International Journal of Computer Applications Technology and Research
Volume 9–Issue 04, 139-149, 2020, ISSN:-2319–8656

SQL Injection Attacks Predictive Analytics Using Supervised Machine Learning Techniques

[1]Akinsola, Jide E. T.
Department of Computer Science, Babcock University, Ilisan-Remo, Ogun State, Nigeria

[2]Awodele, Oludele
Department of Computer Science, Babcock University, Ilisan-Remo, Ogun State, Nigeria

[3]Idowu, Sunday A.
Department of Computer Science, Babcock University, Ilisan-Remo, Ogun State, Nigeria

[4]Kuyoro, Shade O.
Department of Computer Science
Babcock University,
Ilisan-Remo, Ogun State

Abstract: Structured Query Language Injection Attack (SQLIA) is one of the most prevalent cyber attacks against web-based application vulnerabilities; that are manipulated through injection techniques to gain access to restricted data, bypass authentication mechanisms, and execute unauthorized data manipulation language. There are several solutions and approaches for identification and prevention of SQLIA, such as Cryptography, Extensible Markup Language (XML), Pattern Matching, Parsing and Machine Learning. Machine Learning (ML) approach has been found to be profound for SQLIA mitigation, which is implemented through defensive coding approach. Machine Learning Approach requires a lot of data for efficient model training with capability for using several attack patterns. ML approach can be used to mitigate a very hard blind SQL injection attack. An experimental analysis was performed in Waikato Environment for Knowledge Analysis on Logistic Regression (LRN), Stochastic Gradient Descent (SDG), Sequential Minimal Optimization (SMO), Bayes Network (BNK), Instance Based Learner (IBK), Multilayer Perceptron (MLP), Naive Bayes (NBS), and J48. Hold-Out (70%) and 10-fold Cross Validation evaluation techniques were used to evaluate the performance of the supervised learning classification algorithms to choose the best algorithm. The results of Cross Validation technique showed that SMO, IBK and J48 had Accuracy of 99.982%, 99.995% and 99.999% respectively; while Hold-Out technique showed that SMO, IBK and J48 had Accuracy of 99.986 %, 99.989 % and 100 respectively. On the other hand, in Cross Validation technique SMO, IBK and J48 had time to build model value of 10.15sec, 0.06sec, and 14.12sec respectively while in Hold-Out technique SMO, IBK and J48 had time to build model value of 9.71sec, 0.16sec and 14.28sec respectively. From the findings, IBK had the minimum time to build model in Cross Validation technique in addition to better performance in Accuracy, Sensitivity as well as Specificity and was chosen as the classifier for SQLIA detection and prevention. Therefore, beyond Accuracy, other performance evaluation metrics are critical for optimal algorithm selection for predictive analytics.

Keywords: Cyber security, Injection attack, Injection vulnerability, Predictive analytics, SQLIA, Machine learning, Web application

1. INTRODUCTION

Structured Query Language Injection Attack (SQLIA) is a forceful code manipulation insertion attack targeted against database through vulnerable web applications, which is a source of assault into the database. According to [1], SQLIA is a deliberate query manipulation insertion assaults lunched against backend database through compromised web applications, which exposes the database schema, circumvent confidentiality, integrity and availability of the sensitive information in the database. The prime purpose of which is to steal sensitive information.

Structured Query Language (SQL) injection vulnerability is the one of the most common web-based application vulnerabilities that can be exploited to gain access to restricted data, bypass authentication mechanism, and execute unauthorized data manipulation language [2]. Web application vulnerabilities are cyber security problems. Practically, there have been serious apprehensions on Cyber Security issues in the entire industries. These issues affect organizations and transverse practically all industries, but not limited to distribution, monetary, investment, transmission, transportation and communications. Specific of the most efficient defenses against cyber infringements and disruption are big data and analytics [3]. Web applications have SQL injection vulnerabilities because there is no sanitization of the inputs used in constructing structured output [4]. Vulnerability is an application implementation vulnerability or defect that enables an intruder to trigger unwanted activities or obtain unlawful access. The presence of vulnerability introduce a threat to the application as it may result in a compromise of stored information [5].

SQLIA is a cyber attack aimed at database, which uses manipulation of query language called SQL. The SQL is a language intended to support database information which are stored (being at rest) and on motion. However, it is susceptible to various assaults. Developing methods to detect and avert assaults on database is highly essential while allowing web applications to be as user-friendly as possible [6]. Injection vulnerabilities for example SQL injection and Cross Site Scripting (XSS), rank among the top two out of the top 10 from the analysis conducted by Open Web Application

Security Project (OWASP). SQLIA and XSS are the two most prominent attacks affecting applications running on the web with security vulnerabilities or assaults. Code injection attacks such as SQLIA account for 40.8% and Cross Site Scripting (XSS) attacks account for 11.3% of total attacks in 2018 according to OWASP [7]. Consequent upon this alarming outrage impact on online applications, vulnerability assessment of enterprise Internet-based applications are crucial. There has been upsurge in availability of information and device connectivity have brought about increase in application of machine learning (which is a sub-domain Artificial Intelligence (AI) in diverse areas and data extraction research activities into more prominence to tackle the menace of cyber attacks.

There are different types of approaches for initiating SQL injections such as Tautology, Inference, System Stored Procedure, Piggy-backed Query, Union Query, Logically Query, and Alternate Encodings. However, for effective mitigation of all SQLIA, these approaches can be further characterized into eighteen for effective SQLIA detection and prevention, which are Time-based error, Database Fingerprinting, are Stored procedure, Buffer Overflow, Second Order, Deep Blind, Out of band, Alternate Encoding, Conditional Error, Union, Double blind, Conditional response, Illegal / Invalid / Logical Incorrect, Piggy Back, Error based (blind), Database Mapping, Literal, and Tautology with the non-malicious class referred to as Benign.

There are several solutions and techniques for identification and avoidance of SQLIA, such as Cryptography, Extensible Markup Language (XML), Pattern Matching, Parsing and Machine Learning. Machine learning approach have been found to be profound for SQLIA mitigation. It can handle input type checking, pattern matching and encoding input categories of injection attacks to address login, URL and search vulnerabilities mechanisms. Machine Learning Approach for SQLIA detection and prevention utilizes a lot of data for training using several attack patterns. ML approach can be used to mitigate a very hard blind SQL injection attack. The performance of ML technique is dependent on dataset reliability to meet the intended purpose and elimination of bias in choosing the best classifier for testing and training the model.

