Damascus university Journal for engineering sciences
Vol 39 No.3 (2023) : 18-28

مجلة جامعة دمشق للعلوم الهندسية
المجلد 39– العدد 3 (2023) 18-28

Using Nlp Techniques To Detect Sql Injection Attack

Zaher Alshami[1], Dr. Raouf Hamdan[2]

[1]Master's Student at The Computer Engineering and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.

[2] Dr. The Computer Engineering and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.

Abstract:

Most of the applications used on the internet are Web-Based Applications, that accept critical information from users and store this information in databases.

Being connected to the internet, they are susceptible to all kinds of information security threats, including SQL injection attack.

SQL injection attacks, and web-based attacks fall in general under the top ten vulnerabilities according to the assessment of the most important information security centers and international networks, such as (OWASP) and (ENSIA), which means they continue to be a major issue in the cyber security field.

This paper proposes a method for SQL injection attack detection by using natural language processing techniques (BOW, TF-IDF, Word2Vec, Doc2Vec), and machine learning algorithms (LR, MLP) that allow the machine to automatically learn and detect the characteristic patterns of the query used in SQL injection attacks, which could greatly put an end to attackers' intervention and provide an appropriate defense mechanism against this type of widespread attack.

Keywords: Sql Injection Attack, Cyber Security, Natural Language Processing, Machine Learning, Bow, Tf-Idf, Word2vector, Document2vector.

Received: 9/3/2022
Accepted: 21/6/2022

استخدام تقنيات معالجة اللغة الطبيعية في كشف هجوم حقن SQL

زاهر الشامي[1]، د. رؤوف حمدان[2]

[1] طالب ماجستير في قسم هندسة الحواسيب والأتمتة في كلية الهندسة الميكانيكية والكهربائية – جامعة دمشق.

[2] دكتور في قسم هندسة الحواسيب والأتمتة في كلية الهندسة الميكانيكية والكهربائية – جامعة دمشق.

الملخص

معظم التطبيقات المستخدمة على شبكة الإنترنت هي تطبيقات مستندة إلى الويب، هذه التطبيقات تستقبل المعلومات الهامة من المستخدمين وتخزن هذه المعلومات في قواعد البيانات الخاصة بكل تطبيق.

ونظراً لكونها متصلة بالإنترنت بشكل دائم، فهي عرضة لجميع أنواع تهديدات أمن المعلومات، بما في ذلك هجوم حقن SQL.

تندرج هجمات حقنSQL والهجمات المستندة إلى الويب بشكل عام ضمن أهم عشرة نقاط ضعف وفقاً لتقييم أهم مراكز أمن المعلومات والشبكات الدولية مثل (OWASP) و(ENSIA)، مما يعني أنها لاتزال تمثل قضية رئيسية في مجال الأمن السيبراني.

تقترح هذه الورقة طريقة لاكتشاف هجوم حقن SQL باستخدام تقنيات معالجة اللغة الطبيعية (حقيبة الكلمات، تردد الكلمة–تردد المستند العكسي، تضمين الكلمات، تضمين الجمل)، وخوارزميات التعلم الآلي (الانحدار اللوجستي LR، شبكة MLP)، والتي تتيح للجهاز التعلم والكشف بشكل تلقائي عن الأنماط المميزة للاستعلام المستخدم في هجمات حقنSQL، الأمر الذي من الممكن له أن يضع حداً لتدخل المهاجمين بشكل كبير ويوفر آلية دفاعية مناسبة ضد هذا النوع من الهجوم واسع النطاق.

الكلمات المفتاحية: هجوم حقن SQL، الأمن السيبراني، معالجة اللغات الطبيعية، التعلم الآلي، حقيبة الكلمات، تردد الكلمة–تردد المستند العكسي، تضمين الكلمات، تضمين الجمل.

تاريخ الإيداع: 2022/3/9
تاريخ القبول: 2022/6/21

1.Introduction:

The increase in the development and spread of the web applications has also led to an increase in the number and severity of the web attacks.

According to The Open Web Application Security Project (OWASP), and The European Union Agency for Cybersecurity (ENSIA), the injection vulnerability continues to be the most found vulnerability in web applications [1][2].

