**Abstract**

In the technological world, the web is the most popular and common medium of global exchanges. Every day, a huge number of data are loaded through the web by millions of user. Therefore, security becomes important and challenging of all types of data on the web. SQL injection vulnerability has increased in volume and presenting signiﬁcant danger to the security of web application system. Injection attacks are the top web application vulnerability in the most recent Open Web Application Security Project (OWASP) Top 10 report. This has led to a growing interest in using machine learning to improve the detection of SQL injection vulnerability. Recent works in various detection implemented classifier fusion for improving detection accuracy. In this paper a fusion based model that enables fusion of various machine learning algorithms for improved better accuracy. A set of classification based algorithm is used for train the model. Then, n-fold cross-validation is used to validate the model performance. Finally, we have selected an ensemble learning algorithm for detection of SQL injection. Evaluation results show up to 99.2% accuracy.

**1 Introduction**

In this modern world the dependencies of internet application is increased due to the use of smart phones and various electronic devices. The uses of modern technology proof the industrial revolution in this world. With the increasing use of internet, the use of web applications also increased. All the users like clients, the employee should be always accessible to the web application at any time, any place all around the world. The web application has various vulnerability and the attackers apply different types of attacks to conceivable to hack. SQL injection is one of these attacks. Nowadays, the attackers use SQL injection attack to exploit a web application. Structured Query Language (SQL) is used to retrieve the important data from the web system database. The attackers use malicious code into the normal queries in a SQL injection to temper the original query. Attackers use SQL injection can retrieve important information such as internet banking password, ATM pins and user credential and also able to remove those information from database. With the uses of various web applications, a big amount of personal information of web users is been collected in web system databases continuously (Jose

Fonseca, et al., 2014). All types of data can be considered as the most valuable assets of organizations. The number of hacks in a web application has also increased for those data.

In current SQL injection detection research involves the use of machine learning. Popular detection method in this research is logistic regression, decision trees, support vector machines (SVM) and random forest. Most of the research uses one singular algorithm in machine learning technique to detect new attacks. A potential disadvantage of these methods is that they get a lower detection rate.

Research into the SQL detection depends on the availability of good data. Much current research uses text expression method of word2vec (Zhuang et al, 2018). The process that we are uses traffic generation and error capture coming from the web application. We are manually extracted data from the error message of the original attack load. . Our approach involves automated Fusion approach to detect SQL injection vulnerability as the basis for machine learning.

The rest of the paper is organized as follows: Section II. covers background material and related work. Section III. discusses our system design and implementation. Section IV. goes into detail about our experiment and results, and section V. concludes our paper.

**2 Background and Related Work**

**2.1 SQL Injection**

Structured Query Language injection is an injection technique used to attack sites in which the attacker inserts SQL characters or keywords into a SQL statement via unrestricted user input parameters to modify the intended query's logic [ ]. Whenever a request is generated from the user end, a query is generated. Suppose a webapp generates the following SQL statement:

SELECT author FROM books WHERE publisher = ‘Smith’ and published=1

If an attacker were to enter a string such as:

Smith’ OR 1=1—

into the search form, this would result in the following query:

SELECT author FROM books WHERE publisher = ‘Smith’ OR 1=1—‘ and published=1

**2.2 Machine Learning**

Machine learning is a technique that is capable of learning from experience. Machine learning is use to determine the most accurate way to classify network traffic as an attack or as normal traffic. In current research there are many machine learning algorithm in use. Support Vector Machines (SVM) classifier was used to solve the problem of SQL Injection using statistical features (Chen, Z, et al., 2018; Ladole, A, et al., 2016). In 2014, Joshi, A. et al., proposed a model that uses Naïve Bayes algorithm and Role Based Access Control method for detection.

In our project, we're using Support Vector Machine, Logistic Regression, k-nearest neighbours, Stochastic Gradient Decent and Random Forest algorithms. All these algorithms are discussed briefly below.