Hence, this study implements predictive analytics to detect as well as prevent web application exposures with focus on various SQLIA types / classes. Therefore, mitigation of escalating security breaches using supervised machine learning are addressed from Static and dynamic analysis approaches.

2. LITETRATURE REVIEW

Structured Query Language Injection (SQLI) was first used openly in 2000 and came into existence way back to 1998 [8]. SQLIA has since become one of the most frequent Internet attacks [9]. It happens when the mischievous user changes the allowable or genuine query syntax with the introduction of new SQL keywords or operators that result in unforeseen outcomes not intended for web applications [10]. Code injection is used generally to mean injecting code attacks that are consequently performed by a vulnerable application [11]. SQL Injection may be used to cause serious problems in a variety of ways. An intruder can bypass authentication, gain entrance, modification, and deletion of information within a database by using SQL Injection. SQL Injecting can even be implemented in some instances to execute controls on operating system, which could enable an attacker to scale into a network behind a firewall to further commit devastating attacks [12].

2.1 SQL Injection (SQLI) Types

SQL injections and XSS are the two main security risks with un-sanitized user input. SQLIA is divided into three main arrangements such as In-band SQLI (also referred to as Classic SQLI), Inferential SQLI (also called Blind SQLI) as well as Out-of-band SQLI.

2.1.1 Classic In-band SQLI

The utmost prevalent and straightforward SQLIA is in-band SQLI. In-Band-based SQLI takes place after an intruder both initiate the assault and collect outcomes using the same interaction route. The two most prevalent kinds of SQL injection that are in- band based are SQLI based on errors and SQLI based on Union [12].

2.1.2 Inferential Blind SQLI

No actual transmitted of data through the web application when a SQLI attack is inferential, and the attacker cannot view the consequence of an in-band form of attack. The intruder can redefine the database structure of the database by placing payloads in Inferential SQL injection to observe web application reaction and the resultant database server behaviour. Boolean-based (Content-based) Blind SQLI in addition to time-based blind SQLI are the two kinds of inferential SQL injection [12].

2.1.3 SQLI Based On Out-of-Band

Out - of-band SQLI is not quite prevalent, mainly since it will depend on the functionality of the web application that is used on the database server. SQL Out - of-band Injection happens if an invader cannot start the attack by using the same route and gathering outcomes that will be inimical [12].

2.2 SQL Injection Attack (SQLIA) Types

[13] Opined that the effect of SQL injection assaults could range from delicate data collection to file manipulation, from system-level command execution to application Denial of Service (DoS). The effect relies also on the database of the destination computer and the SQL Statement's functions and preferences. If DOS attack is lunched, it could have devastating effect of the entire system. Thus, rending the web application in-accessible. This could also be in a coordinated fashion referred to as Distributed Denial of Service (DDoS). SQLIA can generally be divided into three classifications:

2.2.1 Attack of First Order

A malicious string can basically be entered and the modified code implemented immediately.

2.2.2 Attack of Second Order

The invader inserts into persistent trusted source storage (for example a table row). The hackers then utilized another activity later to perform an attack.

2.2.3 Injection Based on Literals

The attacker may change the To_Char() implicit function through altering of the environment variables, NLS_Date_Format or NLS_Numeric_Characters values.

2.3 SQL Injection Attack Techniques

The SQLIA techniques focuses on the attack mechanism, which hackers can attempt to carry out the hacking. These seven majorly used SQLIA techniques are: Tautology, System Stored Procedure, Inference, Illegal / Logically Incorrect Query, Alternate Encodings, Union Query and Piggy-backed Query. These mechanisms require effective taxonomical

formulation in order to handle the various forms of SQLIA efficiently. Therefore, classification based on these common seven mechanisms only can create loopholes for the intruders to gain access to database schema, evading detection and hence circumvent confidentiality, integrity and availability of the sensitive information in the database. Thus, the characterization into eighteen classes for effective SQLIA detection and prevention, which are Time-based error, Database Fingerprinting, are Stored procedure, Buffer Overflow, Second Order, Deep Blind, Out of band, Alternate Encoding, Conditional Error, Union, Double blind, Conditional response, Illegal / Invalid / Logical Incorrect, Piggy Back, Error based (blind), Database Mapping, Literal, and Tautology as malicious attacks with the non-malicious class referred to as Benign:

2.3.1 Tautology

This is concerned with one or more conditional statements used to inject code so as to always validate the true statements. This method occurs when the input data to the database is not checked. An instance of such a vibrant SQL statement is the code given thus; query= "SELECT details FROM customer WHERE name=' name' AND pwd=' pwd;' attackers may use tautologies to make use of this software balance by providing an entry parameter number(x' OR' 1'='1') with the significance. An intruder could enter customer data without a relevant consideration because the situation of the WHERE clause becomes the same (which makes the system validates the outcome to be true and terminates the remaining query using (--;). (WHERE='x' Or' 1'='1'--;).

Example of Tautology query attack: **SELECT * FROM employee WHERE name = ‘ ’ or 1=1 -- ’ AND password = ‘12345’;**

2.3.2 Piggy-Backed

The hackers will insert additional queries to be performed by the database in this scenario to extract, input or alter information, service performance denial or carry out commands from distance [14]. Attackers do not attempt to change the initial request in that situation. They actually attempt to attach an additional and different entry to the initial request using personal SQL-based phrases such as OR, AND, INSERT, UPDATE, DROP or DELETE to permit various SQL queries to the database [15].

Example of Piggy-backed query attack: **SELECT * FROM employee WHERE name = ‘guest’ and password = ‘1234’; DROP TABLE employee; -- ;**

2.3.3 Alternate Encoding

Hackers mainly aim to avoid identification when using this technique. In fact, this sort of attack is used to encode the attack strings to avoid the filtering from the programmer (e.g. by using hexadecimal, ASCII and Unicode character set). In reality, additional encodings are generally applied in relation to other attack methods and target dissimilar application levels [16]. The usage of quote (‘) in the SQL statement declaration that can be used in the creation of different form of malicious database query request is prohibited for most of SQL injection mechanism that uses filters.. In place of a single quote which can easily be detected as bad character, for instance, the intruder uses char (44). This attacks combines char () function and ASCII hexadecimal encryption. Real characters(s) are returned when char () function is used to convert to hexadecimal character(s) encoding equivalent.