The Structured Query Language (SQL) injection attack is considered as the most dangerous attack of the injection category because it compromises the main security services: confidentiality, authentication, authorization and integrity [3].

SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.

A successful SQL injection attack can read sensitive data from the database, modify the date, execute administration operations on the database, recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system, which makes attackers able to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server[3].

With the continuous escalation of this attack methods, traditional filtering systems and Web Application Firewalls (WAF) face many problems in the recent years, so researchers try to benefit from the machine learning techniques to propose more appropriate solutions. Several research works have been done on using various machine learning algorithms to detect SQL Injection attacks. But there is no single perfect algorithm or technique in machine learning that can be applied to a particular problem. Any problem needs to be tested against various algorithms, and the results need to be compared, before finalizing a particular approach, for maximum accuracy.

In this paper, a new model is proposed to detect SQL injection attack by using four natural language processing methods (BOW, TF-IDF, Word2Vec, and Doc2Vec) to extract features from queries after preprocessing, and two machine learning algorithms (Logistic Regression, MLP Neural Network) to train model, finally using these two classifiers to identify the malicious query.

The rest of paper is structured as follows:

- Section2 reviews the background and SQL injection attack types.
- Section3 reviews the related works about research in this area.
- Section4 presents the proposed model and methodology.
- Section5 presents the experimental results and discussion.
- Section6 contains the conclusion with a note on future directions of research.

2. Background:

The fundamental reason for SQL injection attack is to trust the data submitted by users too much, as developers develop their code without filtering user input, or performing reasonable verification on the server-side. Therefore, attackers can change the SQL statement by entering SQL keywords or special symbols, that are passed to the database.

As a result, the system is attacked, and attackers achieve their intended purpose, such as stealing sensitive system information and obtaining server control authority.

SQL injection attack has several types and forms, that are [9]:

- **Tautologies**: the attacker injects a code in one or more conditional statements so that they always evaluate to be true.
- **Illegal/logically incorrect queries:** the attacker inputs a manipulated query into the database to generate an error message.
- **Union:** the attacker uses the UNION operator to join a malicious query to the original query. The result of the malicious query will be joined to the result of the original query.
- **Inference:** the attacker asks the database true or false questions and determines the answer based on the application's response.

- **Piggy-backed:** the attacker intends to inject additional queries to extract data, modify or add data.
- **Alternate encoding**: the attacker tries to conceal the injected text in order to avoid detection by defensive mechanisms.
- **Stored procedures:** the attacker aims to run stored procedures already saved in the database.

3. The Related Works:

Recently, many models have been developed to deal with SQL injection attacks, this section will be discussed some of them.

Hasan *et al*., (2019) [4] presented a model for detecting SQL injection attacks using their own special method for features extraction to represent the query vector, this method depends on calculating six custom features that could be found in one query, that are: (1-any comment character is present, 2- number of semicolons, 3- presence of always true condition, 4- The number of commands per statement, 5- presence of abnormal commands, 6- presence of special keywords), then they used five ML algorithms (Boosted Trees, Bagged Trees, Linear Discriminant, Cubic SVM, and Fine Gaussian SVM) for classification.

The best accuracy between all used classifiers (Boosted Trees) is 93.8%, the total size of dataset is only (616) samples.

In fact, the way used for features extraction in this paper could be ineffective in detecting malicious query, and might lead to more variance error when applying this method on other datasets.

For example, when looking at the feature number three, there are countless ways to write "always true condition" in the malicious query:

and 1=1 --
and substring (123,1,1) =1 --
and substring(0x3a3a,1,1) =0x3a --
and ascii('a') = 97 --

And so on. As one could see, the "always true condition" can't be detected only based on repeated pattern of (number1=number1) or (string1=string1) as the paper suggested. Moreover, the dataset used in the paper is too small.

Subburaj *et al*., (2020) [5] proposed an experimental setup for detecting SQL injection attacks using Term Frequency and Inverse Document Frequency (TF-IDF) technique for features extraction, and four ML algorithms (Naive Bayes, Logistic Regression, SVM, Random Forest, and Extreme Gradient Boosting) for classification.

Despite using four ML algorithms, and the excellent accuracy of 100%, the main limitation of this paper is its dataset, it is too small, consists of only 783 malicious queries and 700 benign queries, as the samples used for testing the model are only 175 according to the paper's results, which makes these results inefficient.

