IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 30 October 2023, accepted 27 November 2023, date of publication 28 November 2023,
date of current version 6 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337645

RESEARCH ARTICLE

AE-Net: Novel Autoencoder-Based Deep Features for SQL Injection Attack Detection

NISREAN THALJI[1], ALI RAZA[2], MOHAMMAD SHARIFUL ISLAM[3], NAGWAN ABDEL SAMEE[4], AND MONA M. JAMJOOM[5]

[1]Department of Robotics and Artificial Intelligence, Jadara University, Irbid 21110, Jordan
[2]Institute of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan
[3]Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Chattogram 3814, Bangladesh
[4]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[5]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Mona M. Jamjoom (mmjamjoom@pnu.edu.sa)

This work was supported by Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, through Researchers Supporting under Project PNURSP2023R104.

ABSTRACT Structured Query Language (SQL) injection attacks represent a critical threat to database-driven applications and systems, exploiting vulnerabilities in input fields to inject malicious SQL code into database queries. This unauthorized access enables attackers to manipulate, retrieve, or even delete sensitive data. The unauthorized access through SQL injection attacks underscores the critical importance of robust Artificial Intelligence (AI) based security measures to safeguard against SQL injection attacks. This study's primary objective is the automated and timely detection of SQL injection attacks through AI without human intervention. Utilizing a preprocessed database of 46,392 SQL queries, we introduce a novel optimized approach, the Autoencoder network (AE-Net), for automatic feature engineering. The proposed AE-Net extracts new high-level deep features from SQL textual data, subsequently input into machine learning models for performance evaluations. Extensive experimental evaluation reveals that the extreme gradient boosting classifier outperforms existing studies with an impressive k-fold accuracy score of 0.99 for SQL injection detection. Each applied learning approach's performance is further enhanced through hyperparameter tuning and validated via k-fold cross-validation. Additionally, statistical t-test analysis is applied to assess performance variations. Our innovative research has the potential to revolutionize the timely detection of SQL injection attacks, benefiting security specialists and organizations.

INDEX TERMS Autoencoder optimization, deep learning, feature engineering, machine learning, SQL injection.

I. INTRODUCTION

SQL injection attacks are a prevalent and serious security threat in web applications and databases [1]. The attacker exploits vulnerabilities in software systems that interact with databases through SQL injection. This is achieved by injecting malicious SQL code into user-input fields, which is then executed by the application's database [2]. The consequences of a successful SQL injection attack can be severe, ranging from unauthorized access to sensitive information, alteration or deletion of critical data to even complete system compromise [3], [4]. These attacks can have far-reaching implications for individuals and organizations, potentially leading to financial losses, reputational damage, and legal consequences [5]. Therefore, understanding the mechanics of SQL injection attacks and implementing robust AI-based security measures is imperative in ensuring the integrity and confidentiality of databases and web applications.

SQL injection is a prominent vulnerability in web applications that arises from improper handling of user-provided input in SQL queries [6]. This attack method is insidious as it can bypass conventional security measures and directly target

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

the underlying database. Therefore, robust AI-based input validation mechanisms and parameterized queries are crucial countermeasures in thwarting SQL injection attacks [7]. SQL queries exhibit distinct properties that influence their performance and functionality. One crucial property is the complexity of the query, which pertains to the number of operations and joins involved in retrieving or modifying data. Subqueries or nested queries introduce a hierarchical structure, influencing the execution plan. An example SQL query for a database operation is:

```
SELECT id FROM logs WHERE nic = '1234';
```

AI-based machine learning methods have emerged as powerful tools in cybersecurity, particularly for detecting and preventing SQL injection attacks [8]. By leveraging advanced algorithms and deep learning techniques, AI models can autonomously learn patterns indicative of SQL injection attempts [9]. This empowers organizations to identify and mitigate potential threats proactively, bolstering their security posture and safeguarding sensitive information from unauthorized access or manipulation [10]. This research proposes a novel AE-Net approach for automatic feature engineering. The proposed AE-Net extracts high-level deep features from SQL textual query data, subsequently input into machine learning models for performance evaluations. The newly created features by the proposed AE-Net operate on the principle of encoding and decoding, allowing it to learn to represent data in a more compact and meaningful form.

Our primary research contributions for SQL injection attack detection are followed as:

- A novel optimized neural network, AE-Net, is proposed for automatic feature engineering, extracting new high-level deep features from SQL textual query data. The newly created feature set is then utilized for evaluating applied methods.
- Four advanced machine learning and deep learning-based approaches are employed for performance evaluations. Performance is further enhanced through hyperparameter tuning and validated via k-fold cross-validation. Statistical t-tests and computational complexity analysis are also applied to assess performance variations.

The remaining manuscript is structured as follows: Section II provides a comprehensive analysis of past studies applied to SQL injection detection. Section III outlines the workflow of the applied methods. Section IV presents the outcomes of the machine learning and deep learning approaches employed. Finally, we summarize our research findings in Section V.

II. LITERATURE ANALYSIS

This literature analysis section for detecting SQL injection queries with machine learning is a foundational component of this study. We comprehensively understand this domain's prevailing methodologies, techniques, and advancements by synthesizing and critiquing existing literature. This critical analysis provides context for our research approach and allows us to identify gaps, challenges, and opportunities that may inform the development of more effective detection strategies.

The study [11] focuses on SQL Injection Attacks, which pose a significant threat to the security and integrity of online applications utilizing databases. The analysis employs a dataset of SQL commands for conducting research experiments, utilizing a combination of supervised machine learning algorithms and a Convolutional Neural Network (CNN) model [12]. The reported accuracy of 0.97 was impressive, indicating strong detection capabilities using the proposed model. Nevertheless, subjecting the method to further scrutiny and validation in real-world situations is imperative to assess its robustness.