Example of alternate encoding query attack: **SELECT accounts FROM login WHERE username=" AND password=0; exec (char (0x73687574646j776e))**

2.3.4 Illegal / Logical Incorrect Query

In that assault, attacker attempts injecting declarations which cause the application servers to return a syntax error page to identify injectable parameters, it applies finger-printing and extract data from the web application's back-end databases [17]. In reality, error page gives hackers information about few details of tables’ name in the databases, such as instances, or discloses vulnerable / injectable parameters for an intruder and such details will be used in carrying out the next attack phase [18].

Example of Illegal / Logical Incorrect query attack: **SELECT * FROM employee WHERE name = ‘ ’ UNION SELECT SUM(username) from users -- ’ and password= ‘ ’ ;**

2.3.5 Union Query

In this attack technique, the malicious query is added to the initial request via the UNION keyword to obtain information concerning additional database tables. An intruder can pull out column data or type of data details from this sort of attack [19]. By rule, most of the SQL conforming databases, including SQL Server stores metadata with sysobject numbers, syscolumns, sysindexes, and so on, in a set of system tables. This allows a hacker to use the information about the database table to identify schema information for a database in order to help hackers to lunch assaults to the database further.

Example of Union query attack: SELECT emp_id FROM employee WHERE name = ‘’ **UNION SELECT cardNo FROM creditCard WHERE accNo = 10032 -- AND password = ‘ ’ ;**

2.3.6 Stored Procedures

This method uses vicious SQL codes to execute integrated built-in functions, which further escalate privilege, ensures service denial or to execute remote controls. Indeed, most database providers develop database solutions with standard stored procedures and features to enhance the database functionality and brings interactivity with the operating system. Therefore SQLIAs may be created to perform stored procedures on this particular database once an attacker has known the backend database [19] [20].

Example of stored procedure query attack: **CREATE PROCEDURE DBO @userName varchar2, @pass varchar2, AS EXEC (“SELECT * FROM user WHERE id= ‘ “+@userName+”’ and password= ‘ “+@pass+’”); GO**

2.3.7 Inference

An intruder draws logical conclusion from a response to a right / wrong enquiry about database server answer. Two Blind injection and time injection input methods are used to lunch this attack [21]. In-Blind injection, hackers obtain database information by submitting a server's true / false questions and the answers from this page gives leading information that will be exploited further. If the response is accurate, the request is correct and if the response is wrong, then an error will be triggered. An intruder can therefore obtain implicit response from the database [22]. Part of Inference attack can be classified into Blind SQL injection and Timing Attack.

Example of inference (blind) SQL injection attack: **SELECT * FROM emp_name, emp_address, gender, from employee where 1=0; drop employee**

2.4 Defensive Coding Approaches for SQLIA

Defensive Coding (DC) is one of the SQLIA detection and prevention approaches. It is employed to execute safe queries so that it is compatible with unforeseen inputs or user behaviour in a timely way regardless of the kind of inputs supplied or actions exhibited by the user. The concept being taking advantage of is that every module of the program is exclusively independent. DC approaches being used for web application vulnerabilities mitigation are Cryptography, Input Type Checking, Pattern Matching, Extensible Markup Language (XML), Encoding Input, Parameterized Query and Stored Procedure which can be implemented using Parse Tree Approach or Machine Learning Approach. Parse Tree Approach encompasses input type checking and pattern matching while machine learning approach encompasses input type checking, pattern matching and encoding input. Pattern matching is capable of mitigating SQLIA requests through login, URL and search. Figure 1 shows the defensive coding approaches for SQLIA.

2.4.1 Parsing Approach

Parsing Approach is also known as parse tree approach. This is a technique to detect and avoid a SQLIA on the application's URL, was suggested by [23]. In this technique, the SQL_statement_safe query model was developed as a library with a SQL statement syntax grammar. This grammar syntax was based on two viewpoints, one for a fixed query and the other for a stacked request. It also includes the SQL query tree structure. The query will first be tested on SQL_statement_safe when a user requests SQL query from a website URL to check if the query is single and is consistent with the semantics of a genuine SQL statement declaration.

Figure 1. Defensive Coding Approaches for SQLIA

2.4.2 Machine Learning Approach

In order to identify and prevent SQLIAs, [24] suggested an automation method using the Bayesian algorithm. The monitor intercepts the SQL query, breaks it into numerous keywords based on blank space in a dynamic query, and calculates the length of the SQL dynamic query in the URL from a website when the user sends a dynamic SQL query. Furthermore, amount of keywords is calculated and numerical values and dynamic query keywords are sent to the classifier.

The classification algorithm then calculates the probability of SQL injection in a dynamic query based on the result received from a converter, and then compares the likelihood of SQLI with a user threshold defined as a data set that helps calculate the probability of legitimate query and the likelihood of malicious query. If dynamic SQL query likelihood is calculated by classification algorithm, the query is permitted if there is a match with the likelihood of legitimate query computed in training dataset; otherwise, the query is blocked. In this method, one essential approach or mechanism is to simulate numerous attack patterns in training data, along with a very difficult blind SQL injection attack.

3. MACHINE LEARNING

A specialized area of artificial intelligence (AI), referred to as Machine Learning (ML), focuses on allowing computing systems to learn from data how to automatically perform the desired task. Machine learning is a key technology in the use of data and large data mining technology in diverse fields of healthcare, science, engineering, business and finance, and includes decision making, forecasting, or prediction [25].

3.1 Types of Machine Learning

There are different types of machine learning such as Supervised Learning (SL) and Unsupervised Learning (UL), Semi-Supervised Learning (SSL) in addition to Reinforcement Learning (RL) and Evolutionary Learning (EL) and Deep Learning (DL) [26].

