ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

SQL INJECTION ATTACKS DETECTION: A PERFORMANCE COMPARISON ON MULTIPLE CLASSIFICATION MODELS

Anduamlak Abebe[1*], Yonas Belay[2], Adane belay[3], Seffi Gebeyehu[4]

[1,2,3] Department of Computer Science, Debre Tabor University, Debre Tabor, Ethiopia

[4] Bahir Dar University, School of Computing, Bahir Dar, Ethiopia

[*]Corresponding Author's Email: anduamlak09@gmail.com

Abstract

SQL injection attacks pose a significant threat to web applications because they allow unauthorized access to sensitive data and manipulation of databases. Detecting and preventing these attacks is essential for ensuring the security of web applications. While there have been numerous studies on using machine learning to detect SQL injection attacks, there is a lack of comprehensive analysis comparing the performance of different classification models. This research aims to evaluate and compare the effectiveness of various classification models, including KNN, decision trees, support vector machines, Naïve Bayes, and neural networks, in detecting SQL injection attacks. Using a Kaggle dataset with 30919 cases, the study employed an 80%:20% split ratio for training and testing. Data preprocessing was conducted to clean the data by addressing missing values, reducing noise, resolving inconsistencies, and eliminating outliers. The results showed that CNN achieved the highest accuracy (96.55%), with a good balance between precision (98.92%) and recall (91.71%). By evaluating and comparing different classification models, this study contributes to enhancing the security of web applications against SQL injection attacks and advancing research in cybersecurity and machine learning. The study's results can strengthen cybersecurity practices, and defense strategies and empower organizations to proactively defend against evolving threats by creating a better-secured digital environment for web applications and databases.

Keywords: ***Machine learning, Performance Evaluation, SQL injection attacks, Cyber Security, Web applications, Databases***

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*

Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

I. INTRODUCTION

SQL injection attacks represent a prevalent and persistent threat to the security of web applications, posing significant risks to the confidentiality, integrity, and availability of sensitive data. These attacks exploit vulnerabilities in input validation mechanisms, allowing malicious actors to execute arbitrary SQL queries and gain unauthorized access to databases. The consequences of successful SQL injection attacks can be severe, ranging from data breaches and financial losses to reputational damage and legal liabilities [1-4].

In response to the growing sophistication and prevalence of SQL injection attacks, researchers and practitioners have increasingly turned to machine learning techniques for intrusion detection and prevention. By leveraging the inherent patterns and characteristics of SQL injection attempts, machine learning models can effectively distinguish between legitimate user inputs and malicious payloads, enabling proactive defense against cyber threats [5-7].

However, the selection and evaluation of appropriate classification models for detecting SQL injection attacks present considerable challenges. Different machine learning algorithms exhibit varying degrees of performance in terms of accuracy, precision, recall, and computational efficiency. Moreover, the effectiveness of these models can be influenced by factors such as the size and diversity of the training dataset, feature selection, and the presence of noise and imbalanced classes [8-10].

This research focuses on evaluating and comparing the effectiveness of various classification models in detecting SQL injection attacks. The models assessed include K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), Naïve Bayes, and Neural Networks. Utilizing a Kaggle dataset consisting of 30,919 instances, the study employed an 80:20 split ratio for training and testing purposes. Data preprocessing was meticulously conducted to clean the dataset, which involved addressing missing values, reducing noise, resolving inconsistencies, and eliminating outliers.

The experimental results indicated that the Convolutional Neural Network (CNN) model achieved the highest accuracy at 96.55%, exhibiting a strong balance between precision (98.92%) and recall (91.71%). By thoroughly evaluating and comparing these classification models, the study contributes to enhancing the security of web applications against SQL injection attacks and advancing the fields of cybersecurity and machine learning.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

II. LITERATURE REVIEW

Several studies have investigated the use of machine learning techniques for detecting SQL injection attacks in web applications. These studies have explored various classification models, feature selection methods, and evaluation metrics to enhance the accuracy and effectiveness of SQL attack detection systems. In this section, we reviewed some of the selected and most relevant state of art research efforts in this domain.

The research study [11] investigates the use of machine learning techniques to identify and combat SQL injection attacks, a prevalent threat to web applications. SQL injection attacks exploit vulnerabilities in web applications to steal or manipulate data. Traditional methods struggle against evolving attack techniques. The study explored machine learning as a promising approach for detecting SQL injection attacks. The authors trained and compared various machine learning models using datasets containing both benign and malicious web traffic. The models were then assessed based on their ability to identify attacks while minimizing false positives accurately.

A study [12] provides a comparative analysis of various machine-learning algorithms for detecting SQL injection attacks. In this study, the authors evaluated the performance of algorithms such as SVM, Decision Trees, and Neural Networks using features extracted from HTTP requests. Results indicate that ensemble methods, particularly Random Forest, outperform individual classifiers in terms of accuracy and robustness.