Begum *et al*., (2021) [6] proposed a model using the part of speech (POS) tagging method for features extraction and MLP neural network for classification, the accuracy is 94.4%, the dataset has 1000 samples.

the main drawback of this paper is that it focuses on the Tautology type only, plus the dataset is small.

Farooq. (2021) [7] used twenty-one customized features extracted from queries after the tokenization process, and four ensemble ML algorithms (Gradient Boosting, Adaptive Boosting, Extended Gradient Boosting, and Light Gradient Boosting) for classification.

The extracted features depend on statistics by calculating the total numbers of some parameters that could be found in one query to represent the query vector, such as (single quotations, double quotations, punctuations, white spaces, operators, commands, special characters, etc.).

The accuracy is 99.38%, the total size of dataset is (35198) samples.

Kranthikumar *et al*., (2020) [8] used eleven regular expressions as a classifier which works as a filter to classify the applied query, and three ML algorithms (Naive Bayes, SVM, Gradient Boost) to compare their results with regexp approach, the accuracy of regexp is the best, tally up to 97%, the size of dataset is (20474) samples.

The customized statistical method used in [7], and the regexp approach used in [8], both gave good results, however there is a reason that may reduce their effectiveness at times. Hackers always change their behavior when

formulating the malicious query, so these models always need a comprehensive and large dataset containing all malicious queries used in this type of attacks in order to get results that can be generalized, which is difficult to do.

Since attackers are using new patterns of SQL injections each time, they still seem to successfully get through the various defense mechanisms, so there is a need for SQL injection detection mechanisms that are capable of identifying new attacks, never seen before.

This paper tries to find out other ways to deal with this issue by using machine learning, not only in classification process, but in features extraction process itself, since these techniques have proved their high efficiency in detecting the similarities between words and sentences semantically and syntactically in many tasks.

4. The Proposed Model:

The main motive of the proposed model is to detect SQL Injection attack. The whole procedure is performed as Fig.1 shows:

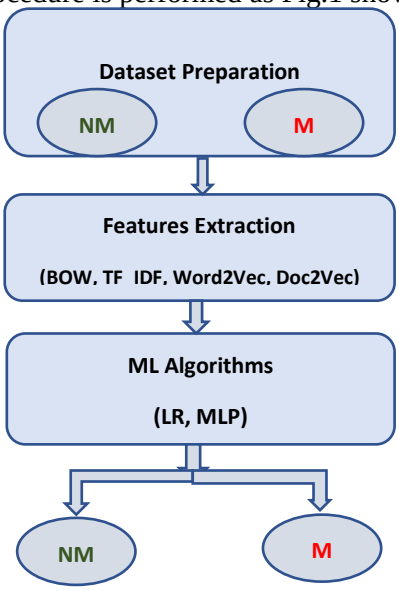

Figure (1) The Proposed Model.

The model consists of four stages:

1- Dataset Preparation: The first stage focuses on collecting the dataset that contains proper SQL injection attack queries, the main challenge is getting a proper dataset containing suitable malicious queries, unfortunately there is no standard dataset issued in this field, However, there is a tool used in cyber security called Libinjection[10], this tool is an open source C library, widely used in conjunction with web applications firewalls to detect SQL injection attack using the lexical analysis, it has been trained on so many various real payloads, therefore, the payloads used by it are captured and used as the SQL injection samples.

These samples mainly contain all SQL injection attack types except the last one (stored procedures), the samples are cleaned, some of them are encoded by URL-Encode for bypassing the measures on WAF, so decoding process is done.

For getting plain text as benign samples, there are textual datasets available for machine learning research purposes [11].

All these malicious and benign samples are collected in a single CSV file as a dataset. The dataset is then labelled (M for malicious, NM for non-malicious), and all samples are lowercase.

The total size of dataset used in this paper is (52609) samples as a whole, (19790) for malicious samples and (32819) for benign samples, it is the biggest dataset between all papers' datasets discussed above.

The tokenization process is done, which is the process of dividing a query into a list of tokens for every word, digit, and special characters inside it.

2-Features Extraction: The second stage deals with extracting features from all queries. In other words, the features extraction is the process of converting every query in the dataset to a vector.