3.1.1 Supervised Learning (SL) Technique

SL trains a system from known input and output data to predict future outputs. The predictive model is developed based on the data input and output. Classification and regression are examples of supervised learning from two different categories. It is used mostly for the prediction and classification of numerical values such as regression and predicting the corresponding class respectively [27].

3.1.2 Unsupervised Learning (UL) Technique

UL technique aims at finding underlying data structures and hidden patterns in data. The datasets consisting of input data without labelled responses are used for drawing inferences. Clustering is a type as well as the utmost prevailing unsupervised method of learning. Unsupervised learning is implemented to locate unknown pattern in turn data grouping. It is mostly applied to market research, object detection, predicting heart attack (medical) and so on. Fuzzy C-Means and k-Medoids, Self-Organized Maps, Gaussian Mixing Models, Hidden Markov Models, Hierarchical Clustering, K-Means and Subtractive Clustering are all algorithms for performing clustering.

3.1.3 Semi Supervised Learning (SSL) Technique

SSL lies between UL as well as SL techniques. It is part or class of the machine learning, including unlabelled training data (e.g. a tiny number of data labelled with a lot of unlabelled data) which includes methods and tasks of the learning. SSL may also be known to be either transductive learning or preparative form of learning [28].

3.1.4 Reinforcement Learning (RL) Technique

RL is one of the Machine Learning techniques that deals on how software agents in an environment should take action to optimize a notion of aggregate reward. In order to maximize recompense in a specific situation, reinforcement is about taking appropriate steps. The best possible conduct or approach to a particular situation is sought through various machines and software. For instance RL is widely used in

Personal Computer games and Robotics for industrial automation.

3.1.5 Evolutionary Learning (EL) Technique

EL is an Evolutionary Computation sub-set, a specific Meta Heuristic Optimization Algorithm based on population. An Evolutionary Algorithm (EA) utilizes biological evolutionary processes, including reproduction, modification, recombination as well as selection.

3.1.6 Deep Learning (DL) Technique

DL is an aspect of ML relying on data depictions rather than algorithms for specific tasks. ML can be supervised form of learning, semi-supervised form of learning [29]. It is applicable in the areas of computer vision, Natural Language Processing (NLP), voice identification, sound identification, machine translation, bioinformatics, drug design, filtering of social networks, analysis of medical images, product inspections and board game programs where Deep Learning frameworks like Deep Neural Networks, Deep Beliefs Networks in addition to Recurrent Neural Networks were created to generate some results similar to and in some instances of higher quality in comparison to human specialists [30].

3.2 Machine Learning Algorithms

Machine learning classification algorithms for evaluation of performance metrics belong to the following four categories / classes of classifiers accordingly such as function (Logistic Regression), Bayes (Bayes Network (BNK) and Naive Bayes (NBS)), Sequential Minimal Optimization (SMO) and Multilayer Perceptron (MLP)), Tree (J48) as well as Lazy (Instance Based Learner (IBK)). It is to be noted that SMO is a variant of Support Vector Machine (SVM). Also MLP is a variant of Artificial Neural Networks (ANNs).

3.2.1 Bayes classifiers

BNK and NBS are examples of Bayes classifiers. Bayes classifiers are probabilistic classifiers relying on the fundamental probability law of Thomas Bayes known as the Bayes Theorem as depicted in equation 1.

$$P(B/A) = \frac{P(B/A)\ x\,P(A)}{P(B)} \quad (1)$$

Equation 1 shows the connection between A and B conditional likelihoods and the likelihoods. A classifier called Naïve Bayes as a classifier is an uncomplicated algorithm having autonomous characteristics that implies that an algorithm believes that the characteristics are not mutually likely. Bayesian networks are comparatively advanced algorithms that evaluate the likelihood of ambiguity and thus allow more complicated information from the analyzed data.

3.2.2 Function Classifiers

The function classifiers are Logistic Regression (LRN), Sequential Minimal Optimization (SMO) and Multilayer Perceptron (MLP).

3.2.2.1 Logistic Regression

This is a classification function using a building class and a single logistic regression multinomial model with a single estimator. Logistics generally specifies where the class border lies. The class probabilities are also determined in a specific approach depending on the distance from the boundary [31]. When dataset is bigger, it passes to ends which are (0 and 1). These probability statements do not just make logistic regression a classifier, but an efficient classifier. It makes stronger, more detailed forecasts and can fit in another way; however, these strong predictions can go wrong sometimes.

3.2.2.2 Sequential Minimal Optimization (SMO)

This is an SVM) variant. Classical Mufti- Layer Perceptron Neural Networks are strongly linked to SVM algorithms. SVMs revolve on both sides of the hyperplanes around the concept of a gap that distinguishes two categories of data [32]. It has been shown that the maximization of the margin and hence the maximum possible distance between the separating hyperplanes and the instances on both sides reduces the upper limit of the expected generalization error [33]. The SVM classification accepts the data of several classes after this now creates vectors for the best hyperplanes to be separated into a feature space or parameter space. The hyperplane that is placed to the closest data points nodes at the highest range is described as ideal [34].

3.2.2.3 Multi-Layer Perceptron (MLP)

The ANN version is MLP. MLP is a categorizing element which determines the weights of the network, not by creating a non-convex, uncompromising minimization issue as in conventional Neural Network training but by addressing a quadratic programming problem with linear limitations [35]. ANN is an algorithm of learning that solves problem of classification. An ANN model consists of several neuron network systems, which are parallel, dynamic and inter-linked. A neuron is used by a defined mathematical processor to generate outputs using inputs [36].

3.2.3 Tree Classifiers

J48 is the Iterative Dichotomiser 3 (ID3) expansion. J48 also contains features for missing values remedy, pruning of decision-trees, continuous value collections for attributes, rules derivation, etc. It's an algorithm for the decision tree. The algorithm named decision tree is used to determine the behaviour of the attributes / vector in several instances. The classes for the recently produced instances are also discovered on the basis of the teaching instances [37].