Another research paper[13] explored ways of detecting SQL injection attacks using machine learning techniques. The study indicated that Web applications are vulnerable to SQL injection attacks, where malicious code is injected into user queries to manipulate databases. Machine learning models can be trained to analyze SQL queries and identify patterns indicative of attacks. The study might be evaluated on various models like Support Vector Machines (SVM), Decision Trees (DT), or Naive Bayes (NB)on a dataset of labeled SQL queries (malicious vs. legitimate). The research aimed to demonstrate that machine learning models can effectively classify queries and highly accurately detect SQL injection attempts. However, the model's performance might depend on the training data's quality and size. Complex models can be difficult to interpret, hindering understanding of their decision-making process.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

***ISSN** (E): 2959-3921*
(P): 2959-393X
***DOI:** https://doi.org/10.59122/154CFC15*

Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

A study [14] proposed an ensemble classification approach for detecting the attack level of SQL injections in web applications. The research introduced a novel method that combines multiple classification algorithms to accurately identify the severity or level of SQL injection attacks. By leveraging the strengths of ensemble learning, the approach aims to enhance detection accuracy and robustness. Experimental results demonstrated the effectiveness of the proposed method in accurately classifying SQL injection attacks based on their severity, thus providing valuable insights for improving cybersecurity measures in web applications.

Similar studies [15, 16] presented a novel method for detecting SQL injection attacks in web applications. The research utilizes Convolutional Neural Networks (CNNs), a type of deep learning architecture, to automatically learn and identify patterns indicative of SQL injection attempts from raw HTTP request data. By leveraging the hierarchical feature extraction capabilities of CNNs, the proposed approach achieves high accuracy in distinguishing between benign and malicious HTTP requests. Experimental results demonstrate the effectiveness of the CNN-based approach in mitigating SQL injection threats and showcase its potential for enhancing cybersecurity measures in web application environments.

A study [17] explores the effectiveness of machine learning techniques in detecting SQL injection attacks by analyzing multiple data sources. It investigates various machine learning algorithms and evaluates their performance in identifying SQL injection attempts using diverse datasets. By considering different data types and feature sets, the research aims to enhance the accuracy and robustness of intrusion detection systems. The findings provide valuable insights into selecting optimal machine learning techniques and feature representations for effectively detecting SQL injection attacks across various web application environments.

Another study [18], proposed a detection model for SQL injection attacks that leverages the Chi-Square statistic along with classification techniques. The research introduced a novel method that combines feature selection through Chi-Square analysis with various classification algorithms to accurately identify SQL injection attempts in web applications. By selecting relevant features and employing classification techniques, the model aims to effectively distinguish between benign and malicious SQL queries. Experimental results demonstrated the efficacy of the proposed approach in detecting SQL injection attacks and highlighted its potential for enhancing cybersecurity measures in web application environments.

***Received:** March 19, 2024; **Revised**: June 25, 2024; **Accepted:** June 30, 2024; **Published:** 24 July 2024.*
*Corresponding author- **Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15

Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

A very relevant study[19], presents an approach to detect SQL injection attacks using machine learning classifiers. The study utilizes features extracted from HTTP requests to train classifiers, such as Support Vector Machines (SVM) or Decision Trees, to distinguish between legitimate and malicious queries. By leveraging machine learning, the method aims to enhance the accuracy and efficiency of detecting SQL injection attacks in web applications. Experimental results demonstrated the effectiveness of the classifier in identifying suspicious queries, thus offering a promising solution to bolster cybersecurity measures against SQL injection threats.

The rigorous review of the state-of-the-art studies indicates that there is a critical need for comprehensive performance evaluation studies to assess the efficacy of multiple classification models in detecting SQL injection attacks. Such evaluations can provide valuable insights into the strengths and limitations of different machine learning approaches, inform the selection of optimal detection strategies, and guide the development of more robust and resilient security mechanisms for web applications.

This paper aims to fill this research gap by conducting a thorough performance evaluation of various classification models for detecting SQL injection attacks. Specifically, the study analyzes the effectiveness of K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and Convolutional Neural Network (CNN) in accurately identifying and classifying SQL injection attempts. By comparing and contrasting the performance of these models across multiple metrics, including accuracy, precision, and recall, we seek to provide insights that can inform the design and implementation of effective security measures for mitigating the risks posed by SQL injection attacks.