Logistic Regression Algorithm: Logistic Regression is an algorithm that is used to solve classification problems (Hosmer, et al. 2013). It is an extension of Linear Regression (Seber, et al., 2012) where the dependent variable is categorical it is a predictive analysis algorithm and based on the concept of probability. In logistic regression, the dependent variable is a binary variable that contains data represent as 1 (yes, success, spam etc.) or 0 (no, failure, not-spam etc.). The core function of the algorithm is logistic function or sigmoid function. It’s curve of S shape that can hold any real value and map it between 0 and 1. The standard logistic function:

f(x)=1/(1+e^(-x)) (1)

Where e is the basis of the natural logarithms and x is the value that we need to convert via the logistic function. Cen et al, 2015 proposed a detection method that decompiled program and user permission based on API calls. They find a probabilistic discriminative of a regularized logistic regression (RLR) algorithm.

Support Vector Machines: A Support Vector Machine (SVM) is an algorithm that classifies between the two classes through hyperplane which maximizes margin. SVM solve classification type problem as one that involves two classes where feature vectors are used for representing labeled training examples. Formally, two classes 𝒚𝒊∈ {−𝟏,} are to be perceived using M labeled training samples represented by 𝒙𝟏, 𝟏,….(𝒙M, 𝒚M), where individual training samples are represented by the feature vectors 𝒙𝒊 (Kolari, et al., 2006). SVM find out the optimal weight vector w such the following way when 𝒚𝒊’s are linearly separable.

‖𝒘‖𝟐 𝒊𝒔 𝒎𝒊𝒏𝒊𝒎𝒖𝒎 and 𝒚𝒊 ∗ (𝒘 ∗ 𝒙𝒊 − 𝒃) ≥ 𝟏 (2)

In combination with kernel functions SVM performs very well. Kernel trick can convert a linear model into non-linear model. We used only linear kernels in all our experiments. In 2011, Choi et al., trains an SVM classifier using N-Grams feature vectorization but would need various patterns to improve the accuracy of the approach. Kar et al., 2016, propose using SQL queries graph of tokens and centrality of nodes to train an SVM classifier but suffers from complexities. There have been previous approaches of numerical encoding of synthetic training data of SQLIA patterns for training a classifier to simulate SQL Injection attack prediction of any size (Uwagbole, S. et al., April 2016; Uwagbole, S. et al., July 2016).

k-Nearest-Neighbours(kNN): K-Nearest Neighbors is a non-parametric method that can be used for both classification and regression types problem (Hand et al., 2001). It is a straightforward calculation that stocks each accessible case and distinguishes new cases in view of a closeness volume (e.g., remove capacities). Since the start of the 1970s, KNN has been used as a non-parametric strategy as part of true reckoning and example acknowledgement. The result depends on whether KNN is utilized for classification or regression issue. From majority vote of its neighbors a fact is characterized and the case is assigned to its closest neighbors. If K = 1, then the case is easily imputed to the class of its closest neighbor. For continuous variables Euclidean, Manhattan, Minkowski these three distance measures are used. Hamming distance is a distance measure that is used for categorical variables. In 2014, Sharma and et al., proposed API based model that use naive Bayes and k-NN to detect SQL injection.

Stochastic Gradient Decent: Stochastic gradient descent also is known as incremental gradient descent. It is a repetitive process for optimizing a differentiable objective function. We can find out the gradient of the cost function of a single example at each iteration instead of the sum of the gradient of the cost function of all the examples. In this algorithm, since only one sample from the dataset is chosen at random for each iteration, the path taken by the algorithm to reach the minima is usually noisier than your typical Gradient Descent algorithm. But that doesn’t matter all that much because the path taken by the algorithm does not matter, as long as we reach the minima and with significantly shorter training time. SGD has been effectively wielded to large-scale and inadequate machine learning issues regularly imagined in text classification and natural language processing (NLP. In this case, as the data is exceptional, the classifiers scale to problems with more than 10 ^ 5 training examples and more than 10 ^ 5 features without difficulty. Overall SGD is efficient and ease of implementation.

Random Forest: Regression, classification and various machine learning problem solving with ensemble based Random Forest (RF) algorithm. A group of decision tress at training period is use for making Random Forest fusion. Tin Kam Ho introduced the first algorithm for random forests (Ho, et al., 1995). Random forests are a variety of tree predictors so that each tree relies on the values of an individually sampled random vector with the same availability for all forest trees (Breiman, et al. 2001). Bagging (Breiman, et al., 1996) and random feature selection technique is used in RF. Each tree is trained in bagging on a bootstrap sample of training data, and predictions are made by trees majority vote. RF inconstantly choose a subset of features to be divided at each node when growing a tree rather than using all features. RF performs cross validation in training stage using OOB for the evaluation of Random Forest algorithm. Milosevic et al., 2017 make a hybrid fusion classifier for analysis the source code and user permission. C.45, DT, random tree, RF, JRip, SVM and LR classiﬁers are used that model.