3.2.4 Lazy Classifiers

Learner based on instances for example IBK is classified as Lazy classifier. It is an algorithm of k-Nearest Neighbour (k-NN). The method is a straightforward and simple method of classifying a certain dataset with fixed apriori K-means algorithm clusters (suppose k clusters). When labelled data are not accessible, K-means algorithms are used [38]. It utilizes a particular way to transform rough thumb rules into extremely precise forecast rule. As a result of weak learning algorithms, classifications (thumb rules) can continuously be at least slightly reliable than random, with about 55 percent accuracy. But a boosting algorithm can likely build one classifier with increased accuracy, say 99 per cent [39].

4. METHODOLOGY

The experimental analysis of the machine learning algorithms was performed using Waikato Environment for Knowledge Analysis (WEKA). A model that can be used for better classification of SQLIA dataset into attack classes effectively was developed using the algorithm with optimal performance. Hold-out (70%) and 10-fold cross-validation evaluation techniques were used to evaluate the performance of the classification algorithms (supervised learning) to choose the best algorithms. This was carried out in relation to evaluating

performance metrics which comprises of criteria such as Kappa Statistic, True Positive (TP) Rate, Accuracy, True Negative (TN) and Training Time (time to build model (TTB)), for each of the machine learning algorithm.

4.1 Performance Evaluation metrics

In evaluating the performance of the classification algorithms, the model was built in WEKA 3.8.0 using the hold-out (70% training data) and 10-fold cross-validation evaluation methods on Logistic Regression (LRN), SMO, Bayes Network (BNK), IBK, Multilayer Perceptron (MLP), Naive Bayes (NBS), and J48. After the training process, the values of benefit criteria such as correctly classified instances (accuracy), Kappa Statistic, True Positive (TP) Rate, True Negative (TN) Rate and Training Time (i.e. Time to Build) were compared.

5. RESULTS

The performance of the machine learning algorithms were measured based on ten (10) existing performance benchmarks: Accuracy, Kappa Statistic, True Positive (TP) Rate, True Negative (TN) Rate and Training Time (i.e. time to build). Tables 1 to 7 and Figures 2 to 6 depict the comparison of the results of the algorithms implemented in WEKA. The choice of algorithm selection for building a model is an important aspect of machine learning problems. The selection of the optimal algorithm should not be based on a singular metric such as accuracy that is mostly chosen by researchers.

5.1 Comparison Based on Accuracy (Correctly Classified Instances)

The result of both Holdout and Cross-Validation methods for Binary Classification showed that the Accuracy outcome for the algorithms are closely related. In Hold-out, SDG and J48 performed excellent equally with 100% Accuracy, followed by LRN. On the other hand, in 10-F C-V, J48 has the best performance flowed by SMO then SDG. However, LRN performance dwindled in relation to others. Table 1 shows the comparison of accuracy results.

Hence, from comparison in Table 1 according to correctly classified instances, the results shows that SDG, J48 and LRN can be used on one hand in Hold-Out as candidate algorithms for model building. Similarly, J48, SDG and IBK can be used on the other hand in 10-F C-V as candidate algorithms for identifying SQL Injection signatures in SQL query strings for effective mitigation as depicted in Figure 2.

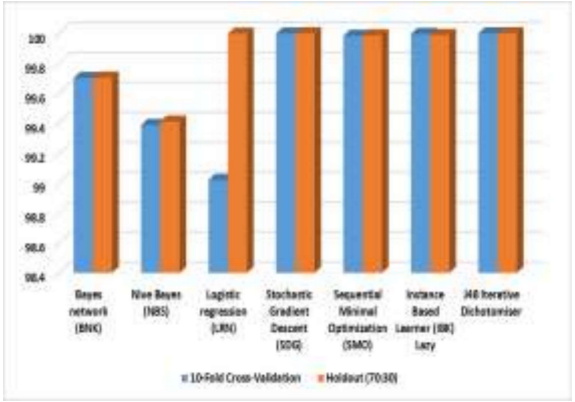

Figure 2. Comparison based on Accuracy of Hold-Out and Cross-Validation Methods

Beyond accuracy, AUC should be keenly considered in algorithm selection and model building [40]. This is highly essential as there could be a lot of false positives as a result of over-fitting and noise in the dataset.

5.2 Comparison Based on Sensitivity (True Positive Rate)

The result of both Holdout and Cross-Validation methods for algorithm classification showed that the Sensitivity outcome for four algorithms such as SDG, SMO, IBK and J48 are the same in both Hold-Out and 10-F C-V methods, thus choosing the best classifier might be biased for model building. Equally, LRN had 100% Sensitivity in Hold-Out method whereas had 99% in 10-F as the least performance. Meanwhile, NBS performed woefully in comparison to other MLAs in Hold-Out method with 99.4% as shown in Table 2.

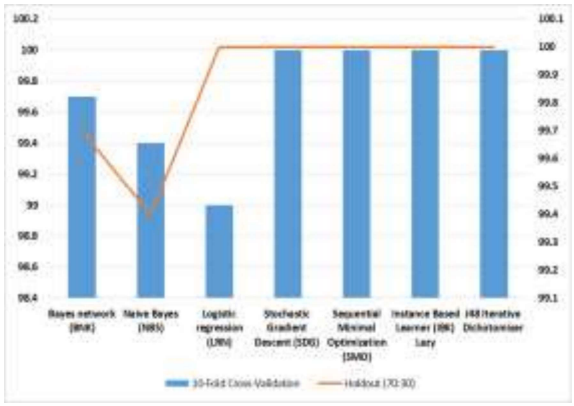

Figure 3. Comparison based on Sensitivity of Hold-Out and Cross-Validation Methods

Figure 3 shows the relationship among various algorithms in terms of sensitivity both in 10 fold Cross validation and Hold-Out techniques.

Therefore, from comparison in Table 2, according to sensitivity, the results shows that J48, SMO, SDG, IBK and LRN can be used on one hand in Hold-Out as candidate algorithms for model building. Likewise, J48, SDG, SMO and IBK can be used on the other hand in 10-Fold Cross Validation as candidate algorithms for detecting SQL Injection signatures in SQL query strings for effective mitigation as depicted in Figure 3. This shows that sensitivity cannot be used in isolation in choosing an optimal algorithm for building the model.

