PREVENTING SQL INJECTION USING MACHINE LEARNING BY ENSEMBLE TECHNIQUES

Mengistu Bonsa Deco

A Thesis Submitted to the Department of Computer Science and Engineering School of Electrical Engineering and Computing

Presented inPartial Fulfillment Of The Requirements For The Award Of The Degree Of Master OfScienceIn Computing Engineering.

Office of Graduate Studies

Adama Science and Technology University

Aug, 2022

Adama, Ethiopia

Preventing SQL Injection Using Machine Learning by Ensemble Techniques

MengistuBonsa Deco

Advisor: Dr. BahiruShifaw

A Thesis Submitted to the Department of Computer Science and Engineering School of Electrical Engineering and Computing Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Aug, 2022

Adama, Ethiopia

DECLARATION

I declare that this thesis entitled“Preventing SQL Injection Using Machine Learningby Ensemble Techniques” in partial fulfillment of the requirements for the award of degree of Master of Science in Computer Science is an authentic record of myown work carried out from April 2022 to Jun 2022 for my advisor **BahiruShifaw** (Dr), This work has not submitted to any University for similar purpose. The references used in this Thesis are duly recognized by proper citations.

MengistuBonsa		
Student Name	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled “Preventing SQL injection using Machine learningby Ensemble Techniques” prepared under my guidance by Mengistu Bonsa submitted in partial fulfillment of the requirements for the degree of Mater’s of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. BahiruShifaw _____ _____ _____

Advisor Name Signature Date

APPROVAL SHEET

I, the advisor of the thesis entitled “Preventing SQL injection using Machine learningby Ensemble Techniques” and developed by Mengistu Bonsa, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Bahirushifawu (Ph.D) _____ _____

Major Supervisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by MengistuBonsahave read and evaluated the thesis entitled “Preventing SQL injection using Machine learningby Ensemble Techniques” and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science Engineering

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

__Eyob_N_____  _____

External Examiner Signature Date

Finally, approval and acceptance of the thesis are contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGMENT

To get success in our life the key point is use of time andstrength. To get strength my must be healthy, this is the gif of God. So that first of all Ithankmy great God. Second, in my life the role of my family has significant meaning so I would like to express my darling for them. Next, I would like to express my special gratitude for my adviser Dr. BahiruShifaw who is not only adviser on this thesis but also teach me to do more in my life. I also tank my wife because she supports me to cover my class on time.On the last all you stand after me I would thanks by name god.

Table of Content

List of Tables

List of Figures

List of Abbreviations and Acronyms

SQLI	Structured Query Language Injection
SQL	Structured Query Language
LDAP	Lightweight Directory Access Protocol
OWASP	Open Web Application Security Project
DVWA	Damn Vulnerability Web Application
JRIP	Joint Reserve Intelligence Program
RF	Random Forest
ANN	Artificial Neural Network
SVM	Support Vector Machines
SQLCIA	SQL command injection attacks
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
LAMP	Linux Apache My SQL PHP
XPATH	XML Path Language
TCP	Transmission Control Protocol
HTTP	HyperText Transfer Protocol

ABSTRACT

The world recently is one by the globalization. Using Internet multiple platforms and web-applications has become a quite common phenomenon in the recent times. The web-based applications that accept critical information from users store information in databases. The applications and the databases connected to Internet are susceptible to all kinds of information security threats. There are many types of threats attacks. SQL Injection attacks is the top ten vulnerabilities based on web applications. By using injection, attackers can quickly destroy the target websites and control the whole web application system, to make any modifications, steal and damage up to identity theft that affects the business or organization.How we prevent it, in this paper by ensemble techniques the *SQL injection attack is prevented using Machine learning.*

Keywords*:- SQL, Union query, SQLI,Machine learning, SQLICA,*

CHAPTERONE

INTRODUCTION

1.1 Background Study

Population that is accessing the internet for all countries of the world, the Internet can be used via a computer, mobile phone, personal digital assistant, games machine, digital TV etc. The Internet has many important applications, the three most important are e-mail, web browsing, and peer-to-peer services. E-mail is the most widely used and successful of Internet applications.[1]

Most of the applications that we use every day are web-based applications. Organizations choose to make the applications accessible over the Internet to increase the exposure they gain. Being exposed to Internet increases the security challenges that come along with uncontrolled access. With the growth of Internet, we are used to performing various kinds of transactions online.

The web applications on the Internet are usually dynamic and driven by one or more databases in the backend. So web applications have made our lives much simpler, security & privacy of the sensitive data in the backend databases has become a big concern. But many programmers are not aware of this serious security threat.[2]All the data entered by the users during these transactions on web applications or websites is stored in some kind of a database. Relational Databases can be communicated with a language called Structured Query Language/ SQL. [3]

SQLInjection is a web vulnerability caused by mistakes made by programmers. It allows an attacker to send commands to the database that the website or web application communicates with. SQL Injection is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures.

They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify and delete records in the database. Now a day's most web applications are being hacked using SQL Injection method.Till now many approaches have been proposed for detecting the injecting query but practical implementation is not possible. As we have three main technique i.e. Prevention, detection and correction. So prevention is a

little bit complication because none of the proposed method provides the exact solution for preventing injected query [4].

To prevent SQL Injection attacks machine learning algorithms is used. The SQL Injection Detection Using Machine Learning is no single perfect algorithm or technique in machine learning. In previous researches to detection SQL Injection using by thetype of supervised learning algorithmclassifiers has been implemented. In this paper organized the basic idea of SQL injection attack, the related work done and how to protect SQL injection using Machine learning byEnsembletechniques.There are many other Web-attack vulnerabilities such as:-

1. Buffer errors
2. Path traversal
3. Code injection etc,

They don't necessarily depend on user's input for the victim's exploitation, but SQLIA is instances of input-based vulnerabilities

The relative prevalence of some of these attacks is shown in the figure below. This data was obtained from HACKING & TRICKS[7], SQL Injection comprises a significant fraction (25%approximately) of all web-based attacks.

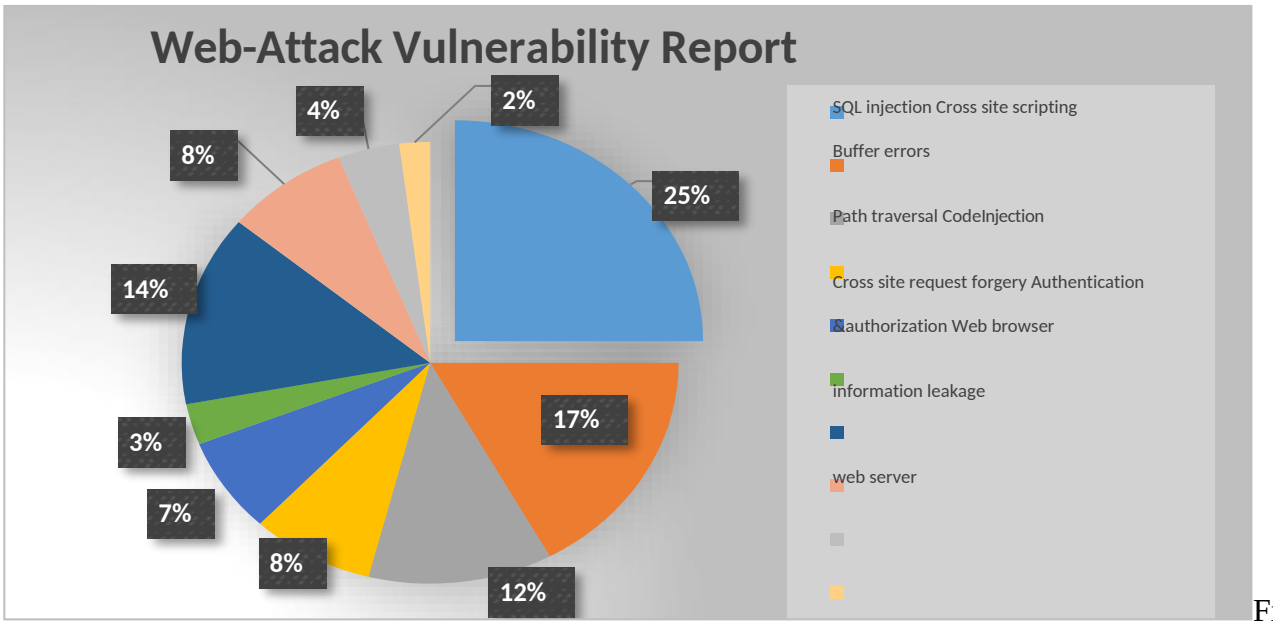

Figure 1.1 Web-attack Vulnerability Report

There are three broad type of SQL.

a. In band SQLI
b. Blind SQLIand
c. Out of Band SQLI,

Form the three categories or parts of SQL injecting the broadly used is, In band SQL Injection. In band SQL injection also divided into two parts Union Based SQLI andError Based SQL Injection. The Union Based SQL Injection attack makes attackers to theft, modify and to destruct sensitive data such as personally identifiable information and usernames and passwords. They also compromised database server to attack other on the same network.[7]

Ensemble learning is a powerful machine learning algorithm. Ensemble learning techniques is the combine of predictions multiple machine learning models.The basic intention of this thesis is to select a solution for preventing SQLIA by Ensemble learning techniques type from many by selects the simplest one which is Max Voting. In addition, the method should provide good performance. Ensemble technique is "improving classification accuracy by aggregating the predictions of multiple classifiers". These techniques use base classifiers such as random forests, byusinga votingscheme from the predictions madebythe base classifiers.

1.2 Statement of the Problem

Many websites are vulnerable to hackers seeking to perform attacks because backdoors are left open by poor coding practices that do not adhere to good security standards. Hackers unleash hordes of bots to look for sites that have these coding weaknesses.vulnerability makes the attacker's to execute commands or access data without appropriate authorization.[8]When attackers get authority over the data or server easily generate SQL injection and process simply.Criminals may use it to access to sensitive data, customer information, personal data, trade secrets, intellectual property and more. After the data is processed by the attackers it is difficult to prevent. SQL injection is a subset of code injection attack. The attacker embeds a piece of code in a computer program via user input/data entry. Execution of the infected program provides the attacker with improper access to the computer program or application. Although web-based applications are developed to achievedesirable access to the database by authorized users, the attacker uses them to his benefit to gain access to confidential information stored in the database.

1.3 Research Questions

Based on the problem, this research aims to answer the following questions:-

Q1:- How to protect web application from attacks that changes the structure of the query?

Q2:-How to build good prediction to prevent SQLI attacksusing Machine learning?

Q3:- What are the methods useto evaluate the accuracy of detecting mechanism?

1.4 Objective of theStudy

General Objective

The general objective of the study is to develop a model that will accurately prevent SQL injection attack.

Specific Objectives

The below list of specific objectives will be addressed to achieve the general objective.

- To review theories, ideas and methods of previous works to identify and understand the existing gaps.
- To prepare the data sets to make them appropriate for building the model.
- To develop a machine learning model or algorithm that will have better accuracy than the previous one.
- To evaluateand test theprepared model of preventing SQL injection.

1.5 Significance of the study

SQL Injection is a type of an injection attack that controls a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They also use SQL Injection to add, modify, and delete records in the database. Many ways of protecting, prevent and detect website from SQL injection attacks/many techniques are implemented, but until now there is a big hunting of information and data in world. This makes the world, world of attackers.

So, to protect world from damage and to prevent bad Gus/ theft/ from their works control method must be implemented. There are many way to protect world internet from attackers,this makes me to do so far on SQLI attacks, by the help of existing attack preventing models. Machine learning is use to identify the accuracies of the models as well as it use to create accurate predictive model by combine many weak learning models or group of machine learning algorithms together.

1.6 Expected outcome of the study

Most organizations use web applications to collect, store, process, and represent their required information using back-end databases. If an application does not preserve its security completely, intrusion to its back-end database can be performed easily. The useful orimportant of these tasks is for service provider companies to protect their customer’s data and for software developer to create application that has strong security system.

1.7 Scope and limitation

Scope of the Study

The study focuseson existing models accuracy to Preventing SQL Injection Using Machine learning that identify SQL Injection attacks by having more efficient and accurate with good prediction.

Limitation of the Study

- Due to the security issues gating labeled dataset form known site is difficulty. There is a limited amount of dataset that are available online, because company or organizationis not willing to share their access log files.
- During experimentation the comparison of the proposed model with the other models is hard to evaluate and implement.
- Any work must be limited based on time and other resources. Depend on this the limitation that take over me is shortage of time and getting full resources/related works/ that be updated.

1.8 Operational definition

The study, compeer and evaluate between the SQL injection solutions before and after. This evaluation helps to known the performance of the model. The model identifies same types of SQL Injections correctly by solution enumerated its capabilities in terms of Detection, Prevention, Generate Report and Classification.

1.9 Organization of the Thesis

The remaining of this thesis is structured as follows:

Chapter Two: Discuss the background literature and related works regarding SQLIA and a Summary of related works.

Chapter Three- Describe research methodologies such as dataset collection.

Chapter Four-Describe the proposed Model for preventing SQLIAs.

Chapter Five- Describe the Result of Prevention SQLIA

Chapter Six - Analysis the proposed solution of SQLIA

Chapter Seven - Concludes the research and further work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Theory of SQLI

SQL injection is the extension or modification of a web application's SQL statement by an attacker in order to extract or update information in the database that they are not authorized to access.[9]There is much existing research that attempts to examine and mitigate SQL injection attacks. One technique in use involves static analysis of code to form a model that can be compared to dynamic queries at runtime to detect SQL injection attacks.