The study [13] addresses the critical issue of safeguarding web applications against SQL injection attacks. The researchers utilized a dataset of flow data from various SQL injection attacks on standard database engines and applied a range of machine learning algorithms to tackle the problem. While the model based on logistic regression achieved an impressive detection rate of over 97% with a false alarm rate of less than 0.07%, concerns arise regarding its overall effectiveness due to the absence of comprehensive performance metrics such as precision and recall. To determine the applicability of this approach beyond the mentioned database engines, it was essential to evaluate its practical usefulness. Further assessment and increased transparency in methodology are necessary to ascertain this approach's true viability and generalization in real-world settings.

In this [14] research, SQL injection detection was proposed. This study addresses the challenge of mitigating the complexity and timeliness issues associated with SQL injection attacks, which arise from variations in deployment environments and backend database engine syntax. The article discusses the application of machine learning algorithms. The proposed feature ratio method for extracting and detecting SQL injection attack payloads was also introduced. It is worth noting that the reported average f1 Score of 96.29% appears to be lower, indicating a potential decrease in accuracy and precision when compared to previous methods.

In [15], the authors address the ongoing issue of SQL Injection attacks in web applications, which can significantly compromise user data and manipulate database management systems, posing a significant cybersecurity threat. The research acknowledges the limitations of traditional rule-based methods and the constantly evolving nature of SQL attacks. The SQL query commands dataset was utilized for building models. The research proposed a hybrid CNN-BiLSTM approach to predict SQL attacks [16]. While the study reports a promising accuracy rate of 98%, it lacks critical information on precision, recall, and F1 score, making it challenging to assess its overall performance comprehensively. Additionally, the proposed model's generalizability across various attack scenarios or datasets was not thoroughly evaluated, which limits our confidence in its practicality.

TABLE 1. The analyzed literature summary analysis.

Ref.	Year	Dataset	Technique	Performance Accuracy
[11]	March 2023	SQL injection query data	Supervised machine learning algorithms and a CNN model	0.97
[13]	January 2023,	SQL injection query data	Logistic Regression	0.97
[14]	June 2023	The SQL injection attack payload	Feature ratio method on machine learning algorithms	0.97
[15]	April 2021	SQL Injection (SQLI) data	Hybrid CNN-BiLSTM	0.98
[17]	April 2021	SQL Injection (SQLI) data	Multilayer perceptron (MLP)	0.98
[18]	July 2023	Kaggle SQL Injection Dataset	Recurrent neural network	0.94
[19]	March 2023	CVE, CNVD, Exploit DB, and diasporic writers	SynBERT Neural network approach.	0.90
[20]	June 2023	Instagram datasets	Generative adversarial nets (APE_GAN++) , CNN-based IDS	0.76

This study [17] focuses on the critical issue of SQL Injection, which poses a significant threat to database-based applications across various platforms and devices. The research points out a common weakness in previous SQL injection models, which struggle to identify new attack patterns and rely heavily on past experiences or training data. In contrast, the proposed model demonstrates the ability to detect SQL injection by analyzing input patterns, a promising innovation. The model autonomously detects various injection techniques, with feature extraction and selection handled by the model, making it more user-friendly. The model's scalability across a range of applications is emphasized. The MPL model achieved an impressive cross-validated accuracy of 98%, with precision and recall scores of 98% and 97%, respectively, indicating strong detection capabilities.

The study [18] proposes a new method for detecting SQL injection attacks using a recurrent neural network. This research highlights the importance of addressing SQL injection's complex and evolving attack patterns. A publicly available Kaggle SQL Injection dataset was used in this study. The reported accuracy and f1-score of 94% and 92%, respectively, are considered poor compared to state-of-the-art studies.

This study [19] focuses on the persistent and damaging issue of SQL injection attacks on web applications, a critical problem listed in the open web application security project top 10. The complexity of detecting these attacks is acknowledged due to the variety of methods, patterns, and attack loads. To improve the accuracy and efficiency of detection, the study proposes synBERT, a semantic learning-based detection model. This model captures semantic information from SQL statements into embedding vectors. The study includes a diverse set of SQL datasets for evaluation and reports promising results with consistently high accuracy, even on untrained models, reaching 90%. However, the performance scores are lower in this study.

The focus of this study [20] is to create a trustworthy Intrusion Detection System (IDS) that utilizes convolutional neural networks (CNNs) to enhance network security and user safety online. However, the study acknowledges a significant drawback - susceptibility to adversarial attacks, which can compromise the model's accuracy. The research presents the development of two IDS models based on CNNs to improve accuracy and evaluates their reliability through seven adversarial attack scenarios. Although the reported accuracy improvements are 97.51% and 95.43%, the study lacks comprehensive performance metrics such as precision and recall, which limits the evaluation of the IDS's overall effectiveness. The observation that adversarial attacks cause drops in accuracy is a concern, but the applied defense method shows some recovery. Nonetheless, the accuracy scores after adversarial attacks, ranging from 78.12% to 89.40%, indicate that the defense method may be helpful but may not fully restore the model's reliability.

A. RESEARCH GAP

Through a comprehensive analysis of the existing literature, we have identified the following research gaps and limitations that we have addressed in our proposed research:

- Classical feature extraction approaches were employed to represent SQL query textual data.
- Also, classical machine learning-based techniques were used for SQL injection attack detection.
- Existing methods have high error rates when detecting SQL injection attacks.

III. PROPOSED METHODOLOGY

This section provides a comprehensive discussion of the materials and methods employed in our research experiments. We analyze the SQL queries dataset and the proposed feature engineering. Additionally, we utilize machine learning and deep learning techniques to evaluate the performance results.