5.3 Comparison Based on Specificity (True Negative Rate)

The outcome of both Holdout and Cross Validation methods for Binary Classification showed that the Specificity outcome for five algorithms such as SDG, SMO, IBK. LRN and J48 are the same in Hold-Out method and four algorithms except LRN, BNL and NBS are the same in 10-Fold Cross Validation method, thus choosing the best classifier might be confusing for model building without taking into consideration the various metrics concerned. Similarly, LRN had the least Sensitivity value in 10-Fold Cross Validation method whereas BNK had 99.3% as the least Sensitivity value in hold-out method as shown in Table 3.

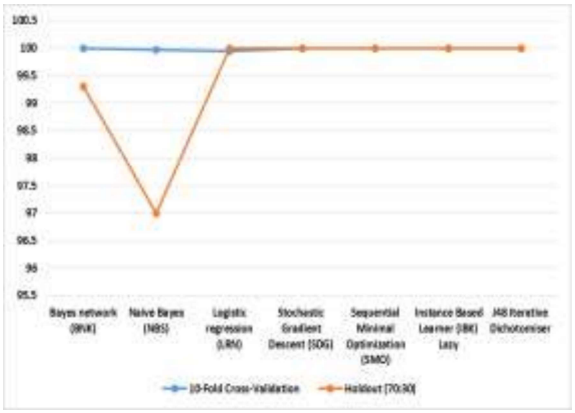

Figure 4. Comparison based on Specificity of Hold-Out and Cross-Validation Methods

Therefore, from comparison in Table 3, based on Specificity, the results revealed that J48, SMO, SDG, IBK and LRN can be used in building the model when Hold-Out is of concern. Equally, J48, SDG, SMO and IBK can be used for model building when 10-Fold Cross Validation is of importance in detecting SQL Injection signatures in SQL query strings for effective prevention of attacks as depicted in Figure 4.

5.4 Comparison Based on Kappa-Statistic

The result of Binary Classification for the algorithms showed that SDG and J48 had the same Kappa-Statistic value of 100% for Hold-Out method. Likewise for 10-Fold Cross Validation with the same value of 99.99%. The least performed algorithm in Hold-Out was NBS and LRN in 10-Fold Cross Validation as revealed in Table 4.

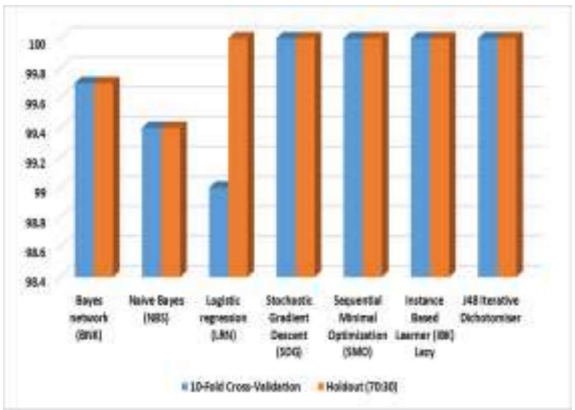

Figure 5. Comparison based on Kappa Statistic of Hold-Out and Cross-Validation Methods.

Therefore, based on comparison in Table 4 in relation to Kappa-Statistic, the results revealed that J48 and SDG can be used in building the model for both Hold-Out and 10-Fold Cross Validation methods. It is to be noted that, Kappa Statistic is a classifier performance measure that estimates the similarity between the members of an ensemble in a multi classifiers systems.

5.5 Comparison Based on Time to Build (Time To Build (TTB))

The end result of algorithm classification for the algorithms showed that IBK had the least conceivable running time to build both at Hold-Out and 10-Fold Cross Validation with values of 0.16second and 0.06 seconds, next in TTB is NBS with 4.09second and 4.95 respectively as shown in Table 5

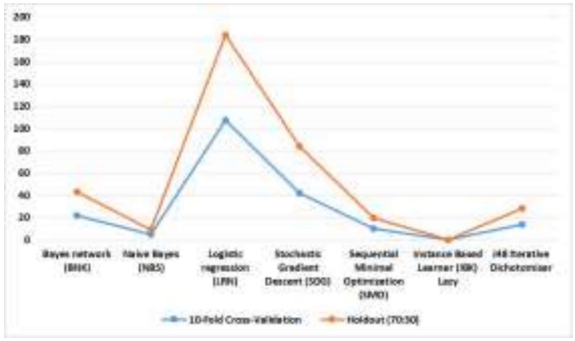

Figure 6. Comparison based on TTB of Hold-Out and Cross-Validation Methods

Therefore, based on comparison in Table 5 according to time to build the algorithms, the results showed that IBK and NBS were those with the least TTB and this does not connote their choice for building the algorithm. The TTB as shown in Figure 6 require the effective decision for optimal algorithm selection since these two algorithms do not have the same specificity and sensitivity values.
Table 6 and Table 7 shows the comparative analysis for all the metrics used in this study for both cross validation and hold-out ML algorithms performance evaluation techniques.

6. CONCLUSION

The results of the performance evaluation of the model for detection and classification of the SQLIA showed a good performance in terms of accuracy, true-positive rate, false-positive rate as well as time to build model. Pattern matching is capable of mitigating SQLIA requests through login, URL and search which can be implemented using machine learning paradigm. Machine learning algorithm selection for model building can be subjective and biased if necessary precautions are not put in place. Therefore, every performance metric must be considered holistically before choosing an optimal algorithm for predictive analytics.

According to [41] to have supervised predictive machine learning, ML algorithms require precise accuracy and minimum errors in addition to putting several factors into consideration. Also, it may be difficult or impossible to find a single classifier doing as well as a good group of classifiers if the only performance metric being utilized is best possible classification accuracy. [42] Opined that Multi-Criteria Decision Method (MCDM) methods can be used to find the optimal classification and regression models in relation to supervised machine learning algorithms.

7. REFERENCES

[1] Sheykhkanloo, N. N. (2017). A Learning-based Neural Network Model for the Detection and Classification of SQL Injection Attacks. International Journal of Cyber Warfare and Terrorism. Available at: https://dl.acm.org/doi/10.4018/IJCWT.2017040102

[2] Aliero, M. S., Ardo, A. A. Ghani, I. & Atiku, M. (2016). Classification of Sql Injection Detection and Prevention Measure. IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org, ISSN (e): 2250-3021, ISSN (p): 2278-8719 Vol. 6, Issue 2, Pp. 6-17. Available at:

https://www.iosrjen.org/Papers/vol6_issue2%20(part-1)/B06210617.pdf

[3] Teymourlouei, H. & Jackson, L. (2017). How Big Data Can Improve Cyber Security. International Conference on Advances in Big Data Analytics (ABDA).