**3 System Design and Implementation**

The approach that we propose in this paper uses machine learning techniques to classify SQL injection vulnerable or normal. Data is captured in two places—HTTP traffic and the webapp server is captured. These two sets of data are then processed and correlated to create a separate dataset containing features from both datasets. Machine learning is done with the Weka Machine Learning Framework, and the machine learning algorithms used are evaluated for classification accuracy as well as efficiency in terms of time to build models and time to classify the training data with 5-fold cross-validation.

**3.1 Architecture**

The SQLFusion model is the fusion based classifier model. It used singular classifier and ensemble classifier. At the first of this model, the (SQLFusion) base classifiers are trained on various singular classifiers with K-fold cross-validation technique to validate the model. Then the outcomes of the base classifier are compared to fusion based algorithm. In fusion based algorithm used ensemble technique. And finally select the best accuracy model to build the final model. The SQLFusion model illustrated in Fig. 3.1:

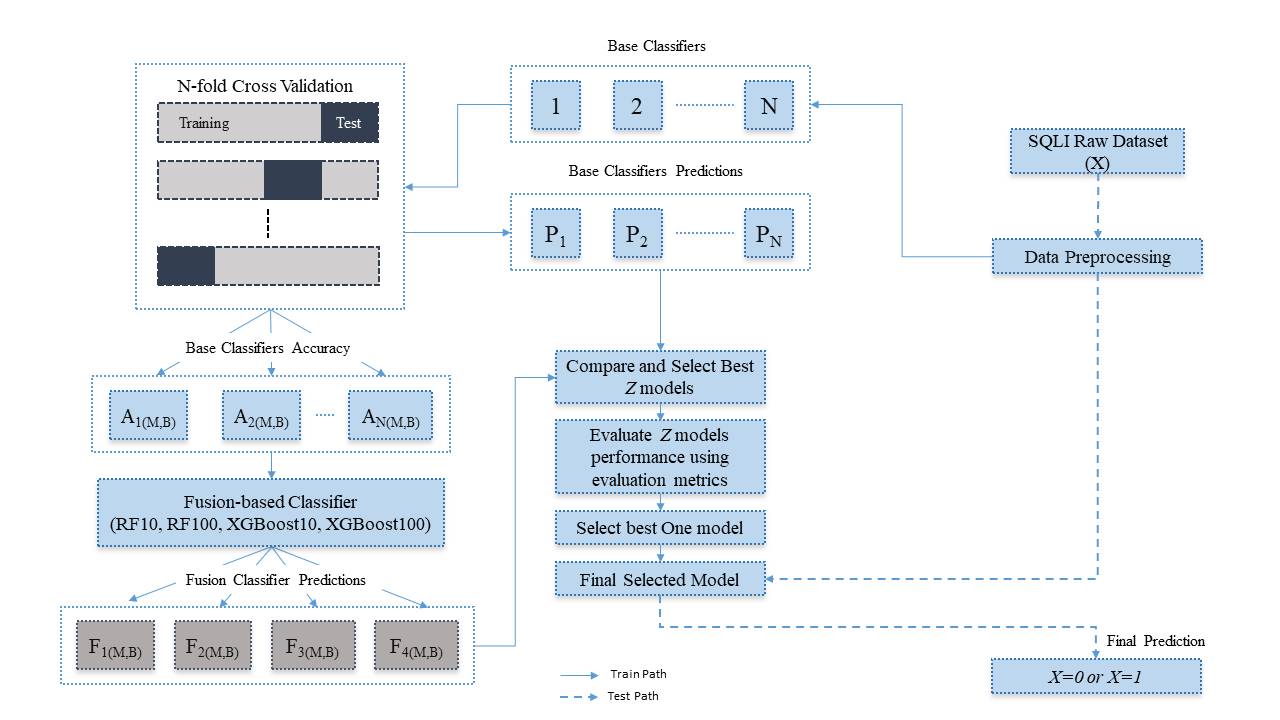


Fig. 3.1. Architecture of the proposed model

Fig. 3.1 illustrates is the model of SQLFusion. Solid arrows show the training path and dashed arrows show the testing path. First, each singular classiﬁer undergoes with K-fold cross-validation. We select three different Supervised Classification algorithm; Logistic Regression (LR) (Hosmer, et al. 2013), Support Vector Machine (SVM) (Hearst, et al. 1998), K-nearest neighbor (KNN) (Bijalwan, et al. 2014) for singular base classifier. Using various conﬁgurations, i.e., random tree-10, random tree-100 and random tree-1000 used for ensemble fusion based classifier, the random forest was used to make those individual classiﬁer models.