This research aimed to propose a novel approach for detecting SQL injection attacks in web applications by leveraging deep learning techniques. The primary objectives of the study were:

1) Developing a machine learning model trained on a large dataset of benign and malicious SQL queries to automatically learn the patterns and characteristics of SQL injection attacks.
2) Demonstrating the effectiveness and robustness of the proposed machine learning approach through extensive experimentation and evaluation of various datasets.
3) Comparing the performance of the deep learning model with other machine learning-based detection methods to showcase its superiority in terms of accuracy, sensitivity, and specificity.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

4) Highlighting the adaptability of the deep learning model to new and unseen attack patterns
5) Enhancing the security posture of web applications against SQL injection attacks.
6) Contributing to the advancement of security mechanisms for web applications by introducing a sophisticated and efficient method for detecting SQL injection attacks using deep learning techniques.

III. METHODOLOGY

The methodology as shown in Fig. 1 provides a framework for evaluating the performance of multiple classification models in detecting SQL injection attacks.

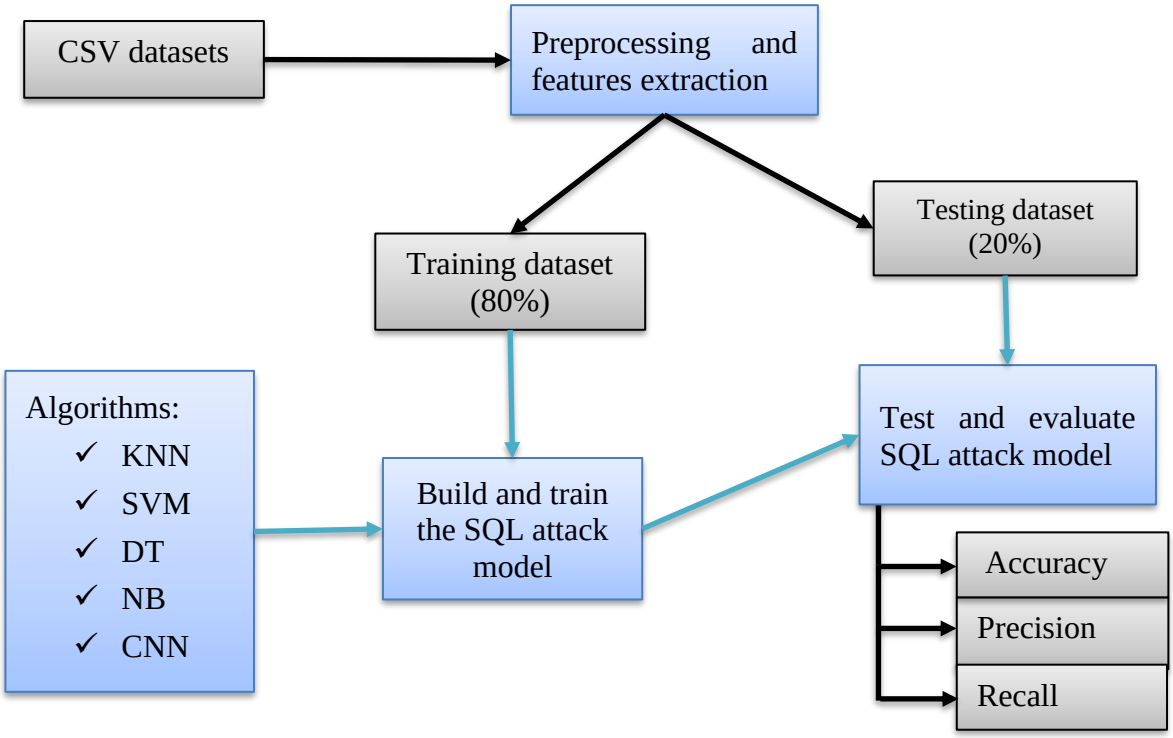

Fig. 1. Framework for proposed SQL injection attack detection

A. Dataset Information and Preprocessing

Whatever the dataset size, building a well-cleaned representative dataset is more important than deciding on a particular learning method [20]. We employed an 80%:20% split ratio to train and test the proposed model, respectively. The Kaggle dataset contains a separate SQL injection attack dataset with 19,589 cases labeled 0 indicating the legitimate query and 11,330 instances labeled 1 indicating the malicious query. The study was conducted using 30,919 cases in total. The

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

heterogeneous origin of most real-world machine-learning datasets makes them particularly prone to missing, inconsistent, and noisy data. Data preprocessing was used to clean the data by correcting missing values, reducing noise, resolving inconsistencies, and eliminating outliers.

B. Implementation of Tools and Algorithms

Python was selected as the study's implementation language due to the number of libraries and packages designed specifically for Deep learning research. Python offers popular libraries like Scikit-learn, TensorFlow, and PyTorch for building and evaluating models. The study conducted extensive experiments utilizing Google Collaboratory to test and train the neural network[21-27].

