IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-2088)
Available online at: https://www.ijariit.com

Reusable AI-based ensemble model for detecting SQL injection in service-oriented architectures

Sudarshan M.
sudarshan.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Pranava B.
pranavab.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Dr. G. S. Mamatha
mamathags@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

ABSTRACT

Cyber security has become one of the most sought-after domains in the field of computer science. Protection of computing resources and information against disruptive cyber threats have garnered utmost attention in recent times, owing to the conventional methods used in the field that often fall short of detecting or preventing the ever-evolving collection of malwares. With the advent of new technologies such as Machine learning and Artificial intelligence, it is possible to streamline the approaches in the field of Cyber security. These technologies can be used to detect and prevent malicious content, thereby developing successful security solutions. The right AI tech could help us process huge volumes of threat data, discover anomalies and effectively eliminate potential threats. Currently, the most common approach involves using regular expressions to sequentially compare the incoming request or its vector with a predefined set of signatures. Though this approach is widely prevalent, it falls short in terms of accuracy. This is due to the fact that the signatures are not updated often, and several logical problems or loops come up when regular expressions are used within thousands of individual rules. In this project, we aim to identify various injections among neutral input vectors using ML models and will be predicting whether the vectors are injection or not. An ensemble of a number of ML models is used to build a voting mechanism to have an accurate prediction. For the sake of demonstration, the application consists of a frontend built using react and a python flask backend server.

Index Terms- *SQLi, Artificial Intelligence, Machine Learning, Logistic Regression, Ensemble model, Neural Network, Gaussian Naïve Bayes, SVM, KNN, Decision Tree, Cybersecurity, Flask, React*

I. INTRODUCTION

Security on the web is a very important aspect as a large amount of data is made available to everyone all over the world through the internet. Cybersecurity deals with the protection of privacy as well as protection against any threats that one may face on the internet. Hackers, viruses, malwares, spywares, phishing attacks, ransomwares, trojans, identity theft, data manipulation, DoS attacks are just some of the threats that we face today. Cyber-attacks aim at modifying or destroying sensitive information which can in turn be used to extort money, destroy companies and people's lives. With the increase in use of software and more and more companies making their businesses online cyber security has never been more important as well as relevant. The Internet has become a critical part of everyone's lives and is also bringing increased benefits into people's lives. The Internet was designed with the notion of mutual trust. However, this assumption leads to a diverse variety of attacks and methods need to be identified to mitigate the issues caused by these attacks. With the wide use of distributed system technologies and inter application communication via APIs network security is of high importance while building applications. Any sensitive information transmitted needs to be encrypted and must not be available for any kind of man in the middle attacks. The number of threats are increasing at an alarming rate as well and there is a dire shortage of qualified cyber security experts in the world. Cyber security must be automated and there must be a system to learn and adapt to new kinds of threats. That is where Artificial Intelligence comes into picture. Artificial intelligence based cybersecurity involves the use of various machine learning algorithms to predict and mitigate vulnerabilities in modern day applications. The machine learning models are trained from time to time on various kinds of real time data which helps the algorithms to identify patterns in them. Sometimes not one algorithm is able to identify the threat and we would have to make use of multiple models to make a final decision. A data mining framework is needed to frame and feed the algorithms with data . Necessary preprocessing steps need to be taken to enhance the model performance and accuracy of the outcomes.

II. LITERATURE SURVEY

SQL injection detection was tested with many machine learning models out of which decision tree algorithms were proved to provide promising results. The decision tree algorithm seemed to be very effective with respect to the time complexity.[1].

In case of web applications, an Elastic Pooling CNN model was used. This model when compared with other machine learning techniques provides a high accuracy and FI score. [2] According to the paper [3] a new algorithm is proposed which is effective against varying malwares. The results seem to have no false positives and are highly resistant against hackers.