**3.2 Process of Data Generation**

The data generation process we're using consists of two phases: data capture, and pre-processing. This research uses a data set that contains 519 observations, 19 features .The data set items are labelled based on the exhibition of SQL Injection types characteristics which are: the presence of parameter tempering in injection point; error handling; sql syntax error; directory readability; server path disclosing etc. The data set items labelling are represented in binary values of 0 (SQL negative) or 1 (SQL positive).

**3.2.1 Data Pre-processing**

We applied different data pre-processing technique to make our data optimum. The pre-processing steps are shown in the Figure 3.2, and describe step by step in sequence.

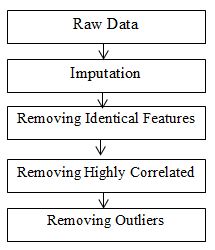


Figure 3.2: Data Pre-Processing Steps

**Removing Highly Correlated Feature**

Correlation is a mutual relation between two or more quantitative or categorical variables. It’s a measures of how things are related to each other. To measure how strong the relationship is between two variables we use Pearson’s Correlation Coefficient (PCC) (Pearson, 1920). The equation of calculating PCC shown in Equation 3.1:

(3.1)

Highly correlated feature is shown in fig 3.2 and after removing highly correlated feature is shown in fig 3.3

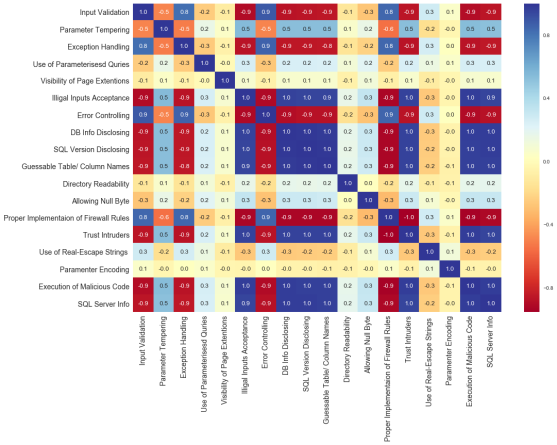


Fig 3.2: Before Removing Highly Correlated Feature

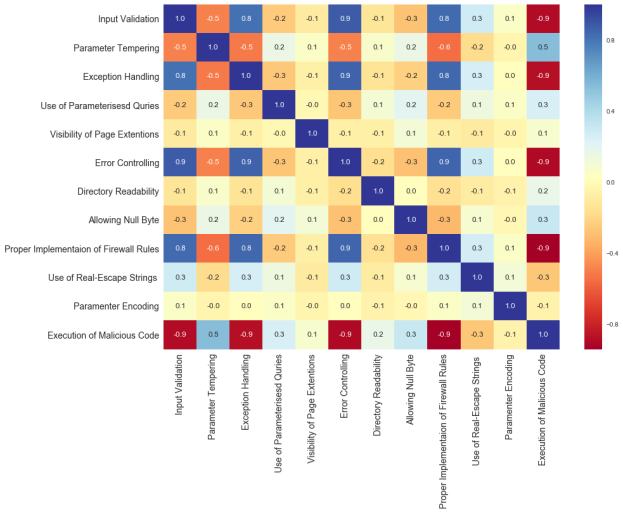


Fig 3.3: After Removing Highly Correlated Feature

**4 Experiment and Results**

We used following metrics for evaluating our model. We considered condition positive (P) as SQL Injected and condition negative (N) as SQL not injected. Correctly classified SQL injected is termed as hit or true positive (TP), correctly classified SQL not injected as true negative (TN), incorrectly classified not injected to SQL is termed as false positive (FP) or false Hit, incorrectly classified SQL not injected is termed as false negative (FN) or false miss.