C. Performance Evaluation Metrics

To forecast which class instance belongs to which class in a computational issue like classification and detection, evaluation measures like accuracy, precision, recall, and F1-score are used. The model's performance in each class was shown by those measures, which were calculated using classification metrics. The confusion matrix value is used to calculate the following metrics: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). In this study, we use accuracy, precision, recall, and confusion matrix to evaluate the models' strengths and weaknesses as shown in Table I [28, 29].

Table I: Proposed SQL injection attack detection evaluation matrix

Evaluation metrics	Formula
Accuracy	$\text{Accuracy} = \frac{TP+TN}{P+N}$
Precision	$\text{Prcision} = \frac{TP}{TP+FP}$
Recall	$\text{Recall} = \frac{TP}{TP+TN}$

IV. RESULTS AND DISCUSSIONS

A. Experimental Setups

In this study, the employed machine learning and Deep learning algorithms are different hyperparameters. The CNN algorithm's hyperparameters include the optimization algorithm, learning rate, loss function, number of epochs, and batch size as shown in Table II and Table III.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

Table II: Hyperparameter value summary for machine learning

Learning algorithm	Optimal values		
KNN	n-neighbors=3	Weights='uniform'	The default for other parameters
SVM	Kernel='rbf'	C=1.0	Gamma='auto'
DT	DT classifier		
NB	function= gaussianNB()		

Table III: Hyperparameter value summary for CNN

Hyperparameters	**Selected**
Optimization algorithm	Adam
Learning rate	0.01
Activation function	Sigmoid
Loss function	Binary cross-entropy
Epoch	10
Batch size	32
Dense layer	256

B. Experimental Results and Discussion

The manuscript employed five selected machine-learning algorithms to unleash their performance using the Kaggle SQL attack dataset. Table IV shows the performance metrics (accuracy, precision, and recall) of five different classification models namely; K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Convolutional Neural Network (CNN).

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

Table IV: Multiple classification models for detecting SQL injection attacks result

Metrics	KNN	SVM	DT	NB	CNN
Accuracy	82.6	77.02	91.51	80.27	96.55
Precision	71.35	99.88	87.61	65.42	98.92
Recall	88.61	38.028	89.78	99.17	91.71

As shown in Table IV, CNN achieved the highest accuracy (96.55%) in correctly classifying both malicious and legitimate SQL queries. SVM (77.02%) and NB (80.27%) had lower accuracy, indicating a higher rate of misclassified queries. DT (91.51%) and KNN (82.6%) performed better than SVM and KNN in terms of accuracy.

The SVM (99.88%) achieved exceptionally high precision, indicating that most of the queries it classified as malicious were truly malicious (low rate of false positives). However, SVM's lower accuracy (77.02%) suggests it might have missed a significant portion of malicious queries (high rate of false negatives). This is supported by its low recall value (38.028%). CNN (98.92%) achieved a good balance between precision and recall, indicating it correctly identified most malicious queries it classified (low false positives) while also catching a good portion of actual attacks (refer to recall for details).

NB (99.17%) achieved the highest recall, meaning it identified a very high proportion of actual malicious queries. However, its lower accuracy (80.27%) suggests it might have classified many legitimate queries as malicious (high false positive rate) which is further supported by its lower precision (65.42%). CNN (91.71%) achieved a good recall, indicating it captured a significant portion of malicious queries (low rate of false negatives).

For this task, CNN seems to be the most reliable model when taking into account all criteria (accuracy, precision, recall). It exhibited the best balance between detecting fraudulent queries and reducing false alarms, as seen by its greatest accuracy and good precision and recall.

Finally, the proposed model was evaluated using the evaluation metrics mentioned in the earlier section. The confusion metrics of the implemented machine learning algorithms are described in Fig 2, Fig 3, Fig 4, Fig 5, and Fig 6.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