Figure 1 illustrates our novel proposed methodology workflow for SQL injection attack detection. Initially, we acquired and prepared a benchmark dataset based on SQL query-based textual data for experimental evaluations. To capture high-level feature information from the dataset, we proposed a novel feature engineering approach. This proposed approach extracts high-level deep features and creates a new feature set. The newly created feature set is then divided into training and testing portions. The training set comprises 80% of the data, while the testing set comprises 20%. Next, we built advanced machine learning and deep

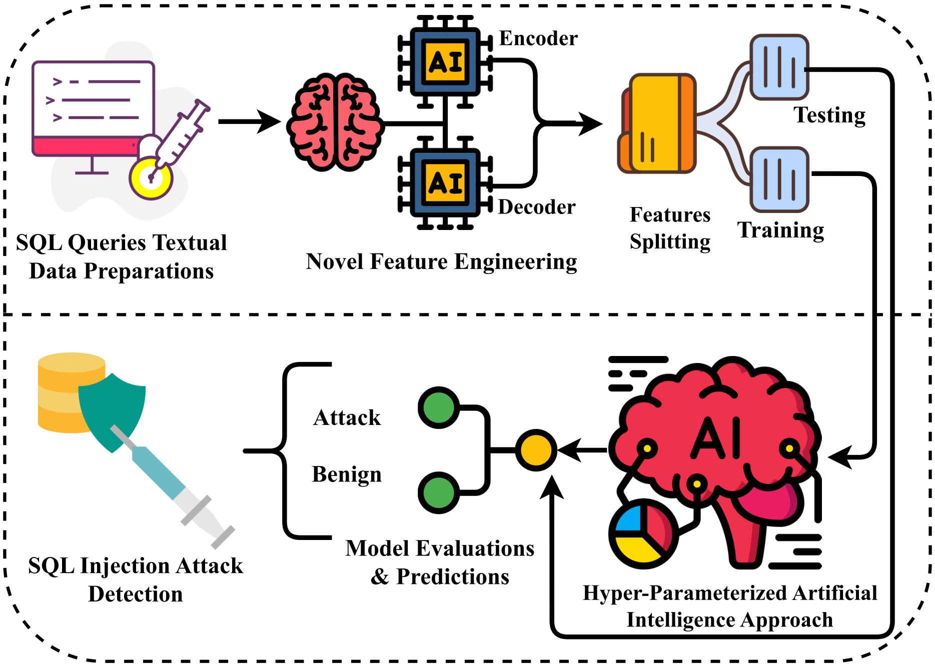

Encoder
Decoder
SQL Queries Textual Data Preparations
Novel Feature Engineering
Features Splitting
Testing
Training
Attack
Benign
Model Evaluations & Predictions
Hyper-Parameterized Artificial Intelligence Approach
SQL Injection Attack Detection
AI
AI
AI

FIGURE 1. The workflow architectural analysis of our proposed approach for detecting SQL injection attacks.

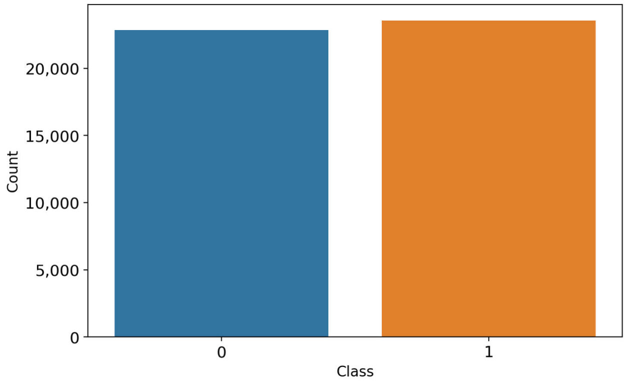


FIGURE 2. The target label distributions analysis.

learning models using the training datasets. The performance of each applied hyperparameterized approach is evaluated using unseen testing data. The AI approach that outperforms others with high-performance scores is then utilized for the detection of SQL injection attacks.

A. SQL QUERY TEXTUAL DATA

We utilized a publicly available SQL query dataset [21] for our research experiments. The dataset comprises three files: "sqli.csv," "sqliv2.csv," and "SQLiV3.csv," which we combined. For initial preprocessing, we encoded the target labels 'Attack' (1) and 'Benign' (0). The query textual data was converted into string format. We removed queries containing only two words. Additionally, null rows were dropped from the dataset. Finally, the preprocessed database of 46,392 SQL queries was used for further experiments. The target class distribution analysis of the dataset is depicted in Figure 2. The analysis reveals a nearly equal distribution of data, with 23,555 queries for the 'Attack' label and 22,837 queries for the 'Benign' label.

B. NOVEL PROPOSED FEATURE ENGINEERING

This section analyzes our novel proposed optimized neural network AE-Net, as illustrated in Figure 3. The analysis demonstrates by employing the unique autoencoder mechanism, our proposed approach extracts new high-level automatic deep features from SQL textual query data. The newly created feature set is subsequently employed to evaluate the applied methods. Additionally, the novel AE-Net layered architecture, which we propose, is analyzed and presented in Table 2.

The basic mathematical equation for an autoencoder method can be represented as follows:

$$h = f(Wx + b) \tag{1}$$

$$\hat{x} = g(Wh + c) \tag{2}$$

where:

x is the input vector

h is the hidden representation (feature vector)