Accuracy: In a classification problem accuracy score is the fastest way to evaluate a set of prediction. Accuracy score is a ratio of total number of true positive and negative out of all examples that were made. The formula for calculating accuracy in binary classification problem shown below:

(3.5)

Precision: Precision identifies the frequency with which a model was correct when predicting the positive (1’s) class. The formula of precision shown below:

(3.6)

Sensitivity: Sensitivity (also known as recall) measures the proportion of actual positives that are correctly identified as such (e.g., the percentage of SQL Injection which are correctly identified as having the condition). The formula of Sensitivity shown below:

(3.7)

Specificity: Specificity also referred to as the true negative rate that measures the ratio of actual negatives correctly identified as such (e.g., the percentage of vulnerable SQL Injection that is correctly identified as not having the condition). The formula of Specificity shown below:

(3.8)

F1 Score: F1 Score is a statistical score based on precision and sensitivity. It is the combination of precision and recall. The formula of F1 Score shown below:

(3.9)

**Performance of SQLFusion**

In order to evaluate SQLFusion, we divided the dataset into training and testing. The experiment is based on the optimum dataset that is listed on the Table 3.2. The ratio of training and testing was 80:20. The 5-fold cross validation was used to validate the SQLFusion model using the training set. Table 4.1 shows the accuracies of the four base classiﬁers and Table 4.2 shows the accuracies of 5-fold cross-validation.

Table 4. 1: Performance of base classifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1** |
| SVM | 0.4759 | 0.5027 | 0.4218 | 0.3595 |
| KNN | 0.4778 | 0.5496 | 0.4712 | 0.3686 |
| SGD | 0.8388 | 0.8149 | 0.8082 | 0.7664 |
| LR | 0.9418 | 0.9454 | 0.9487 | 0.9414 |

From the above Table 4.1, it is clearly shown that SGD has the highest performance in terms of the evaluation metrics. The accuracy of SGD model is 92.30% which is a remarkable performance compare to others. On the other hand, KNN has the lowest performance among all. It has an accuracy score 81% which is far less than from the highest one.

Table 4. 2: Performance of 5-fold cross validation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **F1** | **Sensitivity** | **Specificity** |
| SVM | 0.92 | 0.93 | 0.92 | 0.96 | 0.88 |
| KNN | 0.81 | 0.84 | 0.82 | 0.92 | 0.71 |
| SGD | 0.96 | 0.96 | 0.96 | 0.92 | 1.0 |
| LR | 0.89 | 0.90 | 0.89 | 0.85 | 0.94 |

From the above Table 4.2 SVM and KNN accuracy is very low and the base classifier accuracy Table 4.1 shows that SVM and KNN are very high. It clearly indicates that SVM and KNN are over-fitted. To avoid the over-fitted problem, dropout regularization is applied to out Fusion approach model.

After building this base classifier our SQLFusion model has trained from the output of the base model. Then our model was trained those ensemble classifier: 1. RF10; 2. RF100; 3. RF1000. Multischeme is a method of meta learning. The multischeme method evaluates all the classiﬁers and in order to select the best model. In those experiments, we selected 5-fold cross-validation. In the Table 4.3, shows the performance of SQLFusion model:Table

4.3: Performance of the SQLFusion model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1** |
| RF10 | 0.9708 | 0.9701 | 0.9702 | 0.9701 |
| RF100 | 0.9805 | 0.9801 | 0.9811 | 0.9802 |
| RF1000 | 0.9902 | 0.9903 | 0.9901 | 0.9902 |

From the above table it clearly indicates that the RF1000 model has the highest performance in terms of the evaluation metrics. The accuracy of the RF1000 model is the 99% which is a remarkable performance compare to others. The multischeme approach evaluates three fusion based classifier and in order to select the best RF1000 for the SQLFusion model.