[4] Jagdish, H. (2008). "SQL Injection analysis, Detection and Prevention" (2008). Master's Projects. Available at: http://scholarworks.sjsu.edu/etd_projects/82

[5] Thosar, S., Mane, A. Raykar, S., Jain, R., Khude, P. & Guru, S. (2016). Vulnerability Assessment using Logs as BIG DATA. International Journal of Engineering Science and Computing (IJESC) Available at: https://doi: 10.4010/2016.1156, ISSN 2321 3361, 6 (5), 4644 – 4647

[6] Gaurav, K. T. and Gaurav, O. (2012). Enhanced Query based Layered Approach towards detection and prevention of Web Attacks. Procedia Technology Vol. 4, Published by Elsevier Ltd. Pp. 500 – 505

[7] OWASP (2018). Open Web Application Security Project (OWASP), OWASP Top Ten Project. http://www.owasp.org/index.php/Category: OWASP Top Ten Project.

[8] Fu, X., Lu, X., Peltsverger, B., Chen, S., Qian, K. & Tao, L. (2007). A static analysis framework for detecting SQL injection vulnerabilities. In Proceedings of the 31st Annual International Computer Software and Applications Conference – Volume 01, COMPSAC, 87–96, Washington, DC, USA, 2007. IEEE Computer Society.

[9] Friedl, S. (2005). SQL injection attacks by example

[10] Ali, S, Rauf, A. & Javed, H. (2009). SQLIpa: An authentication mechanism against sql

[11] Stasinopoulos, A., Ntantogiany, C. & Xenakisz, C. (2018). Commix: Automating Evaluation and Exploitation of Command Injection Vulnerabilities in Web Applications

[12] Acunetix (2018). Types of SQL Injection (SQLI). Available at: https://www.acunetix.com/websitesecurity/sql-injection2/

[13] Oracle (2018). Types of SQL Injection Attacks. Available at:http://download.oracle.com/oll/tutorials/SQLInjection/html/lesson1/les01_tm_attacks.htm

[14] Halfond, W.G.J., J. Viegas, & A. Orso, A. (2006). A classification of SQL injection attacks and countermeasures. In Proceedings of the IEEE International Symposium

[15] Khari, M. & Kumar, N. (2013). SQLIA Detection And Prevention Approaches A Survey. International Journal of Computer Science & Information Technology, 3(5)

[16] Kindy, D.A. & Pathan, A. S. K. (2013). A Detailed Survey on various aspects of SQL Injection in Web Applications: Vulnerabilities, Innovative Attacks and Remedies. International Journal of Communication Networks & Information Security, 5(2).

[17] Khari, M. & Kumar, N. (2013). SQLIA Detection And Prevention Approaches A Survey. International Journal of Computer Science & Information Technology, 3(5)

[18] Vinod, K. K & Jatin, D. D. (2013). Advanced Detecting and Defensive Coding Techniques to prevent SQLIAs in Web Applications A Survey. International Journal of Science and Modern Engineering (IJISME), 1(6), 26 - 31 Available at: https://docplayer.net/2925830-Advanced-detecting-and-defensive- coding-techniques-to-prevent-

[19] Roy, S., Singh, A.K. & Sairam, A. S. (2011). Detecting and Defeating SQL Injection Attacks. International Journal of Information and Electronics Engineering, 1(1).

[20] Srivastava, S., (2012). A Survey On: Attacks due to SQL injection and their prevention method for web application.

[21] Borade, M.R. & Deshpande, N.A. (2013). Extensive Review of SQLIA‘s Detection and Prevention Techniques. International Journal of Emerging Technology and Advanced Engineering, Vol. 3 No.10

[22] Khochare, N., Chalurkar, S., Kakade, S. & Meshramm, B. B. (2011). Survey on SQL Injection attacks and their Countermeasures. International Journal of Computational Engineering & Management (IJCEM). 14, 111 – 114 Available at: https://www.ijcem.org/papers102011/ijcem_102011_19.pdf

[23] Narayanan, S. N., Alwyn, R. P., & Mohandas, R. (2011). "Detection and Prevention of SQL Injection Attacks Using Semantic Equivalence." Computer Networks and Intelligent Computing. 103-112, Springer Berlin Heidelberg,

[24] Cheon, E. H., Zhongyue, H. & Yon S. L. (2013). Preventing SQL Injection Attack Based on Machine Learning. International Journal of Advancements in Computing Technology 5(9)

[25] Anwaar, A., Junaid, Q., Raihan, R., Arjuna, S., Andrej, Z. & Jon C. (2016). Big data for development: applications and techniques. Big Data Analytics, 1 / 2, 1- 24. ISSN: 2058-6345. Available at: https://doi: 10.1186/s41044-016-0002-4

[26] Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications. https://doi.org/10.4236/jilsa.2017.91001

[27] Brownlee, J. (2016). Supervised and Unsupervised Machine Learning Algorithms. BSD (2013). 3-clause

BSD license Available at: https://github.com/rapid7/metasploit-framework/blob/master/LICENSE

[28] Zhu, X., & Goldberg, A. B. (2007). Semi-Supervised Learning Tutorial. ICML. https://doi.org/10.2200/S00196ED1V01Y200906AIM006

[29] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. 35(8), 1798-1828. https://doi.org/10.1109/TPAMI.2013.50

[30] Krizhevsky, A., Sutskever, I., Hinton, G. E., Levine, S., Finn, C., Darrell, T., … Szegedy, C. (2012). ImageNet Classification with Deep Convolutional Neural Networks Alex. Proceedings of the 31st International Conference on Machine Learning (ICML-14). https://doi.org/10.1007/s13398-014-0173-7.2

[31] Logistic Regression (LR) pp. 223 – 237. Available at: https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf

[32] Ali, S. & Smith, K. A. (2006). On learning algorithm selection for classification. Applied Soft Computing, 6, 119–138.