Classification method is used to classify the incoming traffic as a SQL Injection or plain text. These malicious queries can bypass authentication and authorization and can finally alter, modify, and delete the database. The proposed a robust model for detection of SQL injection attack queries from normal queries and plain text. The foremost step is to create a balanced dataset that contains normal and malicious SQL queries.[9]

It also introduced plain text to this dataset in order to make the proposed model perform well and differentiate malicious queries from normal and plain text.Various combinations of parameters are tuned and tried together.Manually coded SQL injection attacks as the basis for machine learning.

2.2 Cause ofSQLIA

SQLIAs that are caused by code injection, SQL manipulation and function call injection. There are two ways of injecting malicious data in a database.

1. Data manipulation through the user'sinput
2. Data injection by seeding an indirecttrigger

2.2.1 Data manipulation using user's input: Is the attacker injects malicious code by masquerading as the user's input deep inside the SQL query this makes it difficult for the SQL Data Base Management System (DBMS)to prevent illegalaccesstothedatabase.

2.2.2 Datainjectionbyseedinganindirecttrigger:themaliciouscodeisthe SQL statements injected into persistent storage.The table record is considered a trusted source, but it could indirectly trigger an attack contents maliciouslyinserted inthe table are used later.To makeuseof this kind of vulnerability, it is necessary to submit suitable data in one location and then use some other application's function to process the data in the attacker's intended way.

2.3 Types of SQLIA

The types of SQL injection attacks implemented through user input data are list with the ways in which can be executed asbelow:-

1. **Tautologies:** The attack location for a tautology is in an input field or URL that is vulnerable and openly exploitable. This type of attack requires a conditional statement to execute successfully. There are many models of SQL injection attacks based on the "=" operator, but many relational operators such as "<", ">", "<=", ">=" also relate to a result in a true or a false fashion. tautologies can be used for causing more advanced attacks, such as changing the admin passwords without checking for the old password [10], as shown in the following query:

   ```
   UPDATE users SET password= 'I got u' WHERE Username= "admin" '_ _'
   AND password = '1=1'
   ```

2. **Logically incorrect queries:**-The performance of this type of attack depends upon the number of requestsrequiredtoperformthedataextractionprocessandthetimethatisrequiredto gather enough data to enable the attacker to cause a successful attack. This injection vectors have a history of depending upon the Database structure and the signature of the SQL Injection attacks to which the database is vulnerable.A few ways to investigate the backend database of a system to fingerprint the vulnerability. lists are include:-

   a) Unique error based vulnerability detection
   b) DBMS Function Invoking
   c) Fingerprinting using Inference Attack

3. **Union query**: Union is a relation that includes all the tupelo that are either in the first or in

the second tupelo or in both the tupelo, without including the duplicates in the tupelo, where the tupelo is a setof data. For the union attack to work, both the tupelos must be compatible this means it must have the same number of tupelo and the domain of each attribute in column order should be same in both the tupelo.

A union attack is caused by specifying a union of the original query and a second query with the same structure. This type of query attack requires a good amount of reconnaissance. The prices of all entities in categories A and B are displayed as well by using the UNION SELECT statement in the SQLquery.

4. **Piggy backed queries :-**Piggy backed attacks are used by attackers to extract data, modify data, perform denial of service,and/or execute remote commands, by adding extra statements to an existing statement. The attacker retains the original query but includes new queries that piggy- back on the original query. As a result, the DBMS receives multiple SQL queries.

   The first query executes normally whereas the queries following the original query carry out the attacker's chosen logic in the database. This kind of attack requires the use of delimiters to join multiple queries together and it has a dependency on the database configuration because it cannot be executed if the configuration of the DBMS doesn't allow multiple SQL statements in a single string.

5. **Alternate encoding**:- This is an attack that tries to bypass protection systems by representing the user data in a different language that the attacker hopes will be ultimately decoded by the DBMS. The effectiveness of SQLIA using alternate encoding depends upon the efficiency of the attacker in masking the attack sequence to make it look like a simple SQLquery.For example, SQL servers currently support six alternate encoding schemes such as char, ASCII, Unicode, nchar, convert and cast.

2.4 Related Work

The Following are the related work that covered different important strategies to prevent SQL injection attacks.Thissectionbrieflydescribestheresearchandworkdoneof

1. “SQLInjectionDetectionUsingMachineLearningTechniquesand Multiple Data Source”by Kevin Ross 2018 and SQL Injection
2. “Detection Using Machine Learning” By SonaliMishra 2019ondetectingSQLInjection attacks and preventing them effectively.

SQLInjectionDetectionUsingMachineLearningTechniquesand Multiple DataSource

The research work done so far on detecting SQL Injections can be broadly classified into two types of approaches. The first approach is to securely write the source code itself to enforce enough input validation for SQL queries. The second approach is to deploy additional software to verify the SQL queries being passed through web applications to be executed across the database.

Lately some researchers have also highlighted the importance of using these two approaches in combinationtoachieveamorereliableSQLInjectiondetectionmodel.Inthissection,both the approaches and the research work done in those fields are discussed.

ThedevelopedaJDBCcheckerthatisastaticanalysistooltocheckforerrorsinSQL strings and verify them for potential malicious queries. It verifies the SQL strings for correctnessandpromisestoidentifyandindicatepotentialerrorsinSQLqueries.Theway it works is instead of dynamically checking each query while it is generated at runtime, it staticallycreatesalistofallpotentialSQLstringsthatcouldbeexecutedacrossaparticular application and then analyses all those potential SQL strings for malicious content and semantic errors.

The problem with this approach could be multiple, including the high storagethatwillberequiredforstoringallthepotentialSQLqueries,andhowcouldatool generateallpossiblepotentialqueriesforanapplication.Thereisahugepossibilitythatit would miss out on predicting SQL query statements that were actually executed. To overcome this limitation of static analysis, a tool named CANDID was developed, that standsfor‘CandidateEvaluationforDiscoveringIntentDynamically’.Itworksonthe concept of programmer intent.

The concept of programmer intent describes how a SQL querystructureshouldlooklikeifitisformedexactlyaswasintendedbytheprogrammer of the application. There is a pattern observed in SQL injection attacks that the malicious SQL query executed across database always has a different structure than the one that was intended by the programmer. Authors of this paper believed that identifying this difference in structure could be a significant step towards successfully identifying and preventing SQL injection attacks.

The way this tool achieves its goal is that it dynamically creates the programmer intended SQL query structure when the program reaches a location where it will generate and execute a SQL query. This generated programmer intended SQL query is then compared with the actual SQL query that was passed by the user, to identify and prevent the malicious queries from being executed.

Later a few researchers came up with the idea of combining the use of static analysis and dynamic monitoring to efficiently prevent SQL Injection attacks. AMNESIA is a tool that was developed on this idea of using both static and dynamic approaches.The application code itself contains information to produce all the possible SQL queries that could be generated by the application. All these possible legitimate queries are generated and stored for comparison. At the same time dynamic monitoring is done of SQLqueriesgeneratedatruntime,andeachdynamicallygeneratedSQLqueryiscompared to the list of possible SQL queries. If no match is found for a SQL query generated as a resultofsomeinputfromauser,inthelistofpossiblelegitimateSQLqueries,thequeryis classifiedasmaliciousandisnotallowedtobeexecutedonthedatabase.

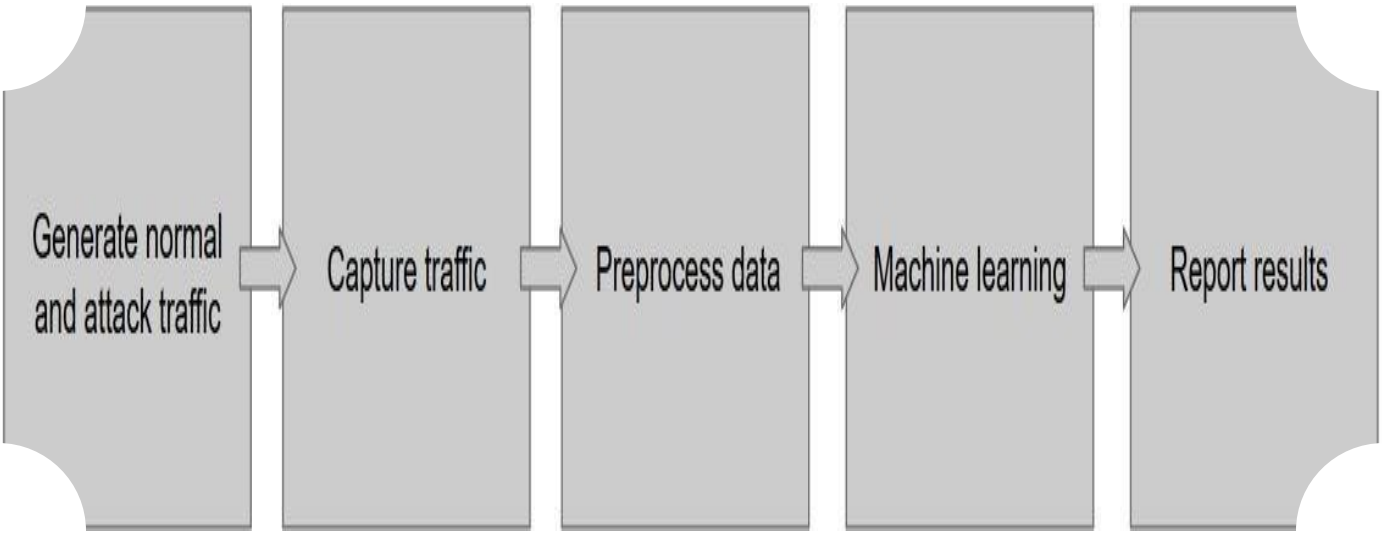

Figure2.1 SystemSteps

Thismethodhas its own set of drawbacks, biggest one being that it is not a hundred percent accurate and could generate a lot of false positives. It used similar approach of comparing the actually generated queries with the one that should have been generated (programmer intended). The only difference in this approach is that it achieves the results by using Parse Trees. Parse trees are generated by parsing the statements by using the syntax of the language that statement in written in.

This is a dynamic approach only, that is, it verifies the SQL queries generated at runtime and has no previously stored set of possible legitimate queries. In this approach two parse trees are generated, one is generated by using the input from the user and generating a tree that would be generated if this was a legitimate query and the other tree is generated by processing the actual input and seeing how the

query actually will be executed.

Now there is one thing about malicious SQL queries, is also discussed previously, that the behavior of such queries is always different than the programmer intended behavior. Hence in case of a malicious query, this paper promises that the two parse trees generated will be different, and hence the SQL Injection attacks can be identified more efficiently.

With the development of AI and Machine Learning, some researchers proposed using the machine learning algorithms to prevent SQL Injection attacks. This paper detects SQL Injection attacks using a machine learning algorithm called Naïve Byes. Naïve Byes is a classification machine learning algorithm that assumes that a particular incident is unrelated to and is independent of other all other incidents. In this paper Naïve Byes classifier is used to classify between malicious and non-malicious SQL queries.

To train the model they have used a training dataset that consists of both malicious and non-malicious SQL queries and also every query in this training data is labeled. Labeling the data helps the model to learn what is malicious and what is non-malicious. This type of model is called a supervised machine learning model. Once the model has been trained it is then used on the test dataset to verify if the model is classifying the SQL queries correctly.

The model suggested in this paper promises to even detect those SQL Injection attacks that are new and whose signatures are not known. They will also use machinelearning todetect SQL Injection attacks but with a different machine learning algorithm.

SQL injection attacks, and web-based attacks in general, continue to be a major issue in the security of financial, health, and other critical data, and this problem only increases in importance as more societal processes become more dependent on the internet. In this project we have proposed a multi-source data analysis system for increased accuracy in detection of SQL injection attacks, and have

established that the algorithms we have experimented with such as rule-based and decision treealgorithms have in our experiments achieved accuracy close to that of Neural Networks and are much better in terms of time necessaryto build models and execution time when classifying testing data.

The main problem is having shortage of usability and efficiency and also have shortage of provides better accuracy.

2.5 Detection Using Machine Learning

SQL injection is the extension or modification of a web application's SQL statement by an attacker in order to extract or update information in the database that they are not authorized to access. Suppose a webapp generates the following SQL statement:

```
SELECTauthor,title,year FROM books WHERE publisher ='Wiley' and published=1
```

Ifanattackerweretoenterastringsuchas:

```
Wiley'OR 1=1–
```

intothesearchform,thiswouldresultinthefollowingquery:

```
SELECTauthor,title,yearFROMbooksWHEREpublisher='Wiley'OR1=1–'and published=1
```

This query would return every row in the database where the publisher is Wiley, or 1=1, which is always true, thus returning all rows in the database.