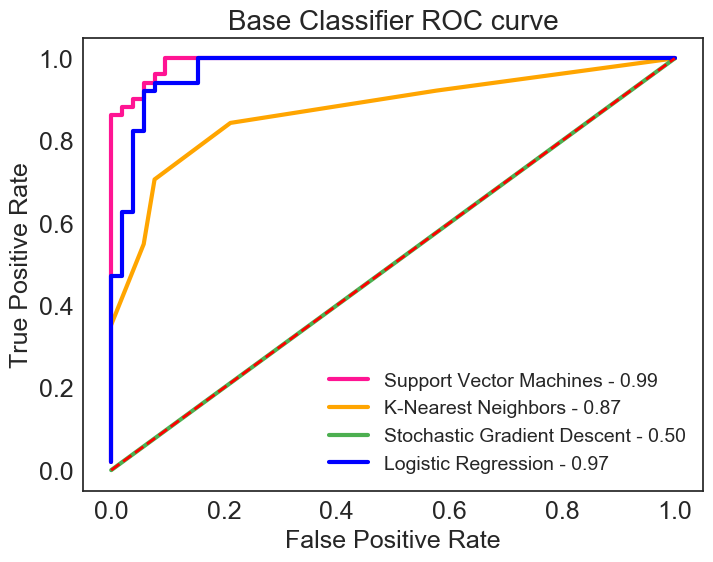


Figure 4.1: ROC curve showing differences between 4 models

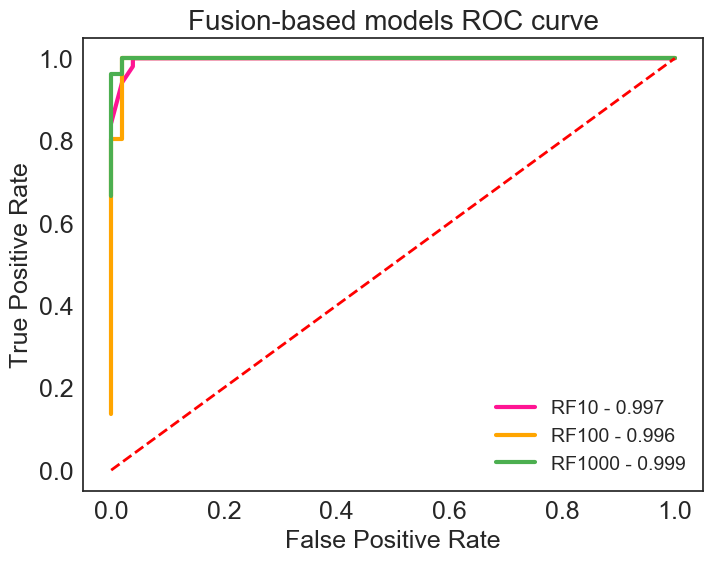


Figure 4.2: ROC curve showing differences between Fusion models.

The Area under the ROC curve is a performance measure for classification problem with all possible threshold settings. Receiver Operating Characteristics (ROC) is a probability curve whereas AUC represents degree or measure of separability. Higher the AUC value, better the model is at predicting 0’s as 0’s and 1’s as 1’s. The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where FPR is on the x-axis and TPR is on the y-axis. The TPR defines the number of positive results that are correct for all positive samples available during the test.On the other hand, FPR defines the number of positive incorrect results in all negative samples available during the test. A point in the upper left corner or coordinate (0, 1) of the ROC space would yield the best possible prediction method, representing 100 percent True Positive Rate and 100 percent False Positive Rate. Also known as the (0, 1) point is a perfect classification. The diagonal divides the ROC space, points above the diagonal represent good classification results; points below the line represent bad results.

In Figure 4.2, the ROC curve demonstrate that RF1000 performs so well compare to the others model. The diagonal separate the ROC space, above the diagonal indicates good performance. The SQLFusion model outperform to all other models.

**Conclusion and future work:**

In this paper, we proposed a novel general purpose

classifier fusion approach (SQLFusion) for SQL Injection detection. We empirically evaluated SQLFusion using real datasets. The results presented demonstrates its effectiveness for improving performance using both nonensemble and ensemble classifiers. Furthermore, we showed that our proposed approach can outperform. The proposed classifier classifies the test set with 99.2% accuracy. We think that our method should be tested against larger real datasets to evaluate the efficiency. The proposed method can be enhanced for detection of SQL injection vulnerable.