$\hat{x}$ is the reconstructed input

f is the activation function for the encoder

g is the activation function for the decoder

W is the weight matrix

b is the bias vector for the encoder

c is the bias vector for the decoder

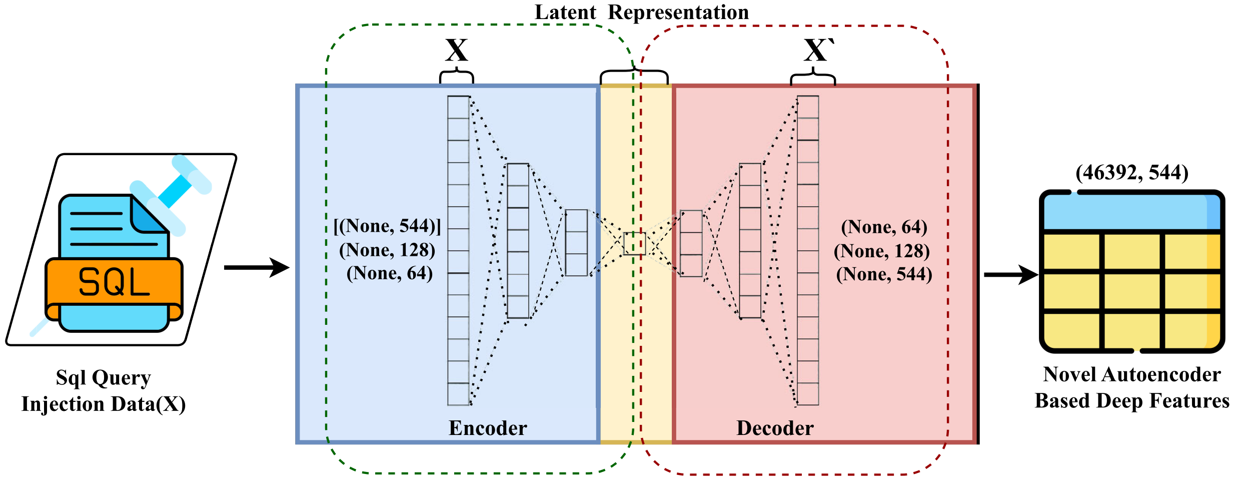


FIGURE 3. The architectural workflow analysis of novel proposed deep feature extraction approach.

The mathematical working mechanism of the proposed AE-Net approach for feature extraction in detecting SQL injection is represented as follows:

$$\text{Minimize:} \quad L(X, g(f(X)))$$
$$\text{Subject to:} \quad g(f(X)) = X$$

where:

X is the input data,

$f(\cdot)$ represents the encoder function,

$g(\cdot)$ represents the decoder function,

$L(\cdot)$ is the loss function.

TABLE 2. The novel proposed AE-Net layered architecture analysis.

Layer	Output Shape	Param #
Input Layer	[(None, 544)]	0
Encoder Dense Layer	(None, 128)	69760
Encoder Dense Layer	(None, 64)	8256
Decoder Dense Layer	(None, 128)	8320
Decoder Dense Layer	(None, 544)	70176
Trainable Params	156512 (611.38 KB)	

The algorithm 1 shows the step-by-step workflow of the proposed automatic feature engineering approach.

```
Algorithm 1 AE-Net Algorithm
Input: SQL query textual data.
Output: New rich level deep features.

initiate;
1- $F_{en} \longleftarrow En_{features}(Sd)$ // $Sd \in SQL\ queries$, here $Sd$ is original SQL data and $F_{en}$ is the deep encoded feature set.
2- $F_{de} \longleftarrow De_{features}(F_{en})$ // here $F_{en}$ is encoded features and $F_{de}$ is the deep decoded feature set.
3- $F_{featues} \longleftarrow F_{de}$ $//hereF_{featues}$ is final deep features set used for SQL injection detection.
end;
```

C. APPLIED ARTIFICIAL INTELLIGENCE APPROACHES

AI approaches, rooted explicitly in machine learning and deep learning techniques [22], have emerged as formidable tools in the realm of SQL injection attack detection. Machine learning models are trained to recognize patterns in data [23], making them adept at identifying anomalous SQL queries indicative of an attack. Techniques such as Support Vector Machines (SVM), Random Forests (RF), and k-Nearest Neighbors (KNN) have shown promise in this domain. Additionally, deep learning methods, notably Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTM), excel in capturing intricate relationships within sequences of SQL queries, enabling them to discern between legitimate and malicious requests effectively.

Deep learning-based feature engineering plays a pivotal role in extracting relevant information from raw SQL data to facilitate learning [24]. In this research, we employ autoencoders as unsupervised learning tools for extracting meaningful features, thereby enhancing the overall accuracy of SQL injection detection systems. The amalgamation of AI-driven techniques with robust feature engineering holds tremendous potential in fortifying defenses against SQL injection attacks, offering a proactive stance in safeguarding critical databases and applications from such security threats. The detailed working mechanisms of the applied methods are demonstrated in Table 3.

D. HYPERPARAMETER TUNING

Through the recursive process of training and evaluations, we have conducted hyperparameter tuning and determined the best-fit hyperparameter for each applied method. The final selected best-fit hyperparameters are demonstrated in Table 4. The best-fit hyperparameter helps us prevent overfitting issues and enhance SQL injection attack detection.

TABLE 3. The workflow mechanism analysis of applied machine learning and deep learning models for SQL injection detection.

Method	Working Description
KNC	K-Neighbors Classifier (KNC) method [25], [26] operates by classifying data points based on the majority class among their k-nearest neighbors in a feature space. In the context of SQL injection attack detection, it analyzes the characteristics of queries and compares them to a labeled dataset. By assessing the similarity to known attack patterns, KNC effectively distinguishes between benign and malicious SQL queries.
RF	Random Forest (RF) method [27], [28] for SQL injection attack detection operates by leveraging an ensemble of decision trees. Each tree evaluates different aspects of the input data, collectively forming a comprehensive assessment. Through a voting mechanism, the method combines these evaluations to make a final determination, enhancing accuracy and robustness in identifying potential SQL injection threats.
LR	Logistic Regression (LR) [29], [30] operates by modeling the relationship between a set of input features and the probability of a binary outcome, typically classifying queries as either benign or indicative of an SQL injection attack. It jain2023heartestimates coefficients for each feature, enabling the creation of a decision boundary, and when new queries are inputted, LR calculates the probability of belonging to each class and assigns the query to the one with the higher probability, thus facilitating SQL injection attack detection.
XGB	Extreme Gradient Boosting (XGB) [31], [32] classifier method operates by iteratively refining the predictions of an ensemble of weak learners, typically decision trees. It focuses on minimizing errors from the previous iterations, effectively creating a robust and accurate model. In the context of SQL injection attack detection, XGB leverages this iterative process to discern patterns in SQL queries
LSTM	Long Short-Term Memory (LSTM) [33]–[35] method employs a specialized recurrent neural network architecture that excels in processing sequences of data, making it adept for analyzing SQL queries. By leveraging its memory cells, LSTM can effectively discern intricate patterns within the queries, distinguishing between legitimate and potentially malicious requests.