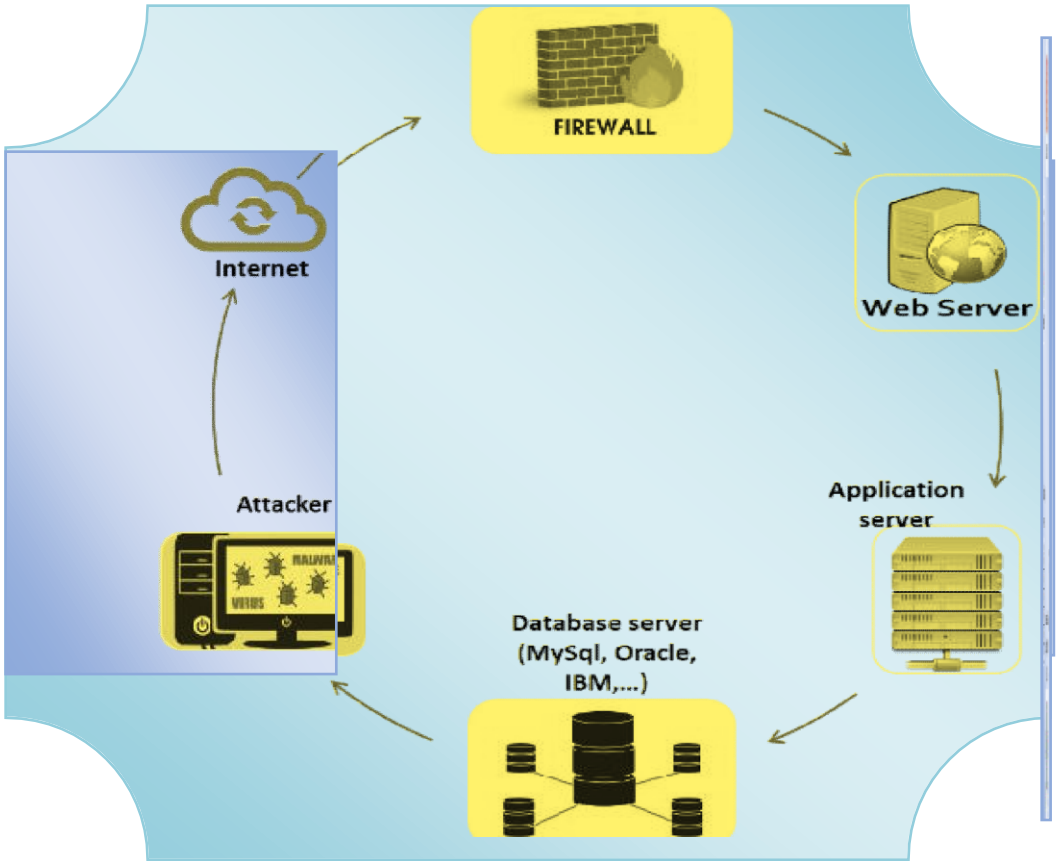

Figure 2.2 Process of SQL injections

There is much existing research that attempts to examine and mitigate SQL injection attacks. One techniquein useinvolvesstaticanalysisof codetoform amodelthatcanbe comparedtodynamic queries at

Runtime to detect SQLinjection attacks. In Lee et al.the authors propose a system that performs analysis of PHP-based websites to evaluate the SQL queries and then processes these queries byremoving parameters to form general query structures, a process they refer to as query transformation.

These are then compared to liveSQLqueriesgenerateddynamicallyandaquerythathasadifferentstructurewillbeflaggedasanattack. In addition to removing parameters,generalize SQL queries into structural elements, and were able to achieve a 100% detection rate with their technique.

More architecture has been explored for Intrusion Detection Systems. In the authors create a clustered architecture using Raspberry Pi computing devices. This project uses the HIHAT honeypot system and the SQL injection detection technique proposed in Lee et al. The system uses a load-

balancing server to route computation to the Raspberry Pi cluster.

The data propose is distributed, multi-honeypot architecture. They use several different honeypot systems connected to a front-end server, and the system is modular to accommodate the additionorremovalofhoneypotsystems.Thisdistributed architecturehastheadvantagethatit's capableof gathering moredata than anyone of the honeypot systems, and the authors mention that the attacks theydetected were primarily targeting SSH, as well as MySQL, MSSQL, and telnet, and originate primarily in China and the US.

Tamara and Jane propose a complicated architecture they refer to as a "virtual honeynet" designed to maximize the time spent by an attacker in the system and thus the amount of data gathered about attackers. This system uses the SNORT IDS to detect malicious traffic which is then diverted to the honeynet.The virtual honeypots use the HIHAT system as well as the Seek data capture tool as components, and combine these into a complicated architecture designed to confuse and delay attackers.

In the Honeydroopsystem, the authors propose a dynamic architecture using Hardtop for data collection. This system uses dynamic allocation based on currently observed network conditions, with the goal of efficient utilization of resources and increased security. The authors mention that in their experience these dynamically created honeypots receive much more traffic than those that are statically generated. The authors useSnort fortraffic capture, but the architectureismodular so anyIDS system could be used.

Machine learning is a technique for automated discovery of patterns in data. In a network security context, machine learning is used to determine the most accurate and efficient way to classify networktraffic as an attack or as normal traffic to best protect digital assets while allowing normal business to proceed uninterrupted. There are many machine learning algorithms in use in current research. In the authors are using Naive Byes and Byes Net techniques which are creating probability models to classify traffic. In the authors are using a decision tree technique, and are also evaluating their results in terms of performance characteristics. Another popular technique in current IDS research is the use of Neural Networks, discussed below.

In our project, we're using Decision Tree, Rule-based, Support Vector Machine, Neural Network, and Random Forest algorithms. Background information about each of these is discussed brieflybelow. We also discuss feature selection, which is the reduction of dimensions in data.

Decision tree algorithms are a categoryof classification algorithms that create a predictive model based on the values of features in the dataset, at each point in the process choosing a feature that divides the dataset to maximize information gain, which is a measure of how well each featureisabletopredicttheclassthatthedatarecordbelongsto,essentiallythepurityorhomogeneityof thecoveredrecordsintermsofclassprediction.Adecisiontreealgorithmiteratesthroughthefeaturesofthe dataset and chooses the feature with the highest information gain and the best coverage of the dataset, in other words it prefers predictive decisions that contain larger numbers of records. In the authors are using a custom decision tree technique, and are also evaluating their results in terms of performance characteristics. Their accuracy on SQL injection data is in the 86-87% range.

Rule-based algorithms are similar to decision tree algorithms in that they are iterating through features of the dataset and selecting those which best predict the class label. The output of rule-based algorithms is aset ofrules forclassification oftheform Condition → Class, an example ofwhich is.

```
(Gives Birth = no) ^ (Aerial Creature = yes) → Birds
```

These rules are evaluated in terms of their coverage and accuracy. Coverage is the ratio of instances ofthe dataset that are covered by the rules; accuracy is the ratio of the instances that belong to the predicted class to the total instances covered by the rule. Rule-based classifiers are typically comparable in performance to decision tree algorithms, but are considered to be very easy to interpret compared to other classification algorithms.

SVM algorithms are a classification technique based on statistical methods which work well with high dimensional datasets. SVM attempts to create a decision boundarybetween classes using instances of training data referred to as support vectors. This is donebyfindinga maximal hyper-planeseparation between instances ofthedifferent classes.

SVMis considered to be better at finding globally optimal solutions than rule-based or neural network techniques. In Al. the authors are using an SVM classifier, and are experimenting with different centrality metrics as input to the SVM classifier. In Al the authors are using an SVM classifier with n-gram based analysis rather than text-based as we’re using for the current project. The

authorsevaluate SVM in comparison with a KNN (K-nearest neighbor) algorithm and achieve better results with SVM. In theauthors compare SVMwith KNNand also Naïve-Byes algorithms and find that the best results are obtained with SVM.

Neural Network algorithms were originally designed to simulate the human brain structure, with a system of linked nodes conceptually related to neurons. At the simplest level, thereare input nodes that are exposed to features of the dataset, and output nodes corresponding to the classificationoutcomes.Theoutputnodesrepresentaweightedcombinationoftheinput,andtheseweightsare updated during the learning process based on learning rate and classification error. A Multi-Layer Perception is a type of Neural Network with one or more hidden layers of nodes intermediate to the input and output nodes. An issue with Neural Networks, as mentioned, is the potential slowness of the training phase. "Training an ANN is a time-consuming process, especially when the number of hidden nodes is large." Another issue discussed in current research is that the results of Neural Networks can be difficult to interpret and in some cases this may leave applications vulnerable to adversaries. In this authors are using MultiLayerPerception and Support Vector Machine (SVM) techniques in combination to improve accuracy, and compare their results to those of other algorithms such as J48decision tree and JRip rule-based, discussed below, and have been able to achieve improved accuracy with their technique.

The purpose of ensemble techniques is "improving classification accuracy by aggregating the predictions of multiple classifiers". These techniques use base classifiers such as those mentioned above and then perform classification byusinga votingscheme from the predictions madebythe base classifiers. One such technique used in this project is random forests, which use decision trees as their base classifier.

FeatureSelection.Feature selection is usedto reduce the dimensionalityof dataforperformance reasons. This is a technique to remove redundant features, which are providing duplicate information of another feature, and irrelevant features, which areprovidinglittleornouseful informationto help with classification. [13]The authors use an information gain technique for feature selection and are using it. In our project we’re using CFS, correlated feature set, in combination with a genetic search algorithm.

Gathering data for SQL injection research is generally done in two primary ways: capturing actual

web traffic coming into an organization or honeypot, or the generation of realistic simulated traffic. Both approaches have their advantages and disadvantages. Real web traffic is of course the most realistic, but it can be difficult to determine which packets belong to an attack. It can also be difficult to obtain this type of traffic, as organizations are typically reluctant to share web traffic due to privacy and security concerns.

Anotherissueis that asimpleresearch honeypot maycapturemostlyautomated scans generatedbycommon attack tools that would be more easily captured in a controlled lab setting. Simulated attacks have the advantage that they are controlled so that normal and malicious traffic is easily distinguished for labeling purposes, and a good assortment of attacks can be included based on the latest techniques.

There are many tools in use by researchers in an effort to generate realistic attack traffic to test proposed detection and mitigation strategies. As mentioned, use the attack simulation tool Paros. Themost commonlyused attack tool as determined bysearches on Google Scholaris SQL Map. Among others are using SQLMap to generate malicious traffic. Theydiscuss and have tested several examples, including SQLMap and manually coded attacks. Many researchers previous research and the current projects are using manually coded SQL injection attacks to generate attack traffic. Small analysis of machine learning techniques for both accuracy and performance, Cannot detect other types of web-based attacksSQL Injection attacks remain to be one of top concerns for cyber security researchers.

Signature based SQL Injection detection methods are no longer reliable as attackers are using new types of SQL Injections each time. There is a need for SQL Injection detection mechanisms that are capable of identifying new, never before seen attacks. Applying machine learning to the field of cyber-security is being considered by many researchers. Since machine learning in cyber-security is still a developing research area, there are not many libraries and open source tools that are machine learning specific and apply to problems related to threats and attacks.

The SQL Injection detection problem is approached by applying machine learning algorithms. Classification method is used to classify the incoming traffic as a SQL Injectionorplaintext. [14]Twomachinelearningclassificationalgorithmsareimplementedon the problem, which are, Naïve Byes Classifier and Gradient Boosting Classifier. Naïve Byes classifier machine learning model

provides results with an accuracy of 92.8%.

Ensemble learning methods are said to provide results with better accuracy as they implement multiple simple classifiers to improve error and accuracy. Hence Gradient Boosting Classifier from ensemble learning is selected to be implemented on the SQL Injection classification problem.

Various combinations of parameters are tuned and tried together and an accuracy of 97.4% is achieved by the Gradient Boosting classifier. The machine learning approaches can be used for SQL Injection detection, and Gradient Boosting classifier algorithm provides better accuracy than Naïve Byesapproach.

The main problem is having shortage Small analysis of machine learning techniques for both accuracy and performance, Cannot detect other types of web-based attacks.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Overview

To enhance understanding of SQL injection, it is better to have good understanding of the kinds of communications that take place during a typical session between a user and a web based application. The figure below describes the typical communication exchange in between all the composites in a typical web application system.

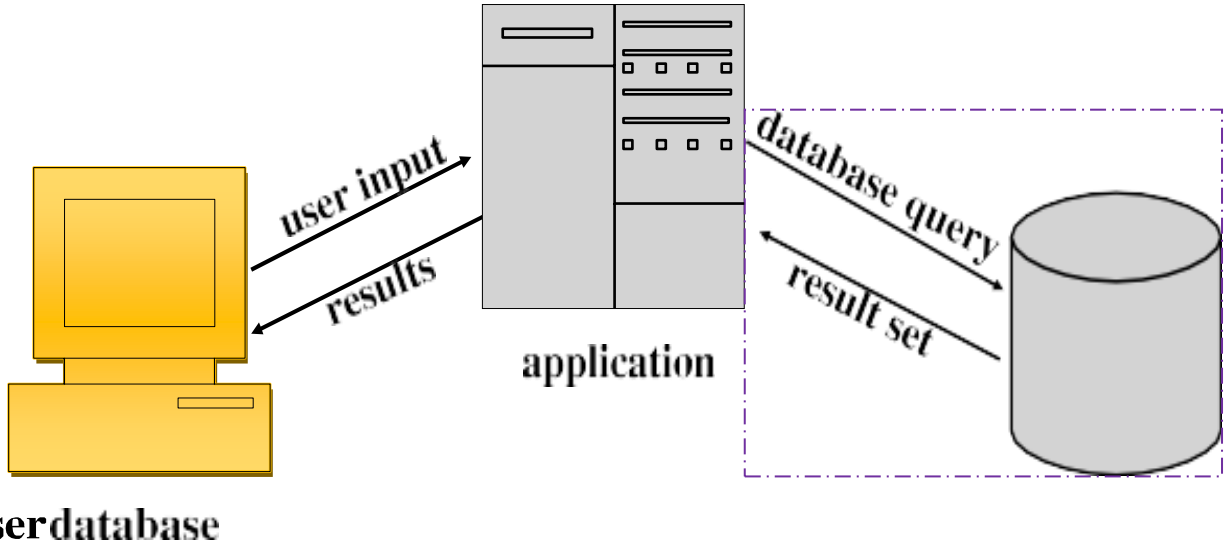

Figure: 3.1web application system

Web based applications have SQL injection susceptibilities as they do not disinfect the inputs they use to construct fabricated desired output. The gathered code is remains from an online storage system. The web-site provides user's input field to allow the user to keep their secret information, such as credit card/debit card password, etc. which user can use for future purchases conveniently.