**References**

1. Ross, K., Moh, M., Moh, T. S., & Yao, J. (2018, March). Multi-source data analysis and evaluation of machine learning techniques for SQL injection detection. In Proceedings of the ACMSE 2018 Conference (p. 1). ACM.
2. Uwagbole, S. O., Buchanan, W. J., & Fan, L. (2017, May). Applied machine learning predictive analytics to SQL injection attack detection and prevention. In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) (pp. 1087-1090). IEEE.
3. Joshi, A., & Geetha, V. (2014, July). SQL Injection detection using machine learning. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) (pp. 1111-1115). IEEE.
4. Ladole, A., & Phalke, M. D. (2016). SQL Injection Attack and User Behavior Detection by Using Query Tree Fisher Score and SVM Classification. International Research Journal of Engineering and Technology, 3(6).
5. Bockermann, C., Apel, M., & Meier, M. (2009, July). Learning sql for database intrusion detection using context-sensitive modelling. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 196-205). Springer, Berlin, Heidelberg.
6. Choi, J., Kim, H., Choi, C., & Kim, P. (2011, September). Efficient malicious code detection using n-gram analysis and SVM. In 2011 14th International Conference on Network-Based Information Systems (pp. 618-621). IEEE.
7. Kar, D., Panigrahi, S., & Sundararajan, S. (2016). SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM. Computers & Security, 60, 206-225.
8. Uwagbole, S. O., Buchanan, W. J., & Fan, L. (2016, April). Numerical encoding to Tame SQL injection attacks. In NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium (pp. 1253-1256). IEEE.
9. Uwagbole, S. O., Buchanan, W., & Fan, L. (2016, July). Applied web traffic analysis for numerical encoding of SQL injection attack features. In European Conference on Cyber Warfare and Security (p. 393). Academic Conferences International Limited.
10. A. Sharma and S. K. Dash, “Mining API calls and permissions for Android malware detection,” in Cryptology and Network Security. Cham, Switzerland: Springer Int., 2014, pp. 191–205.
11. L. Cen, C. S. Gates, L. Si, and N. Li, “A probabilistic discriminative model for Android malware detection with decompiled source code,” IEEE Trans. Depend. Secure Comput., vol. 12, no. 4, pp. 400–412, Jul./Aug. 2015.
12. Yerima, S. Y., & Sezer, S. (2018). DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection. IEEE transactions on cybernetics, (99), 1-14.
13. D. Gaikwad and R. Thool, “DAREnsemble: Decision tree and rule learner based ensemble for network intrusion detection system,” in Proc. 1st Int. Conf. Inf. Commun. Technol. Intell. Syst., 2016, pp. 185–193.
14. A. Balon-Perlin and B. Gambäck, “Ensembles of decision trees for network intrusion detection systems,” Int. J. Adv. Security, vol. 6, nos. 1–2, pp. 62–77, 2013.
15. M. Panda and M. R. Patra, “Ensembling rule based classiﬁers for detecting network intrusions,” in Proc. Int. Conf. Adv. Recent Technol. Commun. Comput., 2009, pp. 19–22, doi: 10.1109/ARTCom.2009.121.
16. A. Zainal, M. A. Maarof, S. M. Shamsuddin, and A. Abraham, “Ensemble of one-class classiﬁers for network intrusion detection system,” in Proc. 4th Int. Conf. Inf. Assurance Security, 2008, pp. 180–185, doi: 10.1109/IAS.2008.35.
17. N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, “Machine learning aided Android malware classiﬁcation,” Comput. Elect. Eng., vol. 61, pp. 266–274, Jul. 2017.
18. S. Y. Yerima, S. Sezer, and I. Muttik, “Android malware detection using parallel machine learning classiﬁers,” in Proc. 8th Int. Conf. Next Gener. Mobile Apps Services Technol. (NGMAST), Oxford, U.K., Sep. 2014, pp. 37–42
19. Clauset, A. (2011, August). A brief primer on probability distributions. In Santa Fe Institute.
20. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.
21. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. IEEE Intelligent Systems and their applications, 13(4), 18-28.
22. Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70.
23. Seber, G. A., & Lee, A. J. (2012). Linear regression analysis(Vol. 329). John Wiley & Sons.
24. Kolari, P., Finin, T., & Joshi, A. (2006, March). SVMs for the blogosphere: Blog identification and splog detection. In AAAI spring symposium on computational approaches to analysing weblogs.
25. D. Hand, H. Mannila, P. Smyth.: Principles of Data Mining. The MIT Press. (2001)
26. Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.