TABLE 4. The hyperparameter tuning analyis of applied methods.

Method	Hyperparameter Description
KNC	n_neighbors=2, weights='uniform', leaf_size=30, metric='minkowski', p=2
LR	penalty='l2', max_iter=6, tol=1e-4, C=1.0
RF	n_estimators=100, max_depth=20, criterion="gini", max_features="sqrt", min_samples_split=2
XGB	n_estimators=30, max_depth=30
LSTM	activation='sigmoid', epochs=20, batch_size=64, loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy']

IV. RESULTS AND DISCUSSIONS

The research study's results and discussions section focuses on detecting SQL injection queries across multiple websites using machine learning techniques. We have provided a comprehensive overview of the outcomes achieved through our research experiments. Additionally, we present a detailed analysis of the performance of various machine learning methods employed to distinguish between signs of an SQL attack and benign queries.

A. EXPERIMENTAL SETUP

Our novel proposed research experiments were conducted on a machine equipped with the following specifications: an Intel(R) Core(TM) i5-10300H CPU, 16.0 GB RAM, 2.50 GHz CPU, 8MB cache size, and a Core i5-10300H model name for the CPU. The Python 3.0 programming language was employed to implement the applied machine learning and deep learning methods. The Python library modules used during method implementations were Sklearn, Keras, and TensorFlow. The performance measures are the runtime computation, p-value, accuracy, f1, recall, and precision.

1) EVALUATION METRICS

This study employs several well-known performance evaluation metrics to evaluate the performance of machine learning models. In particular, accuracy, precision, recall, and F1 score are used with the following equations:

$$Accuracy = \frac{TP + FP}{TP + FP + TN + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

where, TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative, respectively.

B. PERFORMANCE ANALYSIS WITH BOW FEATURES

The performance of applied machine learning approaches with Bag of Words (BoW) features is evaluated in Table 5. BoW is an effective natural language processing (NLP) technique that simplifies text into word-frequency vectors. To detect SQL attack queries, we have implemented KNC, LR, RF, and XGB approaches using BoW features. The analysis reveals that the KNC model achieved the highest accuracy score of 0.94 in comparisons. The RF method achieved the lowest performance accuracy score of 0.89, followed by XGB and LR. The KNC excelled in accurately identifying the "Attack" class, with a high precision of 0.90 and perfect recall. These results demonstrate that both KNC and LR methods achieved acceptable scores; however, they are not the highest. There is still a need for advanced feature engineering to enhance the performance of SQL injection attack detection.

C. PERFORMANCE ANALYSIS WITH TFIDF FEATURES

After evaluating the results with the BoW feature, we implemented a more advanced approach, Term Frequency-Inverse Document Frequency (TFIDF). TFIDF features assess the significance of words in textual data by comparing their frequency in a specific document to their frequency in the entire dataset. The performance scores of the applied methods are outlined in Table 6. The analysis revealed

TABLE 5. Performance results of machine learning methods with BoW features.

Method	Accuracy	Target Class	Precision	Recall	F1
KNC	0.94	Attack	0.90	1.00	0.95
		Benign	0.99	0.89	0.94
		Average	0.95	0.94	0.94
LR	0.91	Attack	0.91	0.92	0.92
		Benign	0.92	0.91	0.91
		Average	0.92	0.92	0.92
RF	0.89	Attack	0.98	0.80	0.88
		Benign	0.83	0.98	0.90
		Average	0.91	0.89	0.89
XGB	0.90	Attack	0.95	0.87	0.91
		Benign	0.88	0.95	0.91
		Average	0.91	0.91	0.91

that, once again, the KNC approach demonstrated strong classification ability with an accuracy rate of 0.96. The classification-wise metric scores are also notably high. The other applied methods, LR, RF, and XGB, also displayed improved performance scores compared to those with BoW features. This analysis demonstrates that by employing a more advanced feature engineering approach, we are able to enhance performance scores for the attack section. Nevertheless, a 4% error remains, indicating a necessity for advanced neural network-based feature engineering.

TABLE 6. Performance results of machine learning methods with TFIDF features.

Method	Accuracy	Target Class	Precision	Recall	F1
KNC	0.96	Attack	0.99	0.92	0.96
		Benign	0.93	0.99	0.96
		Average	0.96	0.96	0.96
LR	0.95	Attack	0.96	0.94	0.95
		Benign	0.94	0.96	0.95
		Average	0.95	0.95	0.95
RF	0.93	Attack	0.99	0.86	0.92
		Benign	0.88	1.00	0.93
		Average	0.94	0.93	0.93
XGB	0.94	Attack	0.96	0.92	0.94
		Benign	0.92	0.96	0.94
		Average	0.94	0.94	0.94

D. PERFORMANCE ANALYSIS WITH NOVEL PROPOSED FEATURES

The results obtained from the applied BoW and TFIDF feature engineering approaches are not up to the mark. In order to address this issue, we have introduced a novel feature engineering approach named AR-Net. With this proposed approach, we have extracted high-level deep features from the SQL query data and re-evaluated the results in this section.

The results of the applied machine learning and deep learning models, along with the proposed feature engineering approach, are described in Table 7. The applied KNC and RF achieved excellent performance scores of 0.96 and 0.99, respectively. The machine learning-based XGB approach outperformed others with a remarkable accuracy score of 0.99. The deep learning-based LSTM method achieved an acceptable score of 0.87; however, it is not the highest compared to the other methods. This analysis concludes that the novel features effectively capture relevant information and enable accurate detection of SQL injection attacks. The proposed novel features appear to offer higher accuracy and more balanced precision and recall rates compared to BOW and TFIDF representations.

TABLE 7. Performance results of machine learning and deep learning methods with novel deep features.