Replace method used to escape the quotes so that any single quote characters in the input are considered to be literal and not string delimiters. Replace method is intended to block attacks by preventing an attacker from ending the string and adding SQL injection code. Althoughcard-type is a numbering column if an attacker passes 2 OR 1-1 as the card type, all account numbers in the database will be returned and displayed on thescreen.The SQL command injection attacks (SQLCIA) occurs when a malicious user, throughspecificallycraftedinput,causesawebapplicationtogenerateandsend a query that functions differently than the programmer intended.

For example, if a database contains user names and passwords,theapplicationmaycontaincodesuchasthefollowing:

```
Query="SELECT*FROMaccountsWHEREname='"
        +requestgetParameter("name")
        +"'ANDpassword='"
        +requestgetParameter("pass")+"'"
```

This code generates a query intended to be used to authenticate auser who tries to login to a web site. However, if a malicious userenters *"badguy"* into the name field and "'OR' a'='a" into thepasswordfield,thequerystringbecomes:

```
SELECT*FROMaccountsWHERE
        name='badguy'ANDpassword='OR'a'='a'
```

Thecondition always evaluates to true, and the user will bypasstheauthenticationlogic. Commandinjectionvulnerabilitiescontinuetobediscoveredonlarge, real-world web applications.

3.2 Research Tools

3.2.1 Hardware Tools

The hardware tools used for the research are listed in the table below.

Table 3.1 Hardware tools

Number	Device Name	Usage for the research
1	Computer	For method Processing
2	Flash disk	To transfer data
3	Hardisk	To store the datasets

3.2.2 Software Tools

A group of computer programs used to create, manage, debug, or support other applications and programs are referred to as software tools.

Python:is object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

MySQL:-is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web serverand MySQL as the relational database management system and PHP as the object-oriented scripting language.

Medley:- is most known for its Reference management software, citation management software. It is software for scholars and authors to use for recording and utilizing bibliographiccitations (references) as well as managing project references either as a company or an individual. It is used to manage and share research papers and generate bibliographiesfor scholarly articles.

3.3Research Methodology

SQL injection is widely used hacking technique, in which the intruder adds SQL statements using a web application as input fields to access to the secret users resources and due to lack of input Validation in web applications causes intruders to be successful in hacking.With above said technique, we can assume that a Web application receives "http//" request from a user client as Input and generates a SQL statement as output for the back-End database server. For example an administrator will be authenticated after providing input as:

Typing: employee id – 0112and password =admin, configure

That describes a login by a suspicious user exploiting SQL Injection vulnerability. Usually, it is structured in threephases,

1. An Intruder sends the malicious "http//" request to the Webapplication,
2. Generates the SQL statement,
3. Dedicatedly Deposited the SQL statement to the back end database.

Gathering data for SQL injection research is generally done in two primary ways, Capturing actual web traffic coming into an organization or the generation of realistic simulated traffic. Both approaches have their advantages and disadvantages. Real web traffic is of course the most realistic, but it can be difficult to determine which packets belong to an attack.

It can also be difficult to obtain this type of traffic, as organizations are typically reluctant to share web traffic due to privacy and security concerns. Another issue is that a simple research honey pot may capture mostly automated scans generated by common attack tools that would be more easily captured in a controlled lab setting. Simulated attacks have the advantage that they are controlled so that normal

and malicious traffic is easily distinguished for labeling purposes, and a good assortment of attacks can be included based on the latest techniques.

The datasets which will be used in the research are available online for research community. Damn Vulnerability Web Application (DVWA). They areused to get the datasets for the testing phase for all are labeled datasets to Machine learning.They areScikit-learn (Sk-learn), w3schools as well as Ensemble learning.

a. **Scikit-learn (Sk-learn):-**is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, and clustering also dimensionality reduction via a consistence interface in Python[11].
b. **Ensemble Model:-**is use to combine several models in order to improve the prediction accuracy in learning problems with a numerical target variable.The effectiveness of an ensemble can be measured by the extent to which the members are error–independent (show different patterns of generalization). The ideal would be a set of models where each of the models generalizes well, when they do make errors on new data, these errors are not shared with any other models.[15]

Ensemble learning is one of the most effective ways to build an efficient machine learning model. You can build an ensemble machine learning model using simple models and yet get great scores which are at par with the resource-hungry models like neural networks.

Simple:

1. Max Voting
2. Averaging
3. Weighted Averaging

Advanced

4. Stacking
5. Blending
6. Bagging
7. Boosting

Ensemble learning can improve the results of machine learning even exponentially at times. There are two major benefits of Ensemble models:

- More accurate predictions(closer to the actual value)

- Combining multiple simple models to make a strong model improves the stability of the overall machine learning model.

As you will learn in the course, we will take a real-life dataset and study how ensemble learning improves the score of our machine learning model as compared to using only simple models.

Voting and Averaging Based Ensemble Methods

Voting and averaging are two of the easiest ensemble methods. They are both easy to understand and

implement. Voting is used for classification and averaging is used for regression.

In both methods, the first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method. The following Python-sequel pseudocode shows the use of same training dataset with different algorithms.

```
train = load_csv("train.csv")
target = train["target"]
train = train.drop("target")
test = load_csv("test.csv")

algorithms = [logistic_regression, decision_tree_classification, ...] #for
classification
algorithms = [linear_regression, decision_tree_regressor,...] #for regression

predictions = matrix(row_length=len(target), column_length=len(algorithms))

for i,algorithm in enumerate(algorithms):
predictions[,i] = algorithm.fit(train, target).predict(test)
```

According to the above pseudocode, we created predictions for each model and saved them in a matrix called predictions where each column contains predictions from one model.

Majority Voting

Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. If none of the predictions get more than half of the votes, we may say that the ensemble method could not make a stable prediction for this instance. Although this is a widely used technique, you may try the most voted prediction (even if that is less than half of the votes) as the final prediction. In some articles, you may see this method being called “plurality voting”.

Weighted Voting

Unlike majority voting, where each model has the same rights, we can increase the importance of one or more models. In weighted voting you count the prediction of the better models multiple times. Finding a reasonable set of weights is up to you.

Simple Averaging

In simple averaging method, for every instance of test dataset, the average predictions are calculated. This method often reduces over fit and creates a smoother regression model. The following pseudo code shows this simple averaging method:

```
Final_predictions = []
Forrow_number inLen (predictions):
Final_predictions.append(
Mean(prediction[row_number,])
    )
```

Weighted Averaging

Weighted averaging is a slightly modified version of simple averaging, where the prediction of each model is multiplied by the weight and then their average is calculated. The following pseudo code shows the weighted averaging:

```
Weights = [...,...,...]#length is equal to Len(algorithms)
Final_predictions = []
Forrow_number in Len(predictions):
    Final_predictions.append(
Mean(prediction[row_ number,]*weights)
    )
```

3.4Methods to Evaluate Accuracy

Machine learning is used to determine the most accurate and efficient way to classify network traffic as an attack or as normal traffic to best protect digital assets while allowing normal business to precede Uninterrupted.All researches are evaluated their results in terms of performance characteristics. The popular techniques are:

3.4.1Decision Tree Algorithms: are a category of classification algorithms that create a predictive model based on the values of features in the dataset, at each point in the process choosing a feature that divides the dataset to maximize information. In other words it prefers predictive decisions that contain larger numbers of records.

3.4.2 Rules-based Algorithms: Rule-based algorithms are similar to decision tree algorithms in that they are iterating through features of the dataset and selecting those which best predict the class label. These rules are evaluated in terms of their coverage and accuracy. Coverage is the ratio of instances of the dataset that are covered by the rule; accuracy is the ratio of the instances that belong to the predicted class to the total instances covered by the rule.

3.4.3 Support Vector Machine (SVM) algorithms:are a classification technique based on statistical methods which work well with high dimensional dataset. SVM attempts to create a decision boundary between classes using instances of training data referred to as support vectors. This is done by finding a maximal hyper-plane separation between instances of the different classes.

3.4.4 Neural Network algorithms:is originally designed to simulate the human brain structure, with a system of linked nodes conceptually related to neurons. At the simplest level, there are input nodes that are exposed to features of the dataset and output nodes corresponding to the classification outcomes.

3.4.5 Ensemble Techniques:The purpose of ensemble techniques is "improving classification accuracy by aggregating the predictions of multiple classifiers".

Ensemble models are multiple supervised learning models used together to predict a value. In an ensemble model, individual supervised learning models are trained independently, and then the results obtained from each model are either averaged or voted to provide a single result. This obtained result obviously provides much better predictive accuracy than the individual models.

Machine Learning models are prone to various errors including Bias Error and Variance Error. Bias and Variance Errors can be defined as follows.

Bias Error: This error signifies how much the predicted value is different from the actual value.

VarianceError: Thiserrorsignifieshowmuchamodelsbehaviorwillchangeifwechange the training dataset. High Variance Error indicates overfitting issue. Understandably the outcome of a function will depend on the training data, since it was used to train the functioninthefirstplace.Hencesomeamountofvarianceisexpectedineverysupervised learning model. But it should not be the case that the model significantly changes if the training data is changed. Ensemble learning could also help with reducing theseerrors.

The red dot in the middle denotes the actual valuesandthebluedotsdenotethepredictedvaluesbythemodel. Itcanbeinferredfrom that in case of higher variance, the predicted values are farther apart from each other and in case of high bias the predicted values are farther apart from the actual values. Ensemble learning can be achieved by the following two ways that are discussed briefly in the nextsection.

1. Bagging
2. Boosting

Bagging is an ensemble learning approach that predicts a value of data by using multiple supervisedlearningmodels andthencombiningtheresultsofalltheseindividuallearning models by a chosen technique. The technique used for combining these results could be anyincludingbyweightingtheresults,takingtheiraverage,votingforthemaximumresult, etc. Bagging is also called as Bootstrap Aggregation. Bagging can help with reducing variance errors by using multiple supervised learning models and then combining their result.[16]An example of bagging technique is Random Forest Algorithm. In this approach, multiple decision trees are created on

random subsets of training data and results are collected from each decision tree. A final result is then selected from these results by taking an average of all the results from individual supervised learning models. Thearchitecturaldepictionofbaggingandboostingtechniques.Ascanbeobserved all the individual models are used in parallel in baggingapproach.

Boosting on the other hand is an Ensemble learning approach that also uses multiple supervised learning models in combination to provide better predictive results. The differenceisthewayinwhichboostingusesthesemultiplemodels.Insteadofusingthem in parallel, in boosting the multiple models are used sequentially. In this technique each predictormodellearnsandtriestominimizetheerrorsfromthepreviouspredictormodel. Boosting algorithms can reduce the bias errors introduced due to small size of datasets. An example of boosting algorithm is Gradient Boosting.

Gradient Boosting approach is used to classify and detect SQL Injection attacks. One of the important reasons of choosing this approach is because not enough data is available to train the machine learning models. Naïve Byes technique has been implemented to detect SQL Injection attacks because it can be trained even on small datasets.[18] However, that can lead to high bias errors as it is possible that data will not be classified correctly all the time. The hope is that using Gradient Boosting approach results in better accuracy while classifying the SQL Injection queries and overall provides better results and higher ratio of detecting an SQL Injection attack.