The usage of ensemble models is recommended to overcome the drawbacks of the individual algorithms used. Same can be said about hybrid classifiers which give better results in detecting SQL injection attacks.[4] With increasing network traffic over the internet, new methods have been proposed to track network security vulnerabilities over the network. The paper [5] deals with network security where different labels are assigned to specify the threat using Independent component analysis and common spatial patterns methods

Software that the administrators can use to watch or monitor the activities of an individual while browsing so that they can be saved or prevented from being a victim of the cybercrime. The domains are classified as Blacklisted or Greylisted by comparing them with the library of the websites that are stored in the software. [6]

By applying unsupervised learning besides supervised learning on the feature vectors (Op code frequency), malwares can be classified. Random Forest algorithm performs better than deep neural networks when the feature vector is Op code. Deep Neural Network with two hidden layers (DNN-2L), with 4 hidden layers, and with 7 hidden layers were considered to learn features at different levels of abstraction [7]

Applications of ML techniques evaluated in detecting malicious insiders in networked systems of corporations and organizations. Features consisting of details like the no. of actions on a shared PC, no. of HTTP downloads, and average size of attachments in the emails sent were considered as feature vectors. ML Algorithms or Techniques like Logistic Regression, XGBoost, Neural Networks and Random Forest algorithms were used for Data Analytics [8]

Static and dynamic analysis used for feature extractions and a feature selection approach using PCA is presented. SVM algorithm used for building the model and results have been presented. Whenever a new application is installed, its MD5 value is extracted and compared with the malicious MD5 present in the SQLite. If a new value exists in SQLite then the system will alert the user to delete it because of the presence of malicious information. Else when there is no malicious information, the APK is submitted to the server. [9]

When the traditional solutions to Cybersecurity become inadequate, advances in cryptographic & AI techniques show promise to help in countering threats that are posed by the adversaries. Types of cybersecurity threats like DoS attacks, SQL Injection attacks, password attacks etc. ML techniques like Decision trees, naive bayes, SVM, KNN etc can be used to detect cyberattacks [10]

III. IMPLEMENTATION

A. Design

The following figure(Fig 1) is the flowchart of the project. The order of events is:

1. A user goes to the login page in the frontend application and fills in username and password credentials
2. Once the user submits his credentials, it creates a POST request and the credentials are sent to the backend application to check for SQL Injection possibility.
3. The backend application uses ensemble machine learning techniques consisting of different ML Models to detect SQL Injection in the input credentials.
4. If the credentials are genuine and no sql injection is detected, then user login is successful.
5. If SQL injection is detected then the user is denied login to the website with a push notification saying “SQL Injection Detected in login/password field.”

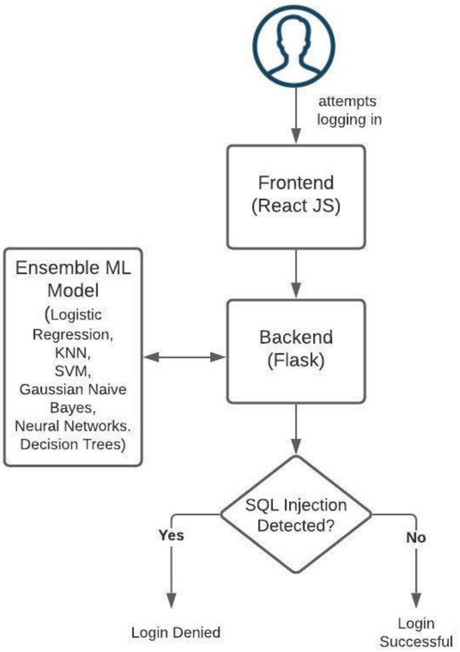


Fig 1: Flowchart of the project

B. Implementation Details of the ML Models

The implementation of the ML models involves phases like Data Collection, Data Preprocessing, EDA (Exploratory Data Analysis), building the models and evaluation & comparison of models in terms of metrics like accuracy, precision and recall.

Data Collection

The data used in this project is taken from Kaggle. It has two columns : “sentence” and “label”. “sentence” column has the inputs in the form of strings and the “label” of 1 means it is an SQL Injection while a label of 0 means it is a normal sentence and not an SQL Injection attempt. The dataset has about 4200 rows and 2 columns, which is split into training and testing dataset using sklearn library in python.

Data Pre-processing and EDA:

The rows having missing values are removed. The sentences are transformed into the vectors based on frequency count of each word that occurred in the entire text, this is done using the Count Vectorizer tool by scikit-learn library in python. The process is called vectorization and is done since in order to apply machine learning algorithms on textual data, it needs to be transformed into the vector representations.