Method	Accuracy	Target Class	Precision	Recall	F1
KNC	0.96	Attack	0.98	0.95	0.97
		Benign	0.95	0.98	0.97
		Average	0.97	0.97	0.97
RF	0.99	Attack	0.98	0.99	0.99
		Benign	0.99	0.98	0.98
		Average	0.99	0.99	0.99
XGB	**0.99**	**Attack**	**0.99**	**0.99**	**0.99**
		Benign	**0.99**	**0.99**	**0.99**
		Average	**0.99**	**0.99**	**0.99**
LSTM	0.87	Attack	0.88	0.87	0.87
		Benign	0.87	0.88	0.87
		Average	0.87	0.87	0.87

The time series-based results analysis of the deep learning model LSTM during training is illustrated in Figure 4. The analysis demonstrates that during the first five epochs of training, the loss scores are high and performance accuracy scores are below 0.80. After the fifth epoch, the LSTM neural network sets optimal weights and gradually improves performance scores. The analysis shows that the LSTM approach achieved accuracy scores above 0.85 for SQL injection detection.

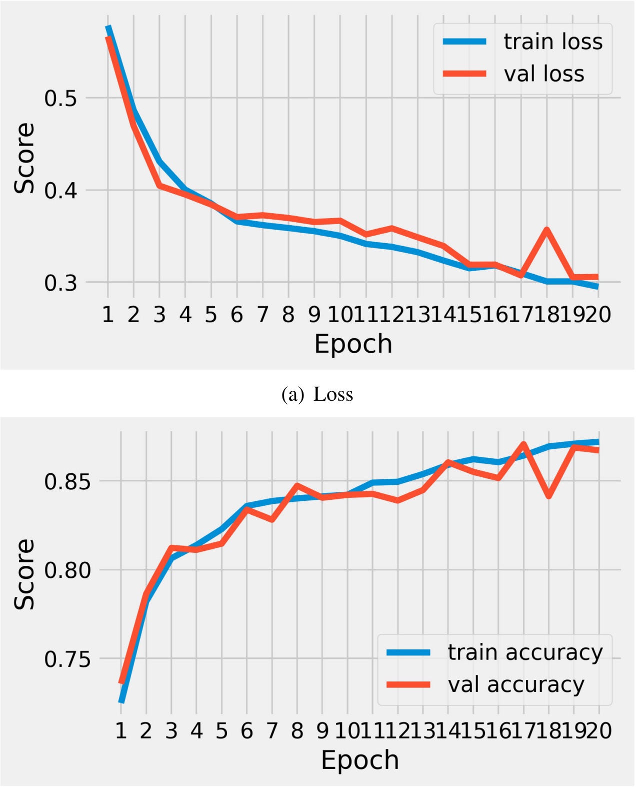
The performance variations of applied methods based on the confusion matrix are demonstrated in Figure 5. This analysis reveals that LSTM achieved a high wrong prediction rate, which is the reason for the low performance scores. The machine learning-based KNC also achieved high error rates during classification, followed by RF. This analysis concludes that the proposed XGB approach achieved a minimum error rate of 102 wrong classifications for SQL injection, ensuring high performance scores.

E. KFOLD VALIDATIONS ANALYSIS

The K-fold cross-validation technique is a popular method for evaluating the performance of machine learning models. Table 8 contains the cross-validation results of the applied method. During cross-validation, the data splitting process is repeated 10 times, and the average results are analyzed to determine the model's effectiveness. Assessing the Standard Deviation (SD) is essential in determining the reliability and consistency of the outcomes. The analysis shows that only LSTM achieved low K-fold scores, followed by the KNC method. This analysis reveals that the proposed XGB approach achieved high k-fold accuracy performance scores of 0.99 with a minimum SD score of 0.0010 for SQL injection attack detection.

F. COMPUTATIONAL COMPLEXITY ANALYSIS

We have determined the computational complexity of each applied model, and the results are reported in Table 9. This



(a) Loss

(b) Accuracy

FIGURE 4. The time series-based results analysis of deep learning model LSTM during training.

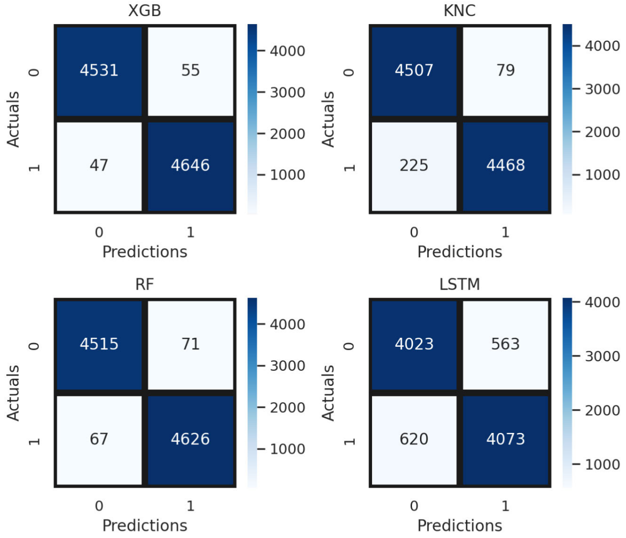


FIGURE 5. The confusion matrix analysis of applied neural network approaches.

section contains the runtime computations for each model during the building process on the proposed feature dataset. Upon analysis, it is discovered that the deep learning-based LSTM method is associated with high complexity rates of 133.7 seconds. The proposed XGB also achieved high runtime complexity; however, it also achieved high performance scores for SQL injection detection. As a result, we consider high performance scores as a priority due to overcome critical SQL attacks. Furthermore, we will work on reducing the complexity by optimizing the proposed approach's architecture.

TABLE 8. Performance results validation of machine learning and deep learning methods with novel deep features.

Method	K-folds	Accuracy	SD(+/-)
KNC	10	0.97	0.0020
RF	10	0.98	0.0019
XGB	**10**	**0.99**	**0.0010**
LSTM	10	0.83	0.0256

TABLE 9. Computational complexity analysis of machine learning and deep learning methods with novel deep features.