[33] Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica 31 (2007). 249 – 268. Retrieved from IJS website: http://wen.ijs.si/ojs-2.4.3/index.php/informatica/article/download/148/140. pp. 249 – 268

[34] Kecman, V. (2005). Support vector machines: An introduction in support vector machines: Theory and applications. In L. Wang (Ed.), 1–47. Berlin: Springer-Verlag. Chapter 1

[35] Taiwo, O. A. (2010). Types of Machine Learning Algorithms, New Advances in Machine Learning, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech, University of Portsmouth United Kingdom, 3 – 31. Available at InTech open website: http://www.intechopen.com/books/new-advances-in-machine-learning/types-of-machine-learning-algorithms

[36] Ko, M., Twari, A., & Mehmen, J. (2010). A review of soft computing applications in supply chain management. Applied Soft Computing, 10, 661–664.

[37] Korting, T. S. (n.d)."C4. 5 algorithm and Multivariate Decision Trees." Image Processing Division, National Institute for Space Research--INPE.

[38] Alex S. & Vishwanathan, S.V.N. (2008). Introduction to Machine Learning. Published by the press syndicate of the University of Cambridge, Cambridge, United Kingdom. Copyright © Cambridge University Press 2008. ISBN: 0-521-82583-0.

[39] Schapire, R. (n.d). Machine Learning Algorithms for Classification.

[40] Akinsola, Jide E. T., Adeagbo, Moruf A., Awoseyi, Ayomikun A. Breast Cancer Predictive Analytics Using Supervised Machine Learning Techniques. International Journal of Advanced Trends in Computer Science and Engineering 8 (6), 3095- 3104, ISSN 2278-3091, November – December 2019. Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse70862019.pdf. DOI: https://doi.org/10.30534/ijatcse/2019/70862019

[41] F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi and J. Akinjobi. Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT) – Volume 48 Number 3*, 2017, https://doi: 10.14445/22312803/IJCTT-V48P126

[42] J. E. T. Akinsola, S. O. Kuyoro, O. Awodele & F. A. Kasali. Performance Evaluation of Supervised Machine Learning Algorithms Using Multi-Criteria Decision Making Techniques. *International Conference on Information Technology in Education and Development (ITED) Proceedings,* 17 – 34, 2019.

Table 1. Comparison based on Accuracy for Hold-Out and Cross-Validation Methods

Machine Learning Algorithms	10-Fold Cross-Validation	Holdout (70:30)
Bayes network (BNK)	99.7015	99.7035
Naive Bayes (NBS)	99.3874	99.41
Logistic regression (LRN)	99.0204	99.9971
Stochastic Gradient Descent (SDG)	99.998	100
Sequential Minimal Optimization (SMO)	99.9824	99.9856
Instance Based Learner (IBK) Lazy	99.9951	99.9885
J48 Iterative Dichotomiser	99.999	100

Table 2. Comparison based on Sensitivity for Hold-Out and Cross-Validation Methods

Machine Learning Algorithms	10-Fold Cross-Validation	Holdout (70:30)
Bayes network (BNK)	99.7	99.7
Naive Bayes (NBS)	99.4	99.4
Logistic Regression (LRN)	99	100
Stochastic Gradient Descent (SDG)	100	100
Sequential Minimal Optimization (SMO)	100	100
Instance Based Learner (IBK) Lazy	100	100
J48 Iterative Dichotomiser	100	100

Table 3. Comparison based on Specificity for Hold-Out and Cross-Validation Methods

Machine Learning Algorithms	10-Fold Cross-Validation	Holdout (70:30)
Bayes network (BNK)	99.993	99.3
Naive Bayes (NBS)	99.969	97
Logistic Regression (LRN)	99.945	100
Stochastic Gradient Descent (SDG)	100	100
Sequential Minimal Optimization (SMO)	100	100
Instance Based Learner (IBK) Lazy	100	100
J48 Iterative Dichotomiser	100	100

Table 4. Comparison based on Kappa-Statistic for Hold-Out and Cross-Validation Methods

Machine Learning Algorithms	10-Fold Cross-Validation	Holdout (70:30)
Bayes network (BNK)	98.39	98.37
Naive Bayes (NBS)	96.67	96.72
Logistic Regression (LRN)	94.63	99.98
Stochastic Gradient Descent (SDG)	99.99	100
Sequential Minimal Optimization (SMO)	99.9	99.92
Instance Based Learner (IBK) Lazy	99.97	99.94
J48 Iterative Dichotomiser	99.99	100

Table 5. Comparison Based on TTB for Hold-Out and Cross-Validation Methods

Machine Learning Algorithms	10-Fold Cross-Validation	Holdout (70:30)
Bayes network (BNK)	22.06	21.1
Naive Bayes (NBS)	4.95	4.09
Logistic Regression (LRN)	107.6	76.56
Stochastic Gradient Descent (SDG)	42.25	41.96
Sequential Minimal Optimization (SMO)	10.15	9.71
Instance Based Learner (IBK) Lazy	0.06	0.16
J48 Iterative Dichotomiser	14.12	14.28

Table 6. Summary of the Model Performance in Cross Validation Method

Machine Learning Algorithms	ACC	TP_R	TN_R	Kappa Statistics	TTB
BNK	99.7015	99.7	99.993	98.39	22.06
NBS	99.3874	99.4	99.969	96.67	4.95
LRN	99.0204	99	99.945	94.63	107.6
SDG	99.998	100	100	99.99	42.25
SMO	99.9824	100	100	99.9	10.15
IBK	99.9951	100	100	99.97	0.06
J48	99.999	100	100	99.99	14.12

Table 7. Summary of the Model Performance in Hold-Out Method

Machine Learning Algorithms	ACC	TP_R	TN_R	Kappa Statistics	TTB
BNK	99.7035	99.7	99.3	98.37	21.1
NBS	99.41	99.4	97	96.72	4.09
LRN	99.9971	100	100	99.98	76.56
SDG	100	100	100	100	41.96
SMO	99.9856	100	100	99.92	9.71
IBK	99.9885	100	100	99.94	0.16
J48	100	100	100	100	14.28