Below figures(Fig 2 & 3) show the count or frequency of occurrence of each token or letters in the specified range of rows in the dataset. By this it can be concluded that the words appearing in SQL queries like SELECT, INSERT, UNION, from, table name etc have higher frequency than in SQL Injection attacks. Also the characters like single quote(‘), brackets, 1, =, 1,-,+ etc have higher frequencies since these are some of most common characters used in the SQL Injection attacks.

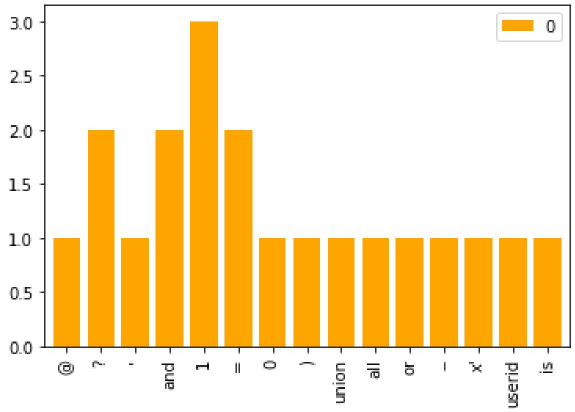


Fig 2: Word frequencies in a section of dataset

The label column has about 3072 fields as 0(No SQL Injection) and about 1128 fields as 1(SQL Injection). The following figure(Fig 3) shows the graphical representation of the same in the form of histogram.

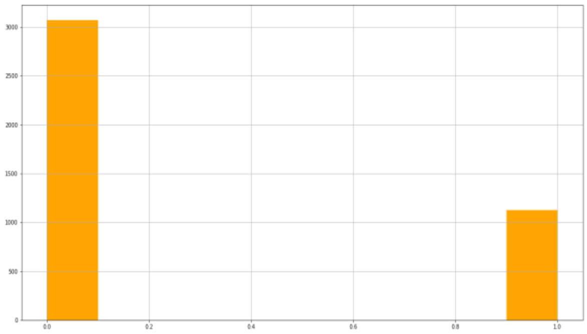


Fig 3: Frequency of unique values in label column

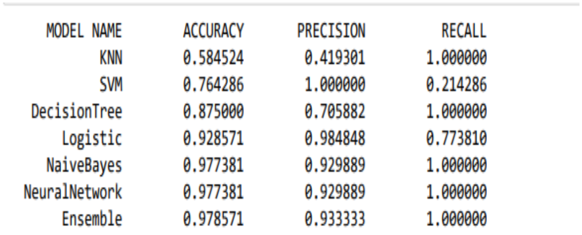
Model Construction

In this project, several ML Models were constructed and compared for their efficiency and performance. The following are the models:

- Logistic Regression
- Neural Network
- Gaussian Naïve Bayes
- SVM
- KNN
- Decision Tree
- Ensemble Method

IV. RESULTS

The ML models used in carrying out this work are logistic regression, naive bayes, decision trees, SVM (Support Vector Machine), KNN, Neural Networks as well as an ensemble of these models (Voting Classifier). Almost all the models did well in predicting the results, although their efficiencies (memory usage and time) varied. Some models like logistic regression, gaussian naive bayes, decision trees and neural networks were fast and accurate in predicting the output for given inputs, while the remaining models were little slow in prediction.



```
  MODEL NAME    ACCURACY   PRECISION     RECALL
         KNN    0.584524    0.419301   1.000000
         SVM    0.764286    1.000000   0.214286
DecisionTree    0.875000    0.705882   1.000000
    Logistic    0.928571    0.984848   0.773810
  NaiveBayes    0.977381    0.929889   1.000000
NeuralNetwork   0.977381    0.929889   1.000000
    Ensemble    0.978571    0.933333   1.000000
```

Fig 4: Comparison of different ML Model results

The above figure(Fig 4) shows that almost all the models showed good results in terms of accuracy, precision and recall. KNN and SVM showed relatively less accurate results as compared to the remaining model. Ensemble model performed better than almost all the other models in terms of all the statistical measurements like accuracy, precision and recall. The Ensemble model gave an accuracy of 97.85%, recall of 100% and with 93.33% precision.