To get the best results and compare with each other, this paper depends on four features extraction methods used in natural language processing to have a better understanding of how these methods would perform over the data, that are:

a) Bag of Words (BOW): In this method, the query is represented as the bag of its words(tokens), the frequency of each word is used as a feature for training a classifier, it doesn't care about the order of the words, all what matters is whether the word is present [12].

Tab.1 shows an example of BOW method.

b) Term Frequency – Inverse Document Frequency (TF-IDF): in this method the frequency of the tokens is rescaled by considering how frequently the

tokens occur in all the queries. As a result, the scores for repeated tokens between all queries are reduced. This way of scoring is known as Term Frequency – Inverse Document Frequency: Term Frequency (TF) is the frequency of the token in the current query. Inverse Document Frequency (IDF) is the score of the tokens among all the queries [12].

Tab.(2) shows an example of TF-IDF method.

Table (1) BOW.

	BOW										
Queries	**select**	*****	**from**	**user**	**name**	**where**	**id**	**=**	**1**	**,**	**union**
select * from user	1	1	1	1	0	0	0	0	0	0	0
select name from user where id =1	1	0	1	1	1	1	1	1	1	0	0
union select 1,1	1	0	0	0	0	0	0	0	2	1	1

Table (2) TF-IDF.

	TF-IDF										
Queries	**select**	*****	**from**	**user**	**name**	**where**	**id**	**=**	**1**	**,**	**union**
select * from user	0.37	0.63	0.48	0.48	0	0	0	0	0	0	0
select name from user where id =1	0.24	0	0.31	0.31	0.41	0.41	0.41	0.41	0.31	0	0
union select 1,1	0.27	0	0	0	0	0	0	0	0.70	0.46	0.46

c) Word to Vector (Word2Vec): The word2vec algorithm uses a neural network model to learn word similarities from large texts, it is very important algorithm in natural language processing, it represents words as a fixed-length vector, so it can represent the degree of similarity between words [13].

Word2Vec has two architectures, (CBOW) and (Skipgram). In this paper, the (CBOW) architecture is used to generate word embedding [13]. Fig.2 shows the structure of (CBOW) architecture.

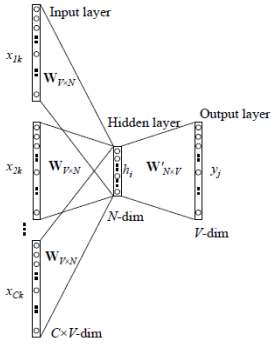

Figure (2) The Word2Vec-CBOWArchitecture [14].

d) Document to Vector (Doc2Vec): The Doc2vec algorithm is considered an expansion of the idea of Word2Vec to represent a whole paragraph or sentence or query in a vector, in this method, every paragraph is mapped to a unique vector, then the paragraph vector is used as an input to predict words [15].

Fig.(3) shows the structure of Doc2Vec for paragraph (the cat sat on the mat).

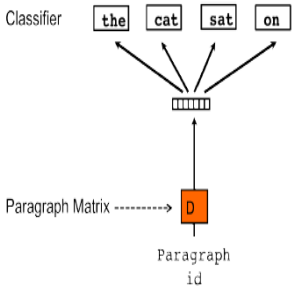

Figure (3) The Doc2Vec Architecture [15].

3- Machine Learning Algorithms: After the completion of features extraction, the main stage is to train the machine learning algorithms with dataset samples for detecting of SQL injection attack.

When it comes to machine learning tasks, it is instructive to approach any task from more than one algorithm perspective, compare between their results to get the best one, so in this research the two classification algorithms are relied on, LR algorithm, which is characterized by its ability to deal with linearly separable data, and MLP algorithm that can find non-linear patterns among the data. The classifiers considered are described below:

a) Logistic Regression (LR): LR is one of the most famous machine learning algorithms used in classification, it is a statistical model that relies on modeling variables according to a mathematical function in order to predict the probability of an output belonging to a particular class. It is characterized by simplicity and great speed in classifying linearly separable data.

