on SQL Injection detection using different features related to the SQL Injection detection. In addition, the scope of the research will be expanded by using other machine learning and deep learning algorithms to classify SQL injection attacks. This research may help other researchers to discover the best combination of features to detect SQL Injection and spread awareness about SQL Injection where it possibly can cause huge damage and problems to everyone by recognizing the pattern of SQL Injection.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Owasp.org. 2021. OWASP Top Ten Web Application Security Risks | OWASP. [online] Available at: <https://owasp.org/www-project-top-ten/>

[2] M. A. Azman, M. F. Marhusin, and R. Sulaiman, “Machine Learning-Based Technique To Detect SQL Injection Attack,” J. Comput. Sci., vol. 17, no. 3, 2021, doi: 10.3844/JCSSP.2021.296.303.

[3] M. S. Aliero and I. Ghani, “A Component Based SQL Injection Vulnerability Detection Tool,” in 2015 9th Malaysian Software Engineering Conference (MySEC), 2015, pp. 224–229, doi: 10.1109/MySEC.2015.7475225.

[4] O. C. Abikoye, A. Abubakar, A. H. Dokoro, O. N. Akande, and A. A. Kayode, “A Novel Technique To Prevent SQL Injection And Cross-Site Scripting Attacks Using Knuth-Morris-Pratt String Match Algorithm,” EURASIP J. Inf. Secur., vol. 2020, no. 1, p. 14, 2020, doi: 10.1186/s13635-020-00113-y.

[5] S. Sindhu Meena K.and Suriya, “A Survey On Supervised And Unsupervised Learning Techniques,” In Proceedings Of International Conference On Artificial Intelligence, Smart Grid And Smart City Applications, 2020, pp. 627–644.

[6] T. O. Ayodele, “Types Of Machine Learning Algorithms,” New Adv. Mach. Learn., vol. 3, pp. 19–48, 2010.

[7] Y. Abdulmalik, “An Improved SQL Injection Attack Detection Model Using Machine Learning Techniques,” Int. J. Innov. Comput., vol. 11, no. 1, pp. 53–57, Apr. 2021, doi: 10.11113/ijic.v11n1.300.

[8] Z. Chen, M. Guo, and L. Zhou, “Research On SQL Injection Detection Technology Based On SVM,” MATEC Web Conf., vol. 173, Jun. 2018, doi: 10.1051/matecconf/201817301004.

[9] A.Krishnan., “SQL Injection Detection Using Machine Learning,” Rev. Gestão Inovação e Tecnol., vol. 11, no. 3, pp. 300–310, Jun. 2021, doi: 10.47059/revistageintec.v11i3.1939.

[10] GitHub. 2020. GitHub - Morzeux/HttpParamsDataset [online] Available at: <https://github.com/Morzeux/HttpParamsDataset>

[11] S. Khalid, T. Khalil, and S. Nasreen, “A Survey Of Feature Selection And Feature Extraction Techniques In Machine Learning,” in 2014 Science and Information Conference, Aug. 2014, pp. 372–378, doi: 10.1109/SAI.2014.6918213.

[12] S. Lei, “A Feature Selection Method Based on Information Gain and Genetic Algorithm,” in 2012 International Conference on Computer Science and Electronics Engineering, Mar. 2012, pp. 355–358, doi: 10.1109/ICCSEE.2012.97.

[13] Y. Liu, Y. Wang, and J. Zhang, “New Machine Learning Algorithm: Random Forest,” in Information Computing and Applications, 2012, pp. 246–252.

[14] R. Rawat and S. Kumar Shrivastav, “SQL Injection Attack Detection Using SVM,” Int. J. Comput. Appl., vol. 42, no. 13, pp. 1–4, Mar. 2012, doi: 10.5120/5749-7043.