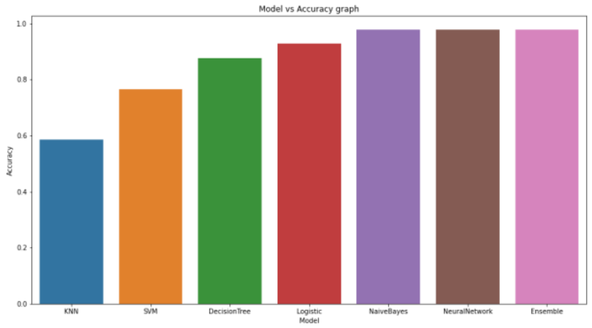


Fig 5: Models Vs Accuracy Graph

The above figure(Fig 5) shows that Naïve Bayes, Neural Network and Ensemble method have the highest accuracies as compared to the other ML models.

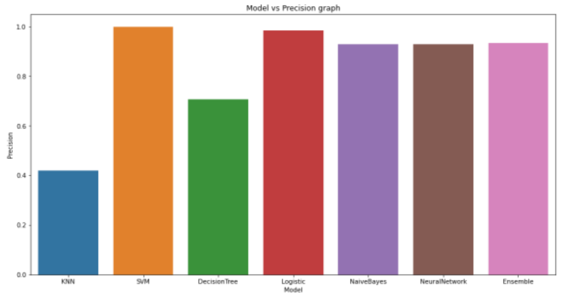


Fig 6: Models Vs Precision Graph

.The above figure(Fig 6) shows that SVM has the highest Precision value, followed by Ensemble method and Logistic Regression.

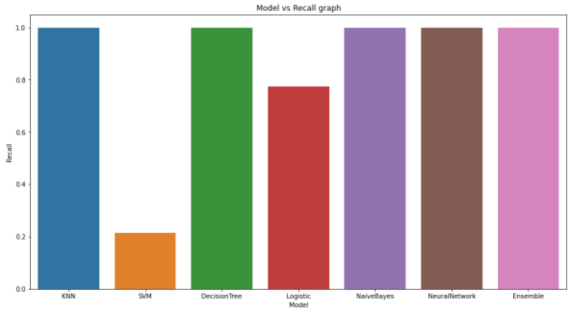
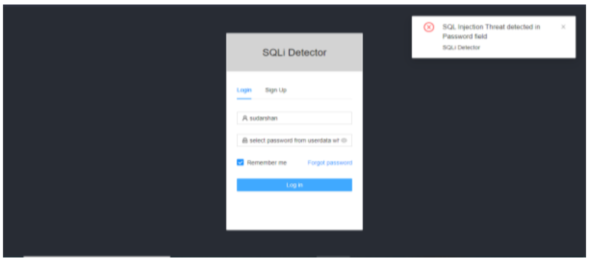


Fig 7: Models Vs Recall Graph

The above figure(Fig 7) shows that SVM did not perform well in terms of Recall, whereas all other models performed well.



SQL Injection Threat detected in Password field
SQLi Detector

sudarshan
select password from userdata wh

Fig 8: Frontend detecting SQLi threat in the password field

The above picture shows the frontend app detecting an SQL injection threat in the password field while the user tries to login, since the username looks genuine and the password is similar to an SQL Query.