The mathematical function used in LR is sigmoid, this function derives the relationship between the variables that represent features and the output that represents a particular class.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

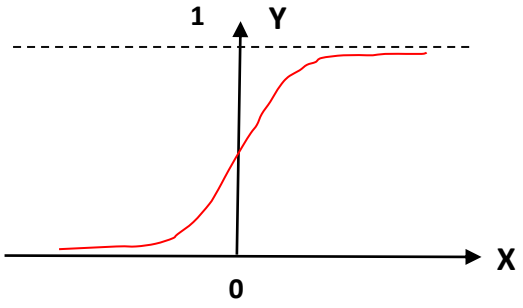

Figure (4) The Sigmoid Function.

Fig.4 shows sigmoid function, b1 represents the initial weight of the variable x in addition to the initial bias value b0, when the term (b0+b1x) is greater than zero, the p representing the value of the sigmoid function will move towards the value of one, when it is less than zero the p will move towards the value of zero.

b) Multilayer Perceptron (MLP): MLP is a feedforward artificial neural network, it is characterized by the fact that each neuron in a particular layer communicates with all neurons in the next layer so that data is constantly fed from one layer to another. Fig.5 shows the general architecture of this type of neural networks.

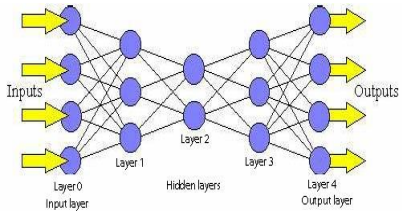

Figure (5) The General Architecture of MLP [16].

The Input Layer receives the features extracted according to the methods explained above, so the neurons of this layer are equal to the number of features that have been extracted for each one method. The Hidden Layers are the group of layers between the input and output layers, the number of neurons in each one hidden layer may differ. The Output Layer represents the prediction given by the neural network, in binary classification tasks such as the paper's task, one neuron is used to determine whether or not the input is malicious (one or zero).

5. Results and Discussion:

The experiment is performed on a 64-bit Windows 10 Home machine, equipped with an AMD A6-7310 APU with AMD Radeon R4 Graphics 2.00 GHz Processor and 8 GB of RAM.

The proposed model is implemented in the Python environment, the main libraries used throughout the research are numpy, pandas, nltk, matplotlib, sklearn, and genism.

The performance metrics such as Accuracy, Recall, Precision, and F1-Score are used to evaluate the model according to the confusion matrix, Tab.3 shows the confusion matrix.

Table (3) Confusion Matrix.

True Positive (TP) Correctly classified as Malicious SQL Query	**False Negative (FN)** Incorrectly classified as Non-malicious SQL Query
False Positive (FP) Incorrectly classified as Malicious SQL Query	**True Negative (TN)** Correctly classified as Non-malicious SQL Query

The performance metrics are:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F1\_Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Before going on, there is a main stage left, which is tuning the parameters of algorithms and methods, this stage can be quite tricky, since some parameters can take a wide range of values. However, there are some systematic methods that can help to estimate the best range or value. The k-fold cross-validation method divides the dataset into k equal sized subsamples, a single subsample is used as the test dataset, and the remaining k − 1 subsamples are used as train dataset, this process is repeated k times to get results which can then be averaged to produce a single estimation.

The k-fold cross-validation approach is useful to test and train a model on a range of values for a single parameter to see how a machine learning model's metric (such as accuracy or f1_score) changes with change in that parameter, so it is chosen to determine the best values of the most important parameters.

Fig.6 shows an example for tuning (hidden_layer_sizes) parameter in MLP algorithm with BOW method, at first one single hidden layer is assumed and a range of neurons is randomly selected for it (1,2,4,6,8,10,13,15), the f1-score at k =3 shows that the best value is 4 neurons, since the f-score value is the highest, and the standard deviation at that number is roughly lower than the rest values, where the black line drawn on bars represents standard deviation.

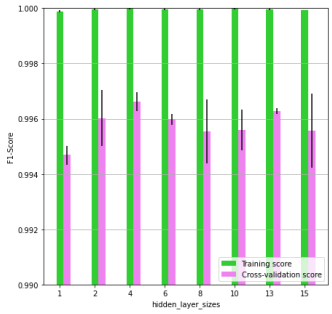

Figure (6) Tunning (hidden_layer_sizes) parameter for MLP with BOW.