***ISSN** (E): 2959-3921*
(P): 2959-393X
***DOI:** https://doi.org/10.59122/154CFC15*

Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

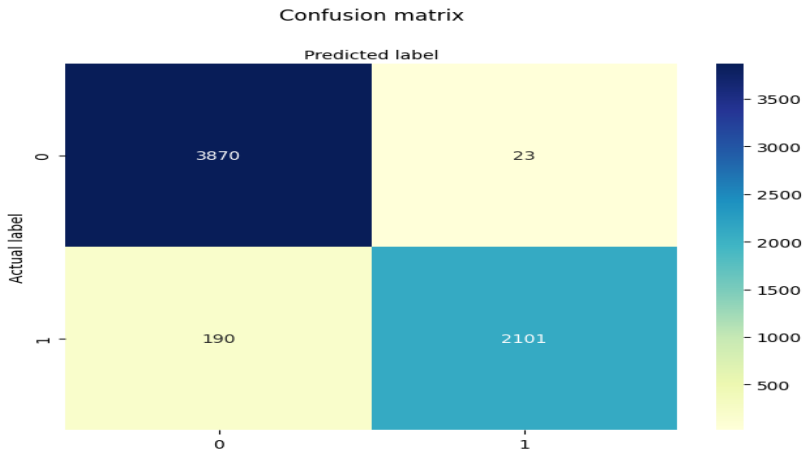

Fig. 2. Confusion metrics for KNN Algorithm

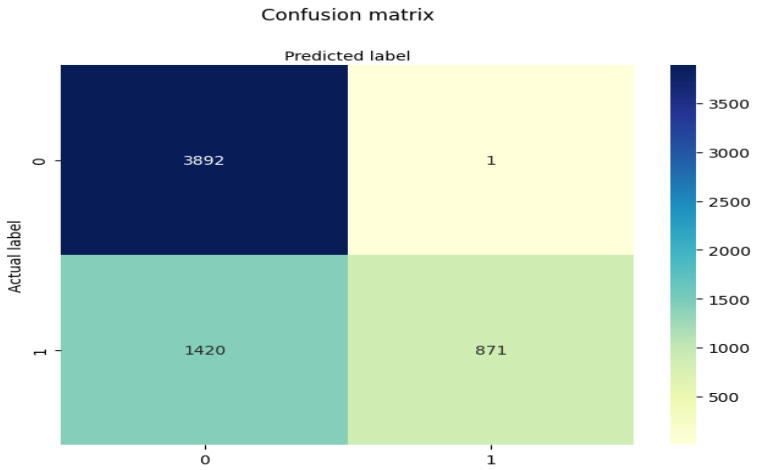

Fig. 3. Confusion metrics for SVM Algorithm

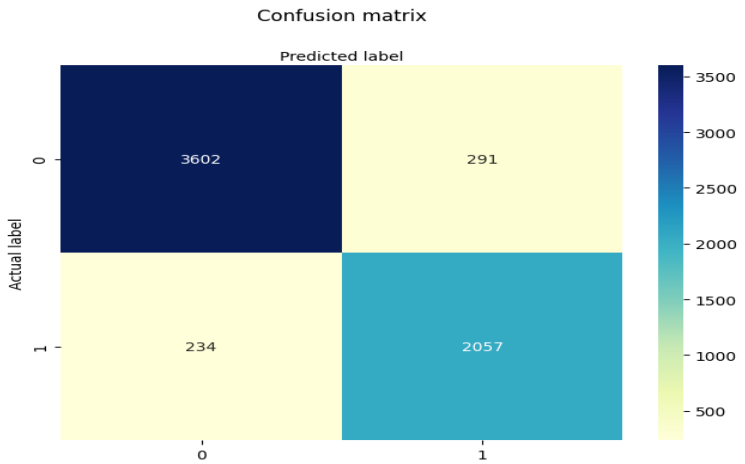

Fig. 4. Confusion metrics for DT Algorithm

***Received:** March 19, 2024; **Revised**: June 25, 2024; **Accepted:** June 30, 2024; **Published:** 24 July 2024.*

*Corresponding author- **Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

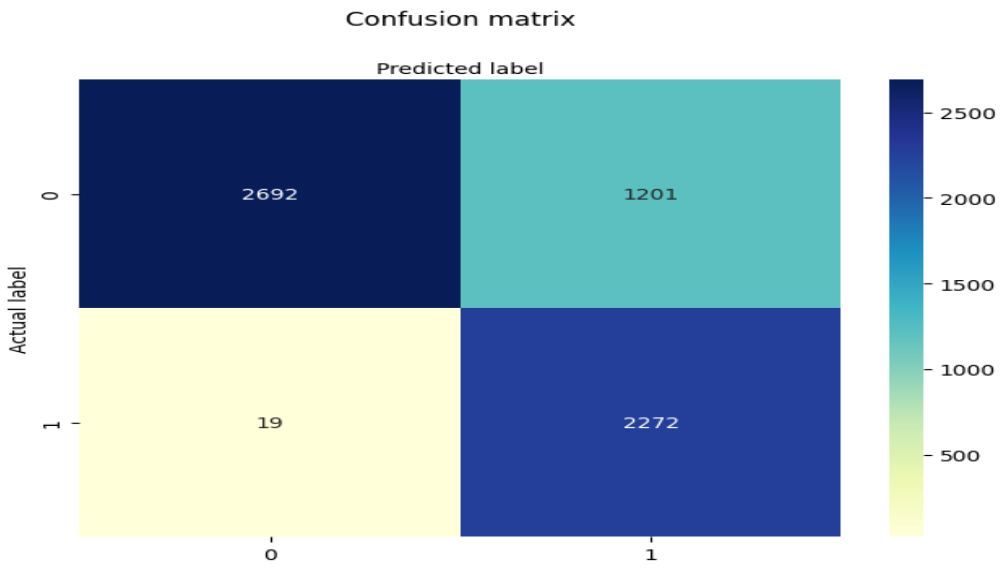

Fig. 5. Confusion metrics for NB Algorithm

Fig. 2 shows the training and testing accuracy of a CNN model. The x-axis represents the epoch, which is a full pass through the training data. The y-axis represents accuracy, a measurement of how well the model performs on a given SQL attack dataset.