V. CONCLUSION

The SQLi Detector is an application which has a wide range of applications and has a lot of scope for improvements as well. The main reason for taking an artificial intelligence based approach was to build a system which can keep track of the latest threats and mitigate them. The underlying machine learning algorithms can be adapted to future improved algorithms. The system can be developed into a full fledged plugin or extension into browsers to completely prevent any kind of SQLi threats in any domain or website on the web. Lot of learnings were made while developing this project. The importance of cybersecurity in network interactions with the increasing use of distributed system architectures and microservice architectures. The steps necessary to take an application from development to production requires a thorough testing of the application for any security vulnerabilities as well. The application must be made open source , so that more machine learning models can be integrated into the existing architecture and so that there may be a scope for early detection of new kinds of vulnerabilities as well. With the rising usage of web applications it is imperative that we use sanitised data everywhere input fields are used. Even then there might be situations where a bypass of this security feature is possible. Such scenarios require a strict validator to be present in order to secure the company as well as the users data. SQL databases such as MySQL, Oracle databases are one of the most widely used databases all over the world. SQLi is one of the most basic as well as highly vulnerable threats that exist in modern times and a simple and elegant solution is what all applications need. The application deals with dev ops, networking, API design, Artificial intelligence, etc. The wide scope of the application helped in gaining a deep insight into the complexities involved in building a cybersecurity application or tool. The treats are updated on a daily basis and new vulnerabilities are created every day. The need for an automated highly efficient framework to analyse and mitigate various security vulnerabilities is of utmost importance in modern days. The existing system can be modified to detect different kinds of vulnerabilities and expanded on the scope of operation by using transfer learning techniques. The voting technique can be replaced by a new invented architecture to improve the accuracy on a broader scope of data. Artificial intelligence in cybersecurity is still in its infancy and the field is definitely going to grow at a rapid pace over the next few years

VI. ACKNOWLEDGMENT

The authors would like to thank Dr. Mamatha G S for the guidance in the proposed work. Pranava B and Sudarshan M contributed equally to this paper.

VII. REFERENCES

[1]. K. Kamtuo and C. Soomlek, "Machine Learning for SQL injection prevention on server-side scripting," 2016 International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2016

[2]. X. Xie, C. Ren, Y. Fu, J. Xu and J. Guo, "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN," in IEEE Access, vol. 7, pp. 151475-151481, 2019

[3]. M. Christodorescu, S. Jha, S. A. Seshia, D. Song and R. E. Bryant, "Semantics-aware malware detection," 2015 IEEE Symposium on Security and Privacy (S&P'15), Oakland, CA, USA, 2015

[4]. U. S. Musa, M. Chhabra, A. Ali and M. Kaur, "Intrusion Detection System using Machine Learning Techniques: A Review," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020

[5]. R. Trifonov, O. Nakov and V. Mladenov, "Artificial Intelligence in Cyber Threats Intelligence," 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Mon Tresor, Mauritius, 2018

[6]. P. Srivastava, R. Sharma and A. K. Pandey, "A novel web application to detect malicious intent people in society," 2020 International Conference on Electrical and Electronics Engineering (ICE3), Gorakhpur, India, 2020

[7]. Rathore H., Agarwal S., Sahay S.K., Sewak M. (2018) Malware Detection Using Machine Learning and Deep Learning. In: Mondal A., Gupta H., Srivastava J., Reddy P., Somayajulu D. (eds) Big Data Analytics. BDA 2018. Lecture Notes in Computer Science, vol 11297. Springer, Cham.

[8]. D. C. Le, N. Zincir-Heywood and M. I. Heywood, "Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 30-44, March 2020

[9]. Long Wen and Haiyang Yu, “An Android malware detection system based on machine learning”. in AIP Conference Proceedings 1864, 020136 (2017);

[10]. S. Zeadally, E. Adi, Z. Baig and I. A. Khan, "Harnessing Artificial Intelligence Capabilities to Improve Cybersecurity," in IEEE Access, vol. 8, pp. 23817-23837, 2020

[11]. A. Joshi and V. Geetha, "SQL Injection detection using machine learning," 2014 International Conference on Control, Instrumentation,Communication and Computational Technologies (ICCICCT), 2014, pp.1111-1115, doi:10.1109/ICCICCT.2014.6993127.

[12]. S. O. Uwagbole, W. J. Buchanan and L. Fan, "Applied Machine Learning predictive analytics to SQL Injection Attack detection and prevention," 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017, pp. 1087-1090, doi: 10.23919/INM.2017.7987433.

[13]. A. Tajpour and M. J. z. Shooshtari, "Evaluation of SQL Injection Detection and Prevention Techniques," 2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks, 2010, pp. 216-221, doi: 10.1109/CICSyN.2010.55.

[14]. Q. Li, F. Wang, J. Wang and W. Li, "LSTM-Based SQL Injection Detection Method for Intelligent Transportation System," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4182-4191, May 2019, doi: 10.1109/TVT.2019.2893675.