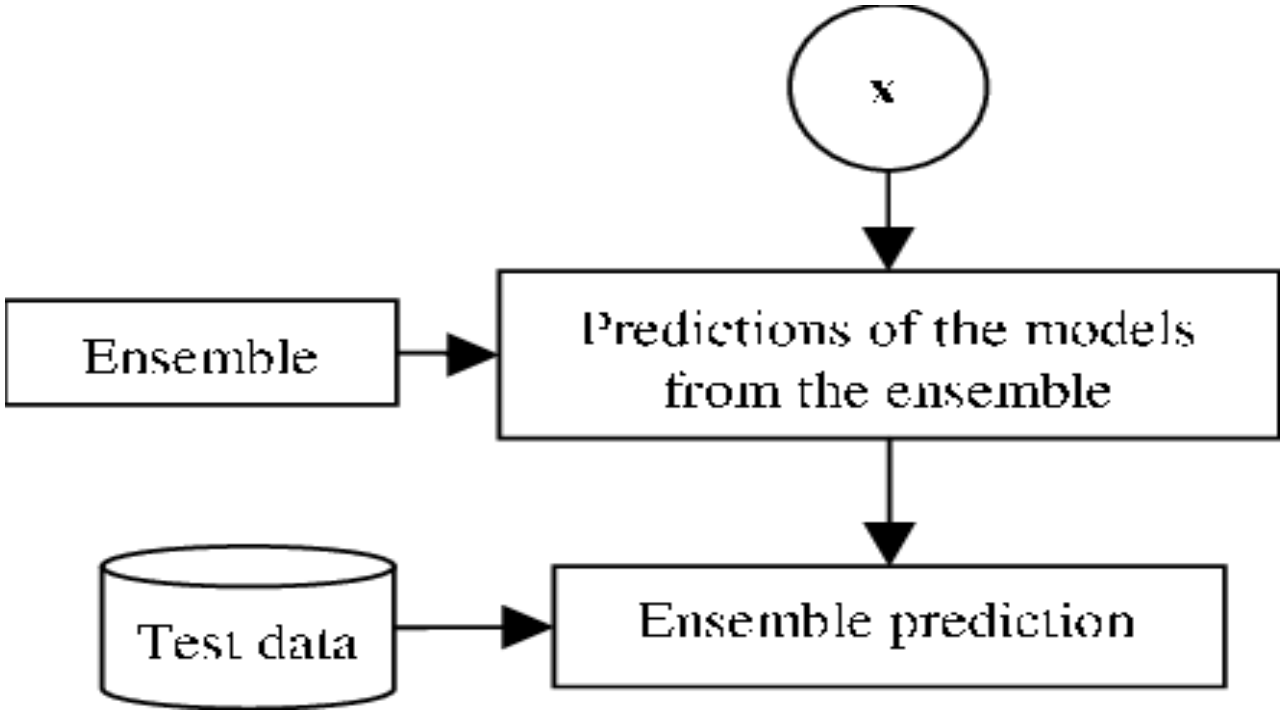

Figure3.2Ensemble Techniques

3.4.6 Feature selection: is used to reduce the dimensionality of data for performance reasons. This is a technique to remove redundant features, which are providing duplicate information of another feature, and irrelevant features, which are providing little or no useful information to help with classification.

Classification methods are probably the most popular of all machine learning classes of algorithms. Therefore, there is a constant interest in devising new techniques that can increase the accuracy of classification.

One idea is by using groups of classifiers instead of individual ones. The best known ensemble methods were introduced with bagging and boosting in which several classifiers were used to produce one single outcome with improved accuracy. However, the classic version only uses the majority or plurality vote to aggregate the outcomes of individual classifiers.

That is why research has continued with the study of other voting methods combined with ensemble methods and some results are presented as follows.

Several voting methods used for testing ensembles of classifiers trained by the bagging procedure are presented in. The classifiers are multi-layer perceptions. Two datasets are used: digits and capital letters, and both were equally divided for training and testing. These datasets were run on different MLP structures (small and large). The results showed that the best voting methods for small MLPs are the sum rule, the product rule and the Broad count. Also, Broad count provided good recognition results on large MLPs.

The performance of certain voting methods was also analyzed in, where bagging was used for the reconciliation model, which is a process of combining classification models. The chosen classifier was the classification tree, with the dataset divided into two-thirds for training and one-third for testing.

The experimental results show the generalization error performance of bagging using voting methods such as plurality, anti-plurality, broad count, plurality with elimination and Condorcet's pairwise comparison. Several configurations were used for the testing observations: various datasets with and without noise, different sizes of datasets and different number of classes.

The overall performances show the following order in classification accuracy of the used voting methods: Broad count, Condorcet's pairwise comparison, anti-plurality and plurality with elimination.

Three voting methods were used with bagging on the MLEM2 rule induction algorithm: support, strength and democratic. The chosen number of LERS (Learning from Examples based on Rough Sets) classifier instances (i.e. bootstrapped data) was 100. On every bootstrap set, the MLEM2 induction rule was used and the resulting rules were then provided as an input for the LERS classifiers.

The experiments were based on 16 datasets and the error rates were presented for each rule set used in bagging including the original prediction without aggregation. The results showed the stability of bagging in which the error rate became almost constant when the number of classifiers used reached a certain number.

The democratic voting presented the best overall performance while the voting based on strength had the worst performance.The result of bagging text classification is presented in, Binary decision trees were created using the C4.5 algorithm and the efficiency of bagging was compared with the standard classifier. The number of classifiers was limited to 200 and the tests were carried out on two datasets: the Reuter's collection and the Maritza collection.

The results showed the precision of bagging and the precision of standard decision tree classifier. For more frequent classes, the bagging method presented better results than the standard one, otherwise vice-versa. Also, bagging had a higher precision for a number of classifiers greater than 20, for the Reuter's collection, and 10, for the Maritzacollection.

The performance of ensemble methods was analyzed using 10% of the KDD CUP '99 dataset. Data instances have 41 attributes and the classification labels that must be determined are: "normal" for the records with normal behavior and "attack" for those that are considered to be malicious.

The malicious attacks can also be classified into three different categories: denial of service attack, user to root attack and remote to local attack. A comparison was made between bagging, Ad Boost and the standard C4.5 algorithm.

Bagging and C4.5 obtained similar performances, both with an error of 1.99%, while Ad Boost performed slightly better with an error of 1.95%.

A weighted voting ensemble learning classifier based on differential evolution (DEWV) was presented in [12]. Compared with bagging, DEWV did not use the same base classifier; instead it used randomly chosen base instances, for diversification purpose, from the following set of five classifiers: C4.5, Naïve Byes, Bayesian Nets, *k*-Nearest Neighbor (*KN*N) and Zero R. The voting system in DEWV was majority voting but what makes it different is the weighting of outputs from the base classifiers.

The differential evolution method was used to optimize the weights of each classifier: high accuracy of an instance would lead to a higher weight of that particular classifier and vice-versa. A comparison was made between bagging, Ad Boost, majority voting and DEWV using 15 datasets. In both Bagging and Ad Boost, the Naïve Byes classifier was used. In most cases DEWVobtained the highest performances, i.e. 13 out of 15 datasets.

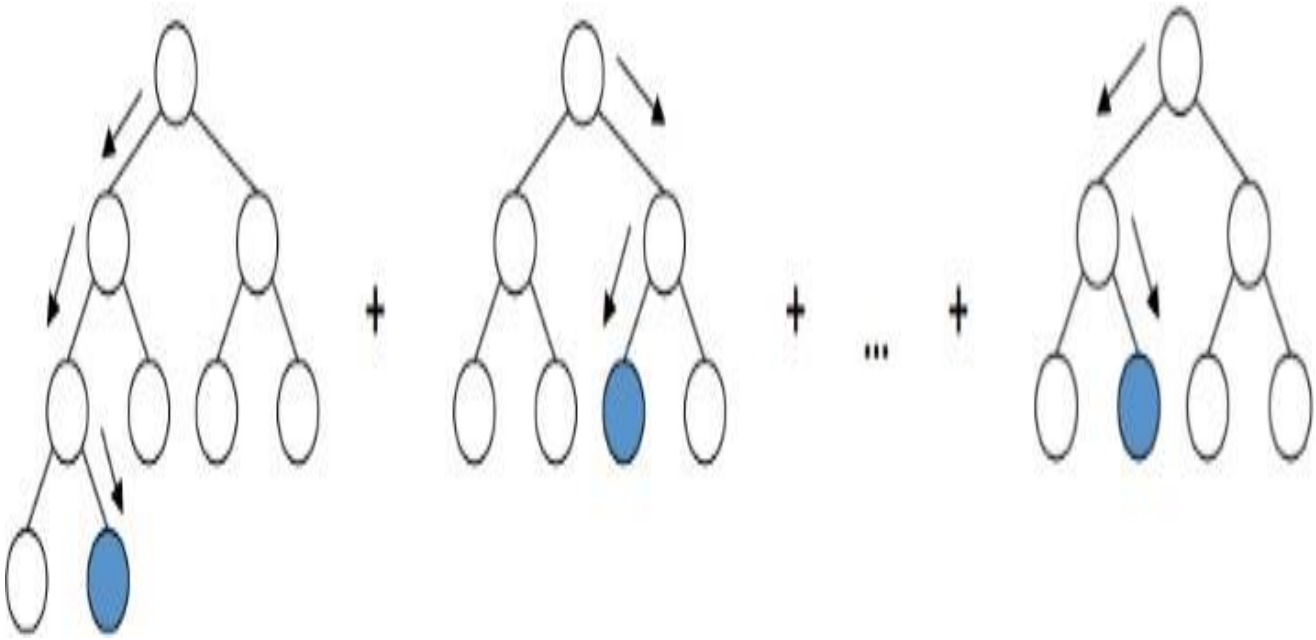

Figure 3.3Multiple Decision Tree Classifiers used for GradientBoosting

Gradient boosting is an Ensemble learning method to reduce errors and provide predictions with better accuracy. Gradient Boosting algorithm uses simple classifiers, mostly decision trees, in a sequential manner, to provide results.

The algorithm first uses the simple classifier to classify the data. Then the results are considered to calculate the errors or the data points that were not easily fit by the simple classifier. The algorithm then focuses on those data points in the next round and tries to fit them as well. In this way the errors are reduced, and outlier data points are also taken into consideration.

CHAPTERFOUR

SQLIAPREVENTION SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Process

The system design is done by using machine learning rule based techniques to classify incoming traffic as normal or malicious. The system consists of custom enterprise chat web application with a remote MySQL server backend. Data is captured in two places HTTP traffic between the traffic generation server and the web app server is captured. These two sets of data are then processed and correlated to create a separate dataset containing features from both datasets. The machine learning algorithms used are evaluated for classification accuracy as well as efficiency in terms of time to build models. The general process of the system is lock as below.

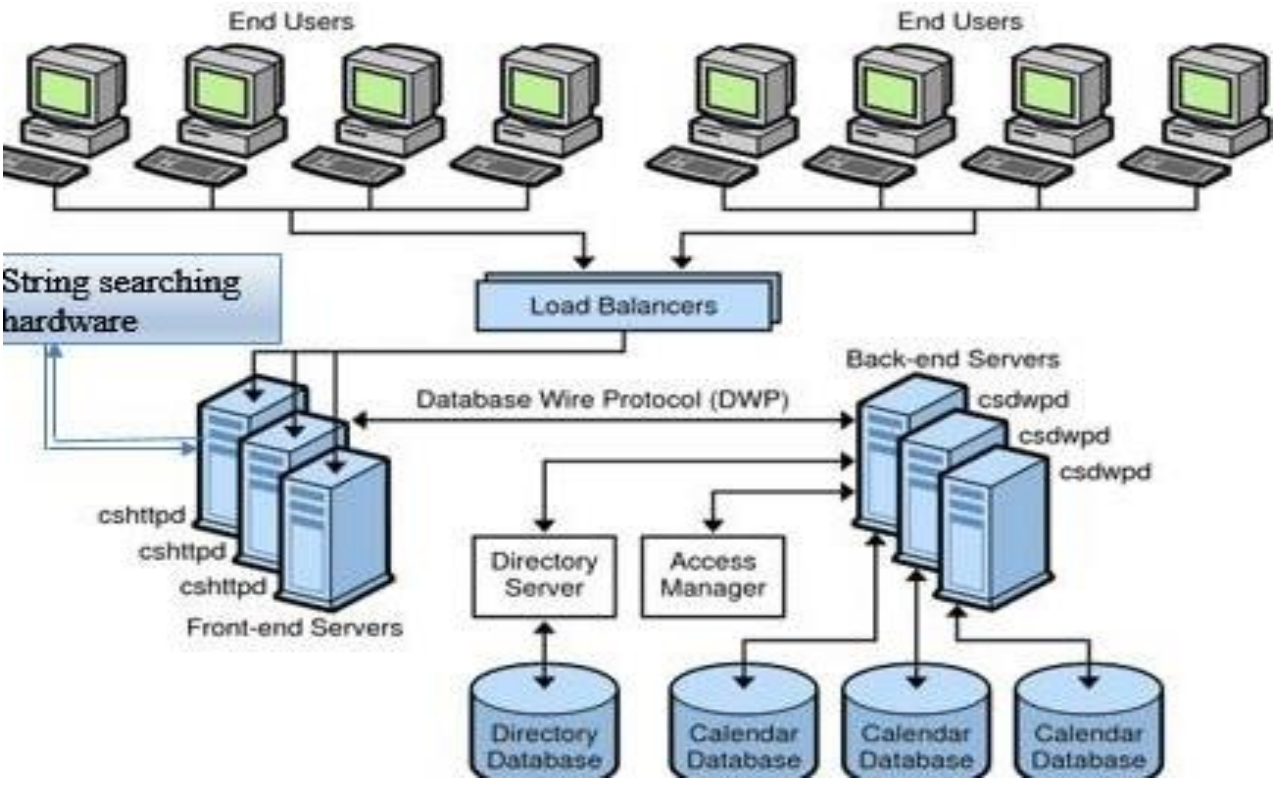

Figure 4 1 System Process

4.2 Architecture andData Processing

The architecture we're using consists of server nodes,: a webapp server, a traffic generation server, a database server, and a MySQL data capture node. The data generation process we're using consists of three phases: traffic generation, capture, and pre- processing. These are briefly discussed below.

Traffic Generation:- The normal and malicious traffic is generated from the scripts located on the traffic generation server. This traffic consists of HTTP POST requests from this server to the chat webapp, which then generates MySQL traffic between the webapp server and the database server. Normal traffic consists of simulated normal interaction with the chat web application. Malicious traffic differs in the inclusion of manually coded SQL injection attacks. The text is randomly generated to be reasonably realistic in terms of commonality of words and sentence structure in order to simulate statistically realistic web-based communication.

Traffic Capture:-The traffic is captured at the webapp server. The traffic reports the result from the interaction between the webapp and the remote MySQL server.