It should be noted that when adding other hidden layers and applying this approach, the performance of MLP doesn't show any improvement, which indicates that a single hidden layer consisting of 4 neurons is the best option for MLP.

Tab.4 presents some of the most important parameters for algorithms and methods, which are chosen according to this approach.

The dataset is shuffled and divided to 70% for training and 30% for testing, following extracting features for all four methods, the model is trained using two proposed classifiers, then the system is tested according to test dataset which consists of (17361) total test samples, (6596) for malicious queries and (10765) for plain text.

Tab.5 presents the whole results**, t**he Receiver Operating Characteristic (ROC) curves are drawn in Fig.7 for LR algorithm and Fig.8 for MLP algorithm with all features extraction methods to visualize the True Positive Rate/ False Positive Rate trade-off, as well the Area Under the Curve (AUC).

Table (4) Tunning Parameters.

	Library	**Parameter**	**Specific Values**	**Description**	**Best Value**
BOW	sklearn	ngram_range	[(1,1), (1,2), (2,1), (2,2)]	The lower and upper boundary of the range to be extracted	(1,1)
TF-IDF	sklearn	use_idf	[True, False]	Enable inverse-document-frequency reweighting	True
Word2Vec	genism	vector_size	[10, 25, 50, 100]	Dimensionality of the word vectors	100
		sg	[0,1]	1 for Skip-Gram 0 for CBOW.	0
Doc2Vec	genism	vector_size	[10, 15, 20, 30]	Dimensionality of the doc vectors	15
		epochs	[10,30,50, 100]	Number of iterations	50
LR	sklearn	penalty	[l1, l2, elasticnet, none}	The norm of the penalty	l2
		fit_intercept	[True, False]	Enable bias or intercept (should be added)	True
MLP	sklearn	hidden_layer_sizes	[(1),(2),(4),(6),(8),(10),(13),(15)]	The number of neurons in the ith hidden layer	(4)
		activation	[logistic, tanh, relu]	Activation function for the hidden layer	logistic

Table (5) Performance Metrics.

ML	Features Extractions Tech	Confusion Matrix		Accuracy	Precision	Recall	F1-Score
LR	BOW	6516	80	0.9945	0.998	0.988	0.993
		14	10751				
	TF-IDF	6512	84	0.9944	0.998	0.987	0.993
		13	10752				
	Word2Vec	6589	7	0.9980	0.996	0.999	0.998
		26	10739				
	Doc2Vec	6553	43	0.9961	0.997	0.993	0.995
		23	10742				
MLP	BOW	6593	3	0.9978	0.995	0.999	0.997
		34	10731				
	TF-IDF	6528	68	0.9952	0.998	0.990	0.994
		15	10750				
	Word2Vec	6593	3	0.9984	0.996	0.999	0.998
		24	10741				
	Doc2Vec	6572	24	0.9970	0.996	0.996	0.996
		27	10738				

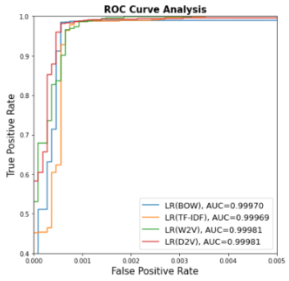

Figure (7) The ROC Curve For LR.

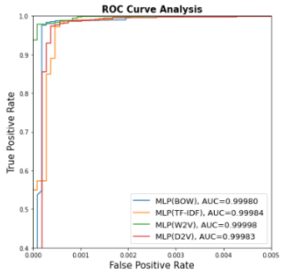

Figure (8) The ROC Curve For MLP.

According to Tab.5, it turns out that all algorithms with features extraction methods used in this experiment give outstanding results, all have at least 99% for accuracy or above.

The best result is MLP algorithm with Word2Vec, with 99.84% Accuracy, 99.6% Precision, 99.9% Recall, and 99.8% F1-score.

Although the results are so converged, the following observations can be concluded:

- The results of Word2Vec method are the best with both algorithms, the reason lies in its ability to comprehend the semantic meaning of words, as it likely will be able through the words it learned while training the model to give better results than other methods, especially statistical methods, when generalizing to other new samples.
- The performance of MLP is slightly better than the performance of LR. However, the results seem to be highly dependent on the features extraction method.