The training accuracy (blue line) consistently increases as the number of epochs increases. This suggests that the model is learning the patterns in the training data and improving its ability to classify data points correctly. The testing accuracy (green line) increases similarly to the training accuracy but remains slightly lower. This is a good sign because it suggests the model is not overfitting the training data.

A smaller gap suggests better generalization, where the model can perform well on new data it has not been explicitly trained on. It suggests that the CNN model is effectively learning from the training data and generalizes reasonably well to unseen data. The model achieves high training accuracy and maintains good testing accuracy.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

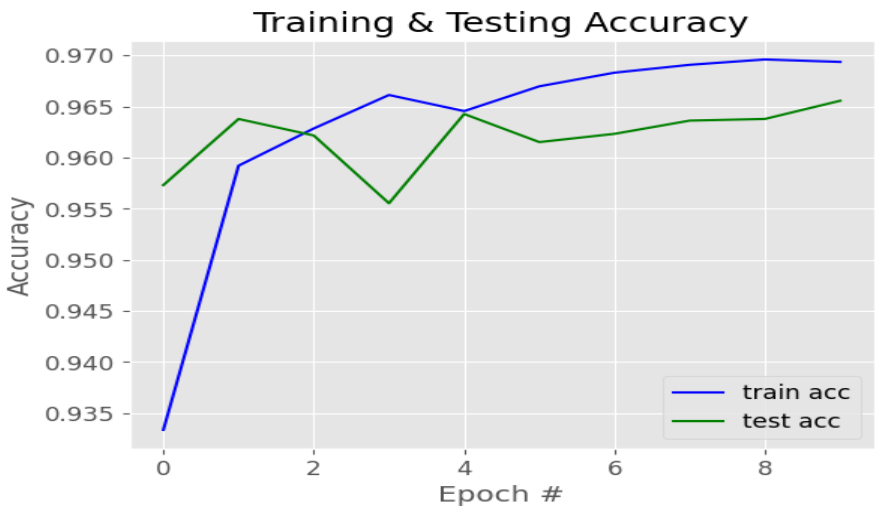

Fig. 2. Training and testing accuracy of the CNN Algorithm

The Fig. 3, depicts a line graph showing the training and testing loss of a Convolutional Neural Network (CNN) model. The x-axis represents the epoch, which signifies one complete pass through the training data. The y-axis represents loss, a metric that indicates how well the model's predictions align with the actual target values. Lower loss values signify better model performance during training and testing as shown in Fig. 3.

The training loss (red line) generally decreases across epochs, suggesting the model is progressively learning from the training data and minimizing its prediction errors. The testing loss (orange line) follows a similar trend as the training loss, but it fluctuates more and stays consistently higher than the training loss.

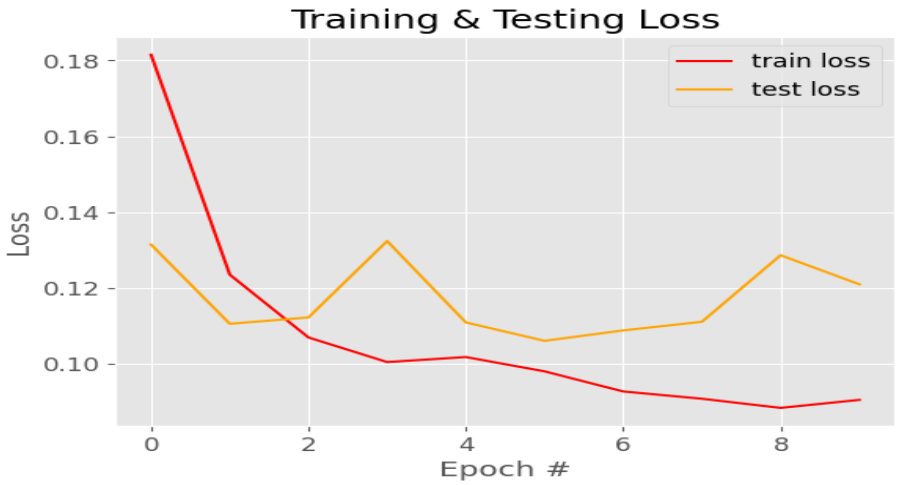

Fig. 3. Training and testing loss of CNN algorithm

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

ISSN (E): 2959-3921
(P): 2959-393X
DOI: https://doi.org/10.59122/154CFC15
Ethiopian International Journal of Engineering and Technology (EIJET)
Volume 2, Issue 1, 2024

C. Result in comparison with the state-of-the-art solutions

In addition to comparing and evaluating the performance of the ML algorithms deployed in this work, the authors also compared such algorithms with the existing related works, as shown in Table V.