Method	Runtime Computations (Seconds)
KNC	0.967
RF	14.99
XGB	23.66
LSTM	133.7

G. COMPARISON WITH STATE OF THE ART APPROACHES

We have compared our novel proposed approach's performance with state-of-the-art studies published for SQL injection detection. Table 10 contains the comparison results. We have used recent studies from the year 2023 for comparison. The analysis reveals that our proposed approach outperformed the state-of-the-art approach with high accuracy performance of 0.99 for SQL injection attack detection.

TABLE 10. The results comparison with state of the art studies used for SQL injection attack detection.

Ref.	Year	Proposed Technique	Performance Accuracy
[11]	2023	Convolutional Neural Network	0.97
[13]	2023	Logistic Regression	0.97
[18]	2023	Recurrent neural network	0.94
Our	**2024**	**AE-Net + XGB**	**0.99**

H. STATISTICAL T-TEST ANALYSIS

In this section, we provide the outcomes of a statistical t-test conducted on the results of our proposed model, utilizing all the incorporated features [36]. This test assesses the significance of the compared approach. It establishes two hypotheses: the null hypothesis posits that the compared approach lacks statistical significance in comparison to others. Should the t-test refute the null hypothesis, it would imply acceptance of the alternative hypothesis, signifying the statistical significance of our proposed approach.

The t-test produces a Statistic Score and a corresponding p-value. When the p-value surpasses the Statistic value, it leads to the rejection of the null hypothesis. Table 11 presents the outcomes across various scenarios. We conducted a comparative analysis of the results obtained from the machine learning model using the suggested methodology against other features. In every instance of comparison, the t-test dismisses the null hypothesis, providing evidence of the statistical significance of the proposed approach.

TABLE 11. The statistical analysis using the t-test mechanism.

Case	Statistic Score	P-Value	Null Hypothesis
Proposed XGB with BoW features vs. novel deep features	-32.7367	5.4041	Rejected
Proposed XGB with TFIDF features vs. novel deep features	-199.00	1.0864	Rejected

I. DISCUSSIONS

The automated and timely detection of SQL injection attacks through AI methods is performed in experiments. We have used a preprocessed database of 46,392 SQL queries. We introduce a novel approach, the AE-Net, for automatic feature engineering. The proposed AE-Net extracts high-level deep features from SQL textual data, subsequently input into machine learning models for performance evaluations. During results evaluations, we have applied four advanced machine learning and deep learning-based approaches for performance assessments. Extensive experimental evaluation reveals that the extreme gradient boosting classifier outperforms existing studies with an impressive k-fold accuracy score of 0.99 for SQL injection detection. The performance of each applied learning approach is further enhanced through hyperparameter tuning and validated via k-fold cross-validation. Additionally, statistical t-test analysis is applied to assess performance variations.

V. CONCLUSION AND FUTURE WORK

This study proposed an automated and timely method for the detection of SQL injection attacks. We introduce a novel approach called AE-Net for automatic feature engineering. AE-Net extracts high-level deep features from SQL textual data, which are subsequently input into machine learning models for performance evaluation. Four advanced machine learning and deep learning-based approaches are employed for these evaluations. Extensive experimental evaluation reveals that the extreme gradient boosting classifier outperforms existing studies with an impressive K-fold accuracy score of 0.99 for SQL injection detection. The performance of each applied learning approach is further enhanced through hyperparameter tuning and validated via K-fold cross-validation. Additionally, statistical t-test analysis is applied to assess performance variations.

A. FUTURE WORK

In the future, we plan to revise the architecture of our proposed approach to reduce computational complexity. Additionally, we intend to develop a user-friendly graphical interface tailored for networking organizations, enabling real-time monitoring of SQL injection attacks.

FUNDING

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R104), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ACKNOWLEDGMENT

The authors would like to express their grateful to Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R104), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, "A systematic review of detection and prevention techniques of SQL injection attacks," *Inf. Secur. J., Global Perspective*, vol. 32, no. 4, pp. 252–265, Jul. 2023.

[2] I. S. Crespo-Martínez, A. Campazas-Vega, Á. M. Guerrero-Higueras, C. Álvarez-Aparicio, and C. Fernández-Llamas, "Impact of the keep-alive parameter on SQL injection attack detection in network flow data," in *Proc. Comput. Intell. Secur. Inf. Syst. Conf.* Cham, Switzerland: Springer, 2023, pp. 69–78.

[3] A. Arshad, M. Jabeen, S. Ubaid, A. Raza, L. Abualigah, K. Aldiabat, and H. Jia, "A novel ensemble method for enhancing Internet of Things device security against botnet attacks," *Decis. Anal. J.*, vol. 8, Sep. 2023, Art. no. 100307.

[4] F. Rustam, A. Raza, I. Ashraf, and A. D. Jurcut, "Deep ensemble-based efficient framework for network attack detection," in *Proc. 21st Medit. Commun. Comput. Netw. Conf. (MedComNet)*, Jun. 2023, pp. 1–10.

[5] R. Madhvan and M. F. Zolkipli, "An overview of malware injection attacks: Techniques, impacts, and countermeasures," *Borneo Int. J. eISSN*, vol. 6, no. 3, pp. 22–30, 2023.

[6] T. Sheth, J. Anap, H. Patel, N. Singh, and R. R. B, "Detection of SQL injection attacks by giving *a priori* to Q-learning agents," in *Proc. IEEE IAS Global Conf. Emerg. Technol. (GlobConET)*, May 2023, pp. 1–6.

[7] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL injection attack using machine learning techniques: A systematic literature review," *J. Cybersecur. Privacy*, vol. 2, no. 4, pp. 764–777, Sep. 2022.

[8] P. Roy, R. Kumar, and P. Rani, "SQL injection attack detection by machine learning classifier," in *Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, May 2022, pp. 394–400.

[9] A. M. A. Badri and S. Alouneh, "Detection of malicious requests to protect web applications and DNS servers against SQL injection using machine learning," in *Proc. Int. Conf. Intell. Comput., Commun., Netw. Services (ICCNS)*, Jun. 2023, pp. 5–11.