For example, the accuracy of LR algorithm with Word2Vec method is better than the accuracy of MLP algorithm with all BOW, TF-IDF, and Doc2Vev methods.

This remarkable point highlights the importance of features extraction method in NLP tasks regardless the algorithms used for classification later.

Table (6) Comparative Analysis.

Paper	**Size of dataset**	**Accuracy**
Hasan *et al* [4]	616	93.8%
Subburaj *et al* [5]	1483	100%
Begum *et al* [6]	1000	94.4%
Farooq [7]	35198	99.34%
Kranthikumar *et al.*, [8]	20474	97%
This paper	52609	99.84%

In general, the proposed model is successful in predicting and classifying

malicious and benign samples, and it has the best result compared with discussed papers.

Tab.6 shows the comparative analysis with discussed papers in terms of accuracy and size of dataset, the proposed model in this paper has the highest performance.

6.Conclusion:

In this paper, a SQL injection attack detection model has been developed, based on four natural language processing methods for features extraction, that are BOW, TF-IDF, Word2vec, and Doc2Vec, and using two machine learning algorithms for classification, that are LR and MLP.

The main objective of this model is to detect SQL injection attack that is increasing day by day while being used to gain unrestricted access to databases and extract sensitive information, bypass authentication and authorization and finally alter, modify, and delete the databases.

The results have shown the best performance compared with other related works, with 99.84 % Accuracy.

The future work could focus on collecting more samples, trying to test other machine learning techniques used in NLP to discover similarities between sentences, in order to get the best features extraction method for this type of cyber attack.

7.References

1. QWASP, OWASP Top Ten Web Application Security Risks, (2021). https://owasp.org/Top10/ (last accessed Dec 2021)
2. ENISA, ENISA Threat Landscape Report 2020, October 20,(2020). https://www.enisa.europa.eu/publications/enisa-threat-landscape2020-list-of-top-15-threats (last accessed Dec 2021)
3. QWASP, OWASP SQL Injection Overview.

   https://owasp.org/www-community/attacks/SQL_Injection.(last accessed Dec 2021)
4. Hasan, H., Balbahaith, Z., Tarique, M. (2019). "Detection of SQL Injection Attacks: A Machine Learning Approach". International Conference on Electrical and Computing Technologies and Applications (ICECTA).
5. Subburaj, V., Pham, B. (2020). "An Experimental setup for Detecting SQLi Attacks using Machine Learning Algorithms". Journal of The Colloquium for Information Systems Security Education, Volume 8, No 1.
6. Begum, Meharaj., Arock, Michael. (2021) "Efficient Detection Of SQL Injection Attack(SQLIA) Using Pattern-based Neural Network Model". 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).
7. Farooq, U. (2021). "Ensemble Machine Learning Approaches for Detection of SQL Injection Attack". TECHNICAL JOURNAL, pp 112-120.
8. Kranthikumar, B. & Velusamy, R. L. (2020). “SQL injection detection using REGEX classifier. Journal of Xi'an University of Architecture & Technology”, 12(6), 800-809.
9. Jemal, I., Cheikhrouhou, O., Hamam, H., Mahfoudhi, A. (2020). "SQL Injection Attack Detection and Prevention Techniques Using Machine Learning". International Journal of Applied Engineering Research, 15, pp. 569-580.

   https://github.com/client9/libinjection/tree/master/data (last accessed Dec 2021

http://mlg.ucd.ie/datasets/bbc.html (last accessed Dec 2021)

10. Waykole, Resham N., Thakare, Anuradha D. (2018). "A REVIEW OF
11. FEATURE EXTRACTION METHODS FOR TEXT CLASSIFICATION". Scientific Journal of Impact Factor (SJIF),5(04).
12. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). “Efficient estimation of word representations in vector space”. arXiv:1301.3781.
13. Rong, X. (2016). "word2vec Parameter Learning Explained". arXiv:1411.2738v4
14. Mikolov, T., Le, K. (2014). "Distributed Representations of Sentences and Documents". Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043
15. Feed-Forward networks,

https://cs.stanford.edu/people/eroberts/courses/soco/projects/neuralnetworks/Architecture/feedforward.html. (last accessed Dec 2021)