Table V: Result comparison from the related works

Author	Title	Methods used	Result %
[11]	Detection of SQL Injection Attacks: A Machine Learning Approach	Ensembled bagged tree	93.8%
[15, 16]	A deep learning approach for detection of SQL injection attacks using convolutional neural networks.	CNN	94.84%
[5]	A Machine Learning Methodology for Detecting SQL Injection Attacks.	CNN	92.7%
Ours	SQL Injection Attacks Detection: Performance Comparisons on Multiple Classification Models	CNN	96.55%

V. CONCLUSION AND RECOMMENDATION

In conclusion, the performance evaluation of multiple classification models for detecting SQL injection attacks provides valuable insights into the efficacy of different approaches in safeguarding web applications against malicious intrusions. Through comprehensive analysis and comparison of various machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and Convolutional Neural Network (CNN) valuable experimental results are obtained. The findings of this evaluation provide valuable insights for organizations tasked with securing web applications against SQL injection attacks. By leveraging advanced machine learning techniques such as CNN and DT, institutions can enhance their ability to detect and mitigate security threats effectively.

Additionally, the performance evaluation highlights the importance of considering factors such as interpretability, computational resources, and the balance between different performance metrics when selecting an appropriate detection model.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

Among the evaluated models (KNN, SVM, DT, NB, CNN), CNN achieved the highest overall accuracy (96.55%) in correctly classifying both malicious and legitimate SQL queries. It also maintained an acceptable balance between precision (98.92%) and recall (91.71%), indicating it effectively identified malicious queries while minimizing false alarms.

By evaluating and comparing the performance of different classification models, the study can help enhance the security measures of web applications against SQL injection attacks. The comparative analysis of multiple classification models contributes to advancing research in the field of cybersecurity and machine learning. strengthening cybersecurity practices, empowering organizations to proactively defend against evolving threats, and fostering a more secure digital environment for web applications and databases.

Based on the analysis of various machine learning models for detecting SQL injection attacks, the author recommends the exploration of additional models beyond KNN, SVM, DT, NB, and CNN. Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks might be suitable for capturing sequential patterns in SQL queries using ensemble methods like Random Forests or Gradient Boosting that combine multiple models and optimize hyperparameters (e.g., learning rate, number of filters in CNN) for each model to improve its performance.

Declaration of Conflicts of Interest

All authors declare that there are no conflicts of interest.

Data Availability

All the necessary data will be found from the corresponding author for a reasonable request.

REFERENCES

[1]. M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, "A systematic review of detection and prevention techniques of SQL injection attacks," *Information Security Journal: A Global Perspective*, vol. 34, no. 4, pp. 252 -265, 2023.

[2]. I. Jemal, O. Cheikhrouhou, H. Hamam, and A. Mahfoudhi, "SQL injection attack detection and prevention techniques using machine learning, " *International Journal of Application Engining Research*, vol. 15, no. 6, pp. 569 - 580, 2020.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*

Corresponding author- ***Anduamlak Abebe***

[3]. F.K. Alarfaj and N.A. Khan, "Enhancing the performance of SQL injection attack detection through probabilistic neural networks," *Applied Sciences*, vol. 13, no. 4365, pp. 1 -11, 2023.

[4]. S.A. Krishnan, A.N. Sabu, P.P. Sajan, and A.L. Sreedeep," SQL injection detection using machine learning," *Revista Geintec-Gestao Inovacao E Tecnologias*, vol. 11, no. 3, pp. 300 -310, 2021.

[5]. A. Gupta, L. K. Tyagi, and A. Mohamed, "A Machine Learning Methodology for Detecting SQL Injection Attacks," *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, Tashkent, Uzbekistan, pp. 184-191, 2023. doi: 10.1109/ICTACS59847.2023.10390153

[6]. A. A. Ashlam, A. Badii and F. Stahl, "A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks," *2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, Hammamet, Tunisia, pp. 513-517, 2022, doi: 10.1109/IC_ASET53395.2022.9765948.

[7]. M.H.A. AL-Maliki and M.N. Jasim, "Review of SQL injection attacks: Detection, to enhance the security of the website from client-side attacks," *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, pp. 3773 - 3782, 2022.