[10] K. Singh, S. Kokardekar, G. Khonde, P. Dekate, N. Badkas, and S. Lachure, "Cloud engineering-based on machine learning model for SQL injection attack," in *Proc. Int. Conf. Commun., Circuits, Syst. (IC3S)*, May 2023, pp. 1–6.

[11] J. Misquitta and S. Asha, "SQL injection detection using machine learning and convolutional neural networks," in *Proc. 5th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Jan. 2023, pp. 1262–1266.

[12] A. Rehman, A. Raza, F. S. Alamri, B. Alghofaily, and T. Saba, "Transfer learning-based smart features engineering for osteoarthritis diagnosis from knee X-ray images," *IEEE Access*, vol. 11, pp. 71326–71338, 2023.

[13] I. S. Crespo-Martínez, A. Campazas-Vega, Á. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Álvarez-Aparicio, and C. Fernández-Llamas, "SQL injection attack detection in network flow data," *Comput. Secur.*, vol. 127, Apr. 2023, Art. no. 103093.

[14] S. Zhang, Y. Li, and Q. Jiang, "Feature ratio method: A payload feature extraction and detection approach for SQL injection attacks," in *Proc. 3rd Asia–Pacific Conf. Commun. Technol. Comput. Sci. (ACCTCS)*, Feb. 2023, pp. 172–175.

[15] N. Gandhi, J. Patel, R. Sisodiya, N. Doshi, and S. Mishra, "A CNN-BiLSTM based approach for detection of SQL injection attacks," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Mar. 2021, pp. 378–383.

[16] A. Raza, F. Rustam, H. U. R. Siddiqui, I. D. L. T. Diez, and I. Ashraf, "Predicting microbe organisms using data of living micro forms of life and hybrid microbes classifier," *PLoS ONE*, vol. 18, no. 4, Apr. 2023, Art. no. e0284522.

[17] K. R. Jothi, S. Balaji B, N. Pandey, P. Beriwal, and A. Amarajan, "An efficient SQL injection detection system using deep learning," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Mar. 2021, pp. 442–445.

[18] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Deep learning architecture for detecting SQL injection attacks based on RNN autoencoder model," *Mathematics*, vol. 11, no. 15, p. 3286, Jul. 2023.

[19] D. Lu, J. Fei, and L. Liu, "A semantic learning-based SQL injection attack detection technology," *Electronics*, vol. 12, no. 6, p. 1344, Mar. 2023.

[20] A. Alotaibi and M. A. Rassam, "Enhancing the sustainability of deep-learning-based network intrusion detection classifiers against adversarial attacks," *Sustainability*, vol. 15, no. 12, p. 9801, Jun. 2023.
[21] S. S. H. Shah. *SQL Injection Dataset | Kaggle*. Accessed: Oct. 9, 2023. [Online]. Available: https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset
[22] A. Raza, A. M. Qadri, I. Akhtar, N. A. Samee, and M. Alabdulhafith, "LogRF: An approach to human pose estimation using skeleton landmarks for physiotherapy fitness exercise correction," *IEEE Access*, vol. 11, pp. 107930–107939, 2023.
[23] A. Raza, M. R. Al Nasar, E. S. Hanandeh, R. A. Zitar, A. Y. Nasereddin, and L. Abualigah, "A novel methodology for human kinematics motion detection based on smartphones sensor data using artificial intelligence," *Technologies*, vol. 11, no. 2, p. 55, Apr. 2023.
[24] G. Kaur and A. Sharma, "A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis," *J. Big Data*, vol. 10, no. 1, p. 5, Jan. 2023.
[25] N. Puri, P. Saggar, A. Kaur, and P. Garg, "Application of ensemble machine learning models for phishing detection on web networks," in *Proc. 5th Int. Conf. Comput. Intell. Commun. Technol. (CCICT)*, Jul. 2022, pp. 296–303.
[26] R. Hajizadeh, "Unconstrained neighbor selection for minimum reconstruction error-based K-NN classifiers," *Complex Intell. Syst.*, vol. 9, no. 5, pp. 5715–5730, Oct. 2023.
[27] M. Imran, H. U. R. Siddiqui, A. Raza, M. A. Raza, F. Rustam, and I. Ashraf, "A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems," *Comput. Secur.*, vol. 134, Nov. 2023, Art. no. 103445.
[28] A. M. Qadri, A. Raza, F. Eid, and L. Abualigah, "A novel transfer learning-based model for diagnosing malaria from parasitized and uninfected red blood cell images," *Decis. Anal. J.*, vol. 9, Dec. 2023, Art. no. 100352.
[29] L. J. M. León, J. L. Herrera, J. Berrocal, and J. Galán-Jiménez, "Logistic regression-based solution to predict the transport assistant placement in SDN networks," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, May 2023, pp. 1–5.
[30] V. Jain and M. Agrawal, "Heart failure prediction using XGB classifier, logistic regression and support vector classifier," in *Proc. Int. Conf. Advancement Comput. Comput. Technol. (InCACCT)*, May 2023, pp. 1–5.
[31] S. Akbar, H. Ali, A. Ahmad, M. R. Sarker, A. Saeed, E. Salwana, S. Gul, A. Khan, and F. Ali, "Prediction of amyloid proteins using embedded evolutionary & ensemble feature selection based descriptors with eXtreme gradient boosting model," *IEEE Access*, vol. 11, pp. 39024–39036, 2023.
[32] A. Raza, F. Rustam, B. Mallampati, P. Gali, and I. Ashraf, "Preventing crimes through gunshots recognition using novel feature engineering and meta-learning approach," *IEEE Access*, vol. 11, pp. 103115–103131, 2023.
[33] S. Das, A. Paramane, S. Chatterjee, and U. M. Rao, "Sensing incipient faults in power transformers using bi-directional long short-term memory network," *IEEE Sensors Lett.*, vol. 7, no. 1, pp. 1–4, Jan. 2023.
[34] A. Raza, K. Munir, M. S. Almutairi, and R. Sehar, "Novel class probability features for optimizing network attack detection with machine learning," *IEEE Access*, vol. 11, pp. 98685–