Data Pre-processing:- The normal and malicious datasets are processed into one file with shell scripts to create the correlated dataset. Correlation is done with the insertion of a token in the form of a unique text string that can be compared and then removed from the dataset once events are correlated.

4.3 Web Application

The app uses a remote MySQL database backend to store the simulated chat messages. A web application takes text as input from users to retrieve information from a database. Some web applications assume that the input is legitimate and use it to build SQL queries to access a database. Since these web applications do not validate user queries before submitting them to retrieve data.

They become more susceptible to SQL injection attacks. Web application accepted malicious query may break the security policies of the underlying database architecture. The result of the query might cause the database to malfunction and release sensitive information.The webapp to be vulnerable to SQL injection and the specific vulnerability occurs in an UNION statement.The first query returns exactly what should have been returned and the second query is the one where malicious code had been used.

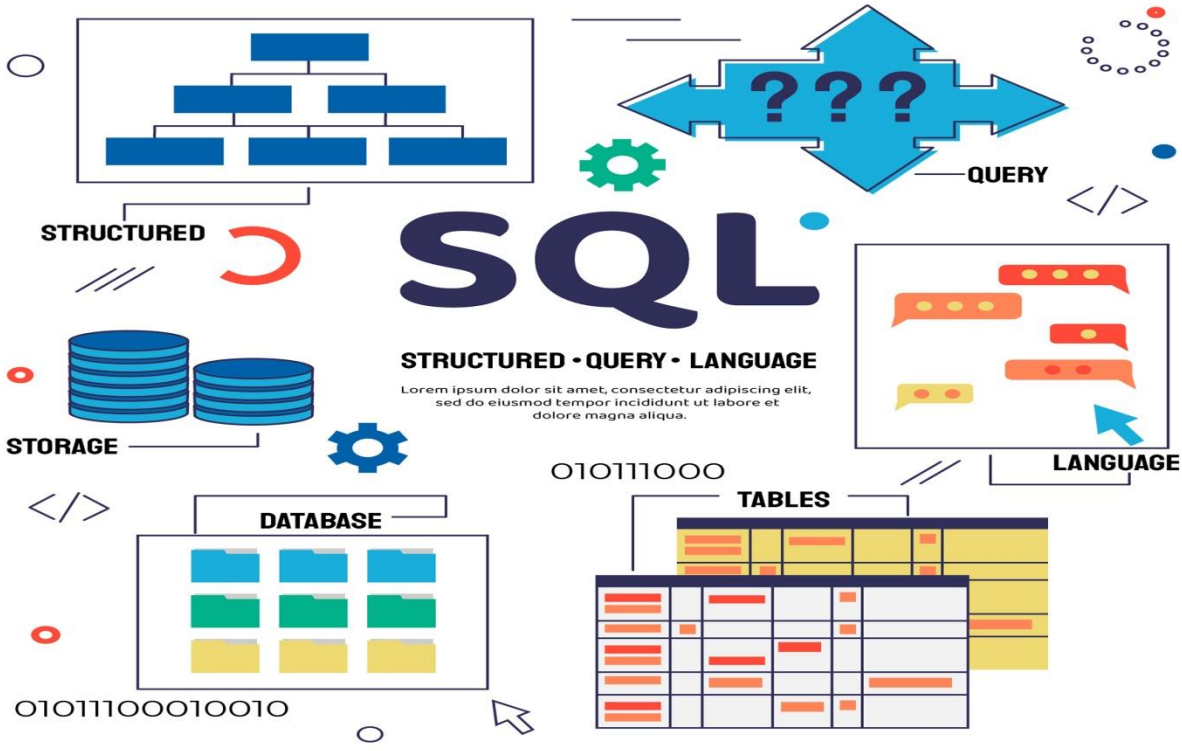

Figure 4.2 Structured Query languages

4.4 SQL Attacks

The SQL attacks are manually coded SQL injection attacks of different types. Like: timing-based attacks, error-based attacks and general injection of SQL functions.

4.4.1 Timing-based attacks are typically used when there is no immediate display of the results of injection attempts. These attacks, which are also often called blind SQL injection, take the general form:

CASE (SUBSTRING(version(),1,1)) when 4 then sleep(5) else sleep (0)...

Successfully injecting code like this has the effect of introducing a five second delay in the processing of the request.The architecture the MySQL version is actually 5.5,

This query will pause five seconds before returning and thus give attacker information about the database backend of the web application.

4.4.2 Error-based attacks an example of error-based attacks that using is the injection of the following code:

... or update exml(1,concat(0x7e,(user())),0) or ...

This code when injected into an SQL statement results in an XPATH error from the database which potentially contains information useful to an attacker, in this case the username under which the webapp is accessing the database.

4.4.3 UNION statement attacksby breaking out of the statement with a single quote character and then injecting the function and proper characters to complete the query, and then union characters to comment out the rest of the query. An example of this type of attack is the injection of the following code:

',(union database()),now()); --

When injected into an UNION statement, this code will cause the database name to be added to the actual input field.

4.5 Machine Learning

Machine learning is used to determine the most accurate and efficient way to classify network traffic as an attack or as normal traffic to best protect digital assets while allowing normal business to proceed uninterrupted.The analysis of the processed data is done with Rule-Based Machine Learning. The rule Based Machine Learning usesto reduce the number of features to allow for efficient machine learning.[19] This algorithm selects a subset of features that are highly correlated with the class attribute the classes of normal or malicious.The evaluating algorithms for accuracy and performance in terms of the classification model are listed above.

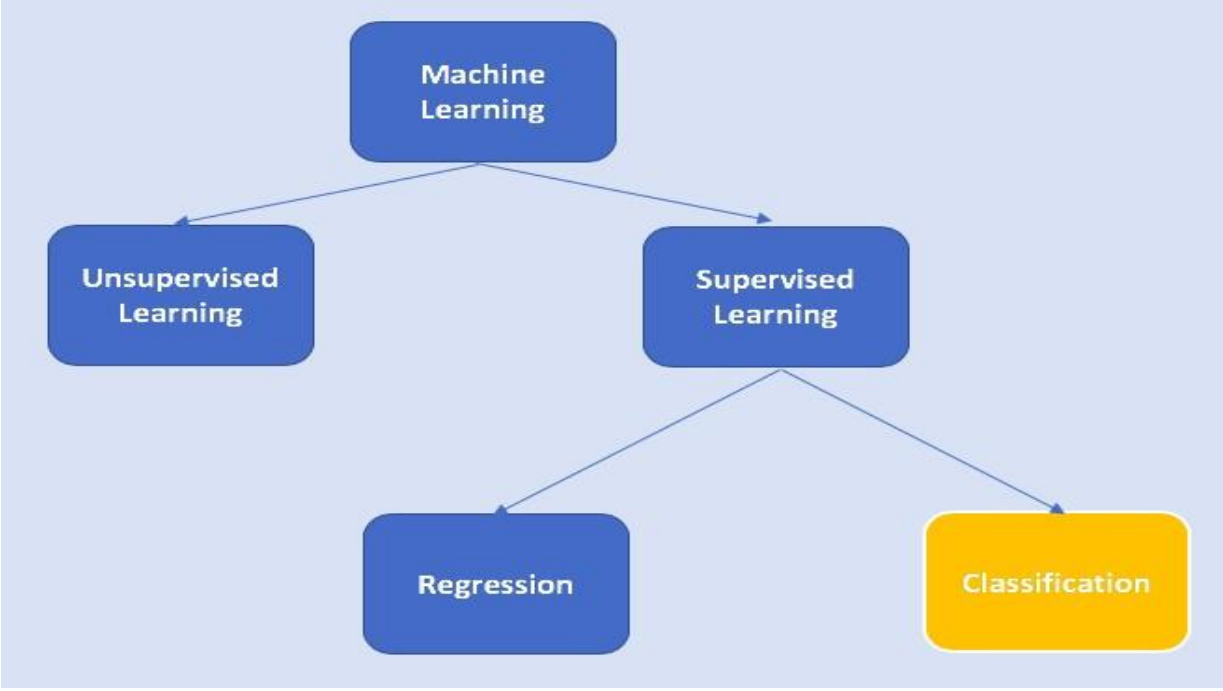

Figure 4.3 Supervised classifications

The supervised learning model basically learns the relationship between the data and the label and then uses this learnt information to classify new data that it has never seen before. This new data is called as the test dataset. We use test dataset to determine the accuracy of a supervised learning algorithm. This is how we predict the values or classify never before seen data using supervised machine learning. Supervised learning algorithms can further be broadly classified as Regression algorithms and Classification algorithms.

CHAPTERFIVE

RESULTS OF PREVENTION SQLIA

5.1 Entries

The datasets usedthe normal and malicious traffic originating from the traffic generation server collected and correlated. For our results the dataset consists of manyentries. The entries are divided into normal traffic and malicious traffic. The many datasets are located by the:-

Web Application dataset:-This dataset as mentioned was captured inbound to the webapp and as such some of the unique features include information about the TCP and HTTP packets.

Correlated dataset:-This dataset is the combines features from the both normal and malicious. The machine learning algorithms are able to use features from both of these datasets.

The problem of making a good choice out of multiple possible solutions or opinions has leaded us to a common agreement called voting. By voting, each voter has the right to express his/her preference for one or more candidates. Evaluating the votes, one can reach to a final decision which is usually in favor of the majority opinion, i.e. the winner will be the most preferred candidate. By analogy, choosing the solution of a problem from a set of several solutions, one can obtain a better solution or even worse if the majority solutions are worse. Our problem is that of classification.

Currently, there are many proposed techniques to construct voting systems: parametric, nonparametric, heuristic, logical, probabilistic methods, etc. For example, one approach of using voting systems with different learning algorithms consists in choosing different classifiers such as decision trees, KNN, multilayer perception, where the same training dataset is provided for each of them. Each classifier will obtain different predictions. To get the best prediction, one can use a voting system in which the winning class will be the majority. Using ensembles of different classifiers has the advantage that not all of them will make the same mistake.

Another method is to use a single type of classifier, but on several training datasets. The approach used in this paper is the one in which we have a single classifier and several training datasets, i.e. bagging. Bagging, one can train each of the same classifiers on various feature samples. In particular, this idea is used in the Rule based algorithm, where each of the constructed decision trees is combined by majority voting and trained not only on different samples, but also on randomly chosenfeatures.

Although the plurality method is the easiest and generally gives good results, these experiments show that also the single transferable vote method can sometimes be a better choice. The main difficulty lies in the computation of the preference ordering, which is time-consuming compared to the simple vote

count. Some methods can be naturally used to express such an ordering, while in other cases this is not straightforward.

The single transferable votemethod can be a good alternative to the plurality voting method that is generally used with ensemble- based classification.

A natural direction for future investigation is the use of other classification algorithms and other classification problems, in order to assess more clearly the influence of different voting methods on bagging. Also, since the number of bootstrapped training sets is small, it is important to make a thorough statistical analysis regarding the performance of the voting methods.

Classifier ensemble was proposed to improve the classification performance of a single classifier. The classifiers trained and tested in Phase 1 are used in this phase to determine the ensemble design. Also, this phase is divided into two sub-phases, that is, Phase 3a and Phase 3b.

Simple majority voting is used to ensemble the classifiers in determining detection accuracy. This is an iterative phase in which a threshold (acceptable detection accuracy set) is set and checked with the evaluation results until an optimum result is achieved.

Equation: formula for calculating detection accuracy.

$$\text{Detection Accuracy Result} = (a \times d1) + (b \times d2) + (c \times d3)$$

Where $a + b + c = 1$ and a, b, and c are variables in the range of [0,1]

Phase 3a is divided into two parts, namely design and decision. In the design part, four algorithms are being considered for ensemble and a committee of three algorithms is used to form an ensemble since majority voting requires an odd number of participants.

On the basis of the output of Phase 2, all the individual algorithms will be evaluated with the same metrics used in Phase 2 and then voted on. The decision part of Phase 3a rely on the output of the design part to decide which of the ensemble is the best performed which is then passed to Phase 3b for comparison with the best of the four algorithms evaluated in Phase 2.

Vote provides a baseline method for combining classifiers. The default scheme is to average their probability estimates or numeric predictions, for classification and regression, respectively. Other combination schemes are available for example, using majority voting for classification. Multi-Scheme selects the best classifier from a set of candidates using cross-validation of percentage accuracy or mean-squared error for classification and regression, respectively. The number of folds is a parameter. Performance on training data can be used.

Majority voting, often times the number of algorithms used is four and a decision is taken to remove the least-performed classifier but in the case of this project, the last two algorithms performed almost the same and as such the chance of removing one of them without being biased is uncertain. Because of this reason, each algorithm is marched with other algorithm in a committee of three which leads us to having four ensembles. Therefore, the problem of selecting one of two closely performed classifiers has been resolved.