[8]. M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL injection attack using machine learning techniques: a systematic literature review," *Joural of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 764 - 777, 2022.

[9]. H. He and E.A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263 - 1284, 2009.

[10]. B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221 -232, 2016.

[11]. M. Hasan, Z. Balbahaith and M. Tarique, "Detection of SQL Injection Attacks: A Machine Learning Approach," *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, United Arab Emirates, pp. 1-6, 2019. doi: 10.1109/ICECTA48151.2019.8959617.

[12]. V. Sharma and S. Kumar, "Comparative Study of Machine Learning Algorithms for Prediction of SQL Injections," in Computer Vision and Robotics. Algorithms for Intelligent

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*
Corresponding author- ***Anduamlak Abebe***

Systems, Shukla, P.K., Singh,K.P., Tripathi, A.K., Engelbrecht, A. Eds. Singapore: Springer, pp. 2023. https://doi.org/10.1007/978-981-19-7892-0_36

[13]. M.A. Azman, M.F. Marhusin, and R. Sulaiman, "Machine learning-based technique to detect SQL injection attack," *Journal of Computer Science*, vol. 17, no. 3, pp. 296 -303, 2021.

[14]. Ö. Kasim, "An ensemble classification-based approach to detect attack level of SQL injections," Journal of Information Security and Applications, vol. 59, no. 102852, 2021.

[15]. A. Falor, M. Hirani, H. Vedant, P. Mehta, and D. Krishnan, "A deep learning approach for detection of SQL injection attacks using convolutional neural networks," in *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2*, 2022.

[16]. M. Hirani, et al., "A Deep Learning Approach for Detection of SQL Injection Attacks using Convolutional Neural Networks," Department of Computer Engineering, MPSTME, NMIMS University, Mumbai, India, 2020.

[17]. K. Ross, et al. "Multi-source data analysis and evaluation of machine learning techniques for SQL injection detection," in *Proceedings of the ACMSE 2018 Conference*, Gupta, D., Polkowski, Z., Khanna, A., Bhattacharyya, S., Castillo, O. Eds. Singapore: Springer, 2018. https://doi.org/10.1007/978-981-16-6285-0_24.

[18]. M. O. Adebiyi, M. O. Arowolo, G. I. Archibong, M. D. Mshelia, and A. A. Adebiyi, "An SQL Injection Detection Model Using Chi-Square with Classification Techniques," *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Cape Town, South Africa, pp. 1-8, 2021. doi: 10.1109/ICECET52533.2021.9698771.

[19]. P. Roy, R. Kumar and P. Rani, "SQL Injection Attack Detection by Machine Learning Classifier," *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2022, pp. 394-400, doi: 10.1109/ICAAIC53929.2022.9792964.

[20]. I.H. Sarker, "Data science and analytics: an overview from data-driven smart computing, decision-making, and applications perspective," *SN Computer Science*, vol. 2, no. 377, pp. 1 -22, 2021.

***Received:** March 19, 2024; **Revised**: June 25, 2024; **Accepted:** June 30, 2024; **Published:** 24 July 2024.*
*Corresponding author- **Anduamlak Abebe***

[21]. J. Gareth, D. Witten, T. Hastie, and R. Tibshirani, "*An introduction to statistical learning: with applications in R,"* Houston, TX, USA: Springer, 2013.

[22]. C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, VOL. 20, PP. 273 -297, 1995.

[23]. B.Y. Kasula, "Enhancing Classification Precision: Exploring the Power of Support-Vector Networks in Machine Learning," *International Scientific Journal for Research*, vol. 1, no. 1. pp. 1 - 7, 2019.

[24]. B. Clarke, E. Fokoue, and H.H. Zhang, *Principles and Theory for Data Mining and Machine Learning,*" London and New York: Springer, 2009.

[25]. I.H. Witten, E. Frank and M. A. Hall. "*Data Mining: Practical Machine Learning Tools and Techniques,"* Amsterdam, Netherlands: Elsevier Amsterdam, 2005.

[26]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 426 -444, 2015.

[27]. I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts: MIT Press, 2016.

[28]. J. Hernández-Orallo, "ROC curves for regression," *Pattern Recognition*, vol 46, no. 12, pp. 3395-3411, 2013.

[29]. A.N.A. Tosteson and C.B. Begg, "A general regression methodology for ROC curve estimation," *Medical Decision Making*, vol 8, no. 3, pp. 204 -215, 1988.

Received: *March 19, 2024;* ***Revised****: June 25, 2024;* ***Accepted:*** *June 30, 2024;* ***Published:*** *24 July 2024.*

Corresponding author- ***Anduamlak Abebe***