Ensemble	Algl	Alg2	Alg3
Ensemble 1	RB	C4.5	LR
Ensemble 2	RB	C4.5	KNN
Ensemble 3	KNN	LR	RB

Table 5.1Ensemble Components

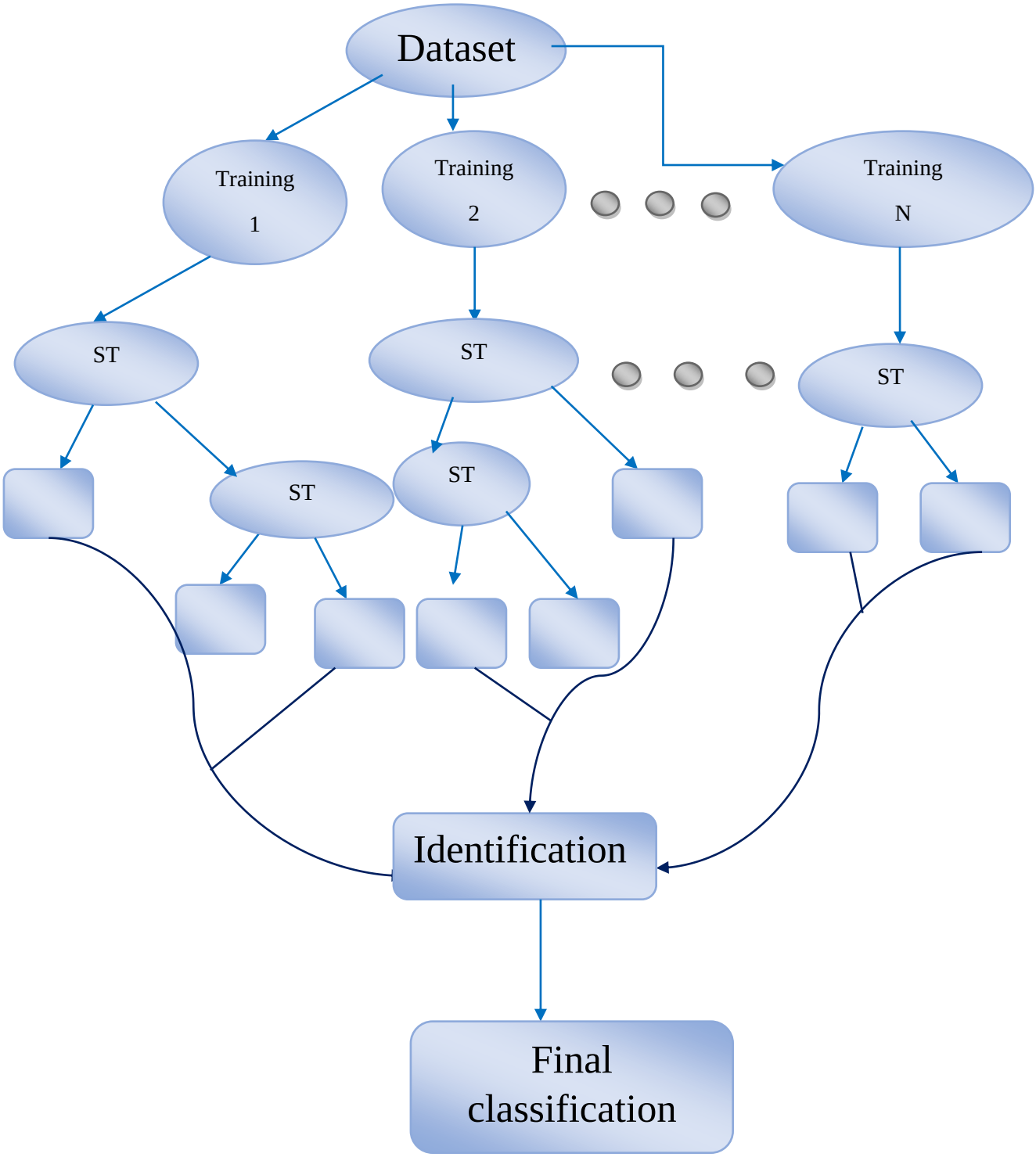

Figure 5.1 Different Ensemble Classification

5.2 Results

The results of Ruled-Based and others machine learning algorithms are summarized with the classification accuracy achieved and effective prediction.

Dataset	Algorithm	True Pos.	False Pos.	True Neg.	False Neg.	Accuracy
Webapp	Joint Reserve Intelligence Program	9117	169	9831	883	94.740%
	J48	9385	259	9741	615	95.630%
	Random Forest	9487	182	9818	513	96.525%
	Support Vector Machines	9134	329	9671	866	94.025%
	Artificial Neural Network	9501	158	9842	499	96.715%
Correlated	Joint Reserve Intelligence Program	9623	193	9807	377	97.150%
	J48	9656	197	9803	344	97.295%
	Random Forest	9860	32	9986	140	98.055%
	Support Vector Machines	9457	314	9686	543	95.715%
	Artificial Neural Network	9644	121	9879	356	97.615%

Table 5.2 Record Data

Machine Learning Techniques for SQL injection Solutions

Techniques	Classifier(s)	Performance	Dataset
HIPS	Bayesian Naif Bayesian Multinomial	Accuracy= 97.6%	SQL injections data collection framework
SQLI-IDS	Backpropagation Neural Network	Overall accuracy= 96.8%	Dataset 13,000 URL addresses including 500 benign URLs and 12,500 malicious URLs
Sheykhkanloo et al.	Neural Network Based Model	Accuracy= 95%	Dataset 25000 URL addresses including 12250 benign URLs and 12250 malicious URLs
Verbruggen et al.	Decision Tree, SVM, Random Tree, Jribber, Neural Network, Random Forest	Recognition Rate= 98.6% for Neural Network	6 different datasets contains 1876 malicious packets and 11444 normal packets.
Wang et al.	Stacked AutoEncoder	Precision =100%	0.3 million TCP flow data collected from internal network
Ingre e al.	Decision Tree	Accuracy= 83.7%	NSL-KDD
Moosa et al	Neural network Multi-layer Feed Forward	Accuracy= 66.67%	300 SQL injection signatures and around 200 XSS signatures collected from different websites
Joshi et al.	Naive Byes	Accuracy= 93.9%	178 Codes including 101 normal codes and 77 malicious codes
SQLI GOT	Support Vector Machines	Accuracy= 96.23%	4610 injected sequences and 4884 genuine sequences
Modsecurity with ML	K-NN (K=3), SVM, Random Forest	Precision=97%	CSIC-2010 , DRUPAL And PKDD2007
TbD-NNbR	Neural Network	Accuracy= 99.23%	1655 queries tested, 451 malicious queries and 1204 legal queries
J48	Decision Tree Algorithm	Accuracy=97.15%	2000 queries tested, 1000 malicious queries and 1000 legal queries
JRip	Rule-Based Algorithm	Accuracy=97.295%	100 Codes including 70 normal codes and 30malicious codes

Table 5.3Classification Accuracy

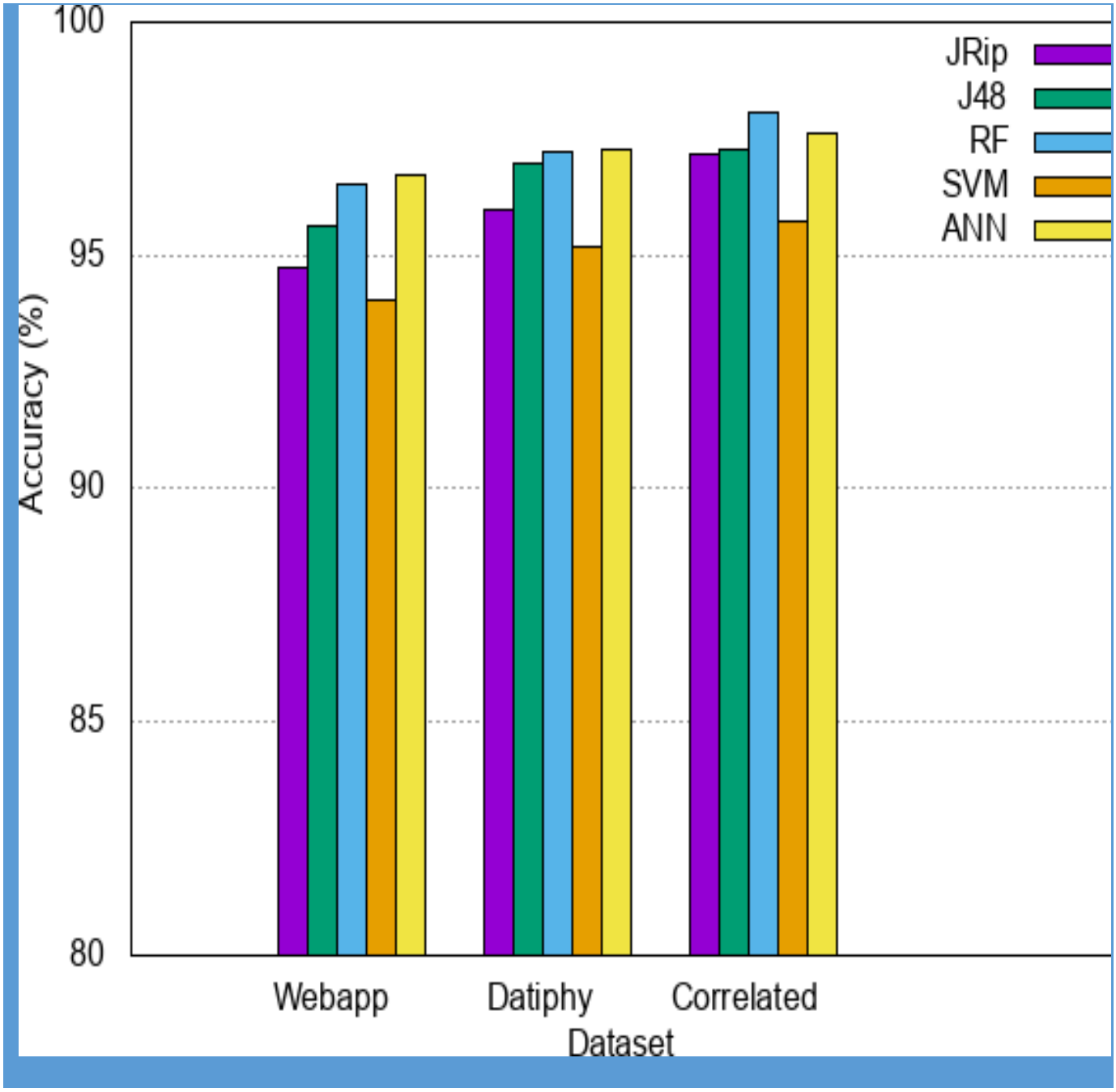

Figure5.2 Classification Accuracy

Accuracy is the classification accuracy achieved with the various algorithms, Model Time is the processing time to build the machine learning models, and Testing Time is the time to classify the testing dataset.

CHAPTER SIX

SQLIAPREVENTION ANALYSIS

Thecombination of two datasets result would be better in terms of classification accuracy. Several of the highest ranking rules created from the JRip algorithm are incorporating both the used records datasets and experimented with the JRip and J48 algorithms. Those results were similar in that there is consistently getting much better results with the correlated datasets than with either of the individual datasets alone.

There is also a consistent improvement in results across all metrics with larger datasets. Although the randomization, the data is apparently falling into categories that the machine learning algorithms can create a model from and that capability improves as the algorithms are exposed to more data.

One aspect of the performance results to note is the good relative performance of the ruled-based algorithm experiments. Ruled-based algorithmis often considered to be resource intensive. When the data is generally easy to classify based on a few well-selected features, then ruled-based algorithms perform quite well relative to other classification algorithms.

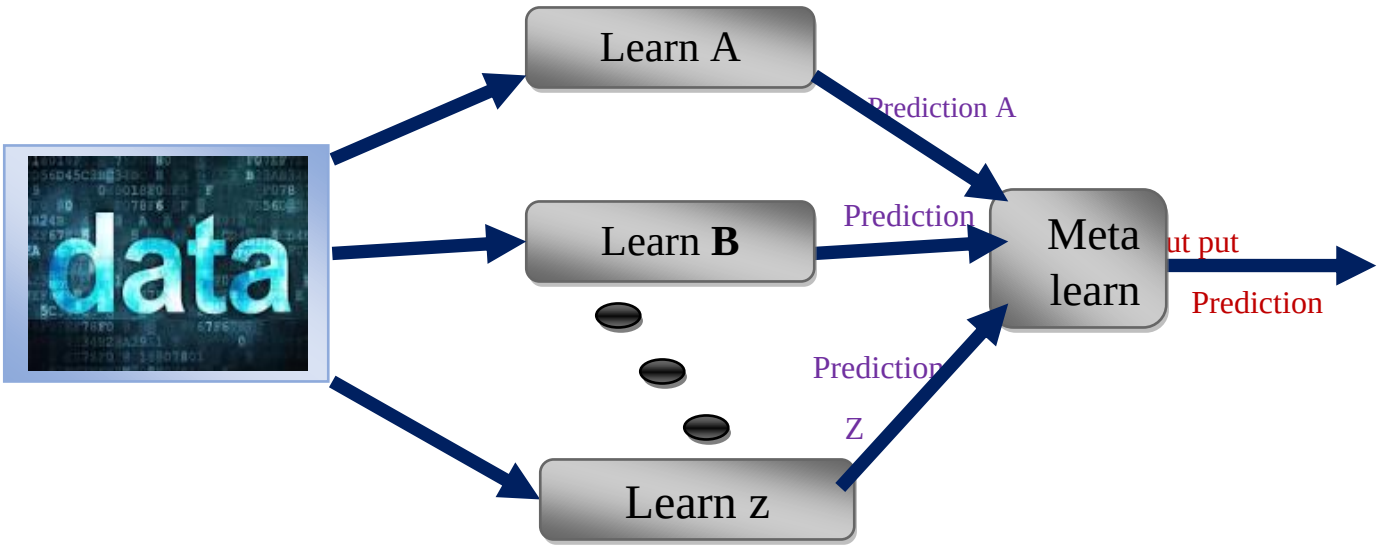

Figure 6.1 Ensemble learning methodfor Final Prediction

Ensemble learning algorithmscombines the decision of different regression or classification algorithms. The component models are trained on the entire training dataset. After these component models are trained, a meta-model is assembled from the different models and then it's trained on the outputs of the component models.

This approach typically creates a heterogeneous ensemble because the component models are usually different algorithms.

Based on this, the datasets results obtained with the correlated dataset using the faster algorithms are nearly as accurate as our results with the ruled-based algorithm. One implication of these experiments is that gathering data from multiple sources can potentially improve classification results as much as or more than the choice of potentially more accurate algorithms.

Converting Weak Models to Strong Ones

The term “boosting” is used to describe a family of algorithms which are able to convert weak models to strong models. The model is weak if it has a substantial error rate, but the performance is not random resulting in an error rate of 0.5 for binary classification.

Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so can’t use parallel operations here.

The general procedure of the boosting algorithm is defined as follows:

```
defadjust_dataset(_train, errors):
#create a new dataset by using the hardest instances
ix = get_highest_errors_index(train)
Returnconcat(_train[ix], random_select(train))

Models=[]
_train = random_select(train)
For i in range(n): #n rounds
Model = base_algorithm.fit(_train)
Predictions= model.Predict_train)
    Models.append(model)
    Errors = calculate_error(predictions)
    _train = adjust_dataset(_train, errors)
Final_predictions = combine(models, test)
```

The adjust dataset function returns a new dataset containing the hardest instances, which can then be used to force the base algorithm to learn from.

Ad boost is a widely known algorithm which is a boosting method. The founders of Ad boost won the Gödel Prize for their work. Mostly, decision tree algorithm is preferred as a base algorithm for Ad boost and in sk-learn library the default base algorithm for Ad boost decision tree.

As we discussed in the previous paragraph, the same incremental method applies for Ad boost. Information gathered at each step of the Ad Boost algorithm about the 'hardness' of each training sample is fed into the model. The 'adjusting dataset' step is different from the one described above and the 'combining models' step is calculated by using weighted voting.

Ensemble methods can help you win machine learning competitions by devising sophisticated algorithms and producing results with high accuracy, it is often not preferred in the industries where interpretability is more important. Nonetheless, the effectiveness of these methods is undeniable, and their benefits in appropriate applications can be tremendous. In fields such as healthcare, even the smallest amount of improvement in the accuracy of machine learning algorithms can be something truly valuable.

Bagging is a popular method used to increase the accuracy of classification, by training a set of classifiers on slightly different datasets and aggregating their output by voting. Usually, the majority voting is used for this purpose, or the plurality voting, when the problem has multiple class values.

ManySystems analyze the influence of several voting methods on the performance of two classification algorithms used for datasets with different levels of difficulty. The results reveal that the single transferable vote can be a good alternative to plurality voting, although it has the drawback of a higher computational cost related to the calculation of preferenceordering.

One idea is by using groups of classifiers instead of individual ones. The best known ensemble methods were introduced with bagging and boosting in which several classifiers were used to produce one single outcome with improved accuracy. However, the classic version only uses the majority or plurality vote to aggregate the outcomes of individual classifiers. That is why research has continued with the study of other voting methods combined with ensemble methods

Although the plurality method is the easiest and generally gives good results, these experiments show that also the single transferable vote method can sometimes be a better choice. The main difficulty lies in the computation of the preference ordering, which is time-consuming compared to the simple vote count. Some methods can be naturally used to express such an ordering, while in other cases this is not straightforward.

The single transferable votemethod can be a good alternative to the pluralityvoting method that is generally used with ensemble- based classification.

A natural direction for future investigation is the use of other classification algorithms and other classification problems, in order to assess more clearly the influence of different voting methods on bagging. Also, since the number of bootstrapped training sets is small, it is important to make a thorough statistical analysis regarding the performance of the voting methods.

CHAPTER SEVEN

CONCLUSIONS & FUTURE WORK

The characteristics of SQL injection and the structure of web-based systems are listed at above. SQL injection attacks and web-based attacks are a major issue in the security of financial, health, and other critical data and this problem only increases in importance as more societal processes become more dependent on the internet. The data analysis system is used for increased accuracy in detection of SQL injection attacks and established that the algorithms with experimented such as rule-based and decision tree algorithms. Neural Networks are much better in terms of time necessary to build models and execution time when classifying testing data.

Future works include collection of additional data such as traffic outbound from the web application to the browser, collection of larger datasets. By adding software searching algorithmthat has a better performance and throughput in comparison with the current implementation and also to see if this improves performance analysis of additional machine learning techniques for both accuracy and performance and adapting this system to detect other types of web-based attacks.

References

1 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, doi: 10.1109/ACCESS.2019.2947527.

2 D. A. Kindy and A. K. Pathan, "A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques," 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), 2011, pp. 468-471, doi: 10.1109/ISCE.2011.5973873.

3 William, G.J., Fond, H., Orso, A Projecting applications using positive tainting and syntax Member. IEEE Computer Society 34(I) (January - February 2008)

4 Wang Shihui and Zhang Xiyun. Research of cause, detection and defense measures of SQL injection. Journal of Hubei University (Natural Sciences). 2010, 32 (3): 269–272.

5 Ross, Kevin, "SQL Injection Detection Using Machine Learning Techniques and Multiple Data Sources" (2018). Master's Projects. 650. DOI: https://doi.org/10.31979/etd.zknb-4z36

6 Qi Mingxingchen. White Paper of Defense Technology for SQL Injection and XSS Attack, 2010: 3–9.

7 Wang Zhihu. Research of SQL injection attack and prevention measures. Coal Technology, 2011, 30 (1): 95–97.

8 ang Yun, GuoWaiping and Chen Chenghuan. Research on the SQL injection problem and defense methods in Web projects. Computer Engineering and Design, 2010, 31 (5): 976–978.

9 Justin Clarke. SQL Injection Attacks and Defense [M]. Elsevier. 2012.

10 Young-Su Jang, Jin-Young Choi. Detecting SQL injection attacks using query result size. Computers&Security, 2014, 44:104–118.

11 Stephen Thomas, Laurie Williams, Tao Xie. On automated prepared statement generation to remove SQL injection vulnerabilities. Information and Software Technology. 2009, 51:589–598

12 IndraniBalasundaram, E. Ramaraj. An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching. Procedia Engineering, 2012, 30:183–190.

[13] Witt Yi Win, Hnin Hnin Htun. A Simple and Efficient Framework for Detection of SQL Injection Attack. International Journal of Computer & Communication Engineering Research (IJCCER). 2013, 1(2):26–29.

[14] ZHANG Yong, LI Li, XUE Qian. Detection and Defense against SQL Injection Attacks. 2004, 15:103–105,108.

[15] M. Qbea'h, M. Alshraideh, and K. E. Sabri, ''Detecting and preventing SQL injection attacks: A formal approach,'' in Proc. Cybersecur. Cyber-forensics Conf. (CCC), Aug. 2016, pp. 123–129

[16] S. O. Uwagbole, W. J. Buchanan, and L. Fan, ''Applied machine learning predictive analytics to SQL injection attack detection and prevention,'' in Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM), May 2017, pp. 1087–1090

[17] N. Singh, M. Dayal, R. S. Raw, and S. Kumar, ''SQL injection: Types, methodology, attack queries and prevention,'' in Proc. 3rd Int. Conf. Com-put. Sustain. Global Develop. (INDIACom), Mar. 2016, pp. 2872–2876

[18] Mishra, Sonali, "SQL Injection Detection Using Machine Learning" (2019). Master's Projects727.

[19] Muhammad AmirulluqmanAzman Machine Learning-Based Technique to Detect SQL InjectionAttack Journal of Computer Science 2021, 17 (3): 296.303 DOI: 10.3844/jcssp.2021.296.303

[20] Chen Xiaobing, Zhang Hanyu, Luo Liming and Huang He. Research of SQL injection attack and technology of defense and detection. Computer Engineering and Applications, 2007, 43 (11): 150–152.

[21] Jemal, Ines &Cheikhrouhou, Omar &Hamam, Habib&Mahfoudhi, Adel. (2020). SQL Injection Attack Detection and Prevention Techniques Using Machine Learning. International Journal of Applied Engineering Research. 569-580.

[22] Anup Kumar, SandeepRai, Rajesh Boghey. 2021. A Novel Approach for SQL Injection Avoidance Using Two-Level Restricted Application Prevention (TRAP) Technique. International Conference on Innovative Computing and Communications, 227-238

2 HACKING & TRICKS, a blog about Hacking &Computer Security by Nirav Desai, in
3 22 an entry from January 8th, 2013
2 International Journal of Applied Engineering Research ISSN 0973-4562 Volume 15,
4 Number 6 (2020) pp. 569-580 ©Research India Publications. http://www.ripublication.com
2 Automated Fix Generator for SQL Injection AttacksFred Dysart and Mark
5 SherriffDepartment of Computer Science, University of Virginia, Charlottesville, VA 22903
2 K.Ross,M.Moh,T-S.Moh,J.Yao,Poster:Multi-
6 SourceDataAnalysisForSQLInjectionDetection,*38thIEEESymposiumonSecurityandPrivacy(IEEES&P)*,SanJo-se,CA, 2017.

2 R. Tiwari, A. Jain, 2012. "Improving network security and design using honeypots."
7 *In Proceedings of the CUBE International Information Technology Conference (CUBE '12)*. ACM, New York, NY, USA, 847-852.
2 B. Hanmanthu, B. R. Ram, P. Niranjan, "SQL Injection Attack prevention based on
8 decision tree classification," *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO),* Coimbatore, 2015, pp. 1-5.
2 J. Choi, H. Kim, C. Choi, P. Kim, "Efficient Malicious Code Detection Using N-Gram
9 Analysis and SVM," *2011 14th International Conference on Network-Based Information Systems*, Tirana, 2011, pp. 618-621.
3
0 S. Djanali, F.X. Arunanto, B.A. Pratomo, H. Studiawan, S.G. Nugraha, "SQL injection detection and prevention system with raspberryPi honeypot cluster for trapping attacker,"*in Technology Management and Emerging Technologies (ISTMET), 2014 International Symposium on*, vol., no., pp.163-166, 27-29 May 2014
3 D. Kar, S. Panigrahi, "Prevention of SQL Injection attack using query transformation
1 and hashing," *in Advance Computing Conference (IACC), 2013 IEEE 3rd International*, vol., no., pp.1317-1323, 22-23 Feb. 2013
3 D.Kar,A.K.Sahoo,K. Agarwal,S.Panigrahi,
2 M.Das,"LearningtodetectSQLIAusingnodecentrality with feature selection," *2016*

International Conference onComputing, Analytics and Security Trends (CAST), Pune, 2016, pp. 18-23.

[33] I. Lee, S. Jeong, S. Yeo, J. Moon, "A novel method for SQL injection attack detection based on removing SQL query attribute values," Mathematical and Computer Modelling, vol. 55, no. 1, 2012, pp. 56-68.

[34] M. Mueter, F. Freiling, T. Holz,J. Matthews, (2008). "A generic toolkit for converting web applications into high-interaction honeypots," University of Mannheim, 280.

[35] G.K. Sadasivam, C. Hota, "Scalable Honeypot Architecture for Identifying Malicious Network Activities," *2015 International Conference on Emerging Information Technology and Engineering Solutions,* Pune, 2015, pp. 27-31.

[36] Ke Wei, M. Muthuprasanna, Suraj Kothari, "Preventing SQL Injection Attacks in Stored Procedures" Proceedings of the 2006 Australian Software Engineering Conference (ASWEC'06IEEE).

[37] P. Bisht, P. Madhusudan, and V. N. Venkatakrishnan. CANDID: Dynamic Candidate EvaluationsforAutomaticPreventionofSQLInjectionAttacks.ACMTrans.Inf.Syst. Secur., 13(2): 1-39,2010.

[38] YongJoon Park, JaeChul Park, "Web Application Intrusion Detection System for InputValidationAttack"Third2008InternationalConferenceonConvergenceAnd Hybrid InformationTechnology.

[39] Needleman, S.B., Wunsch, C.D. "A general method applicable to the search for similarities in the amino acid sequence of two proteins", .T.MoI.BioI.48:443-453, 1970.

[40] Nausheen, K.: Detection and Prevention of SQL Injection Attacks by Request Receiver, Analyzer and Test Model.2011.

[41] Shikhar Jain &Alwyn R. Pais," Model Based Approach to Prevent SQL Injection Attacks on.NET Applications" International Journal of Computer Science & Informatics, Volume-1, Issue-11,2011.

[42] P. Joshi, Dissecting Bias vs. Variance Tradeoff In Machine Learning, 2015[Online]

2 Available:https://prateekvjoshi.com/2015/10/20/dissecting-bias-vs-variance-tradeoff-in-machine-learning/

4 G Buehrer, B.W. Weide, P.A.G Sivilotti, Using Parse Tree Validation to Prevent
3 SQL Injection Attacks, in: 5th International Workshop on SoftWore Engineering and Middlere, Lisbon, Portugal, 2005, pp.106-113.

4 S. W. Boyd and A. D. Keromytis. SQLrand: Preventing SQL Injection Attacks. In
4 Proceedings of the 2nd Applled Cryptography and Network Security Conference, pages 292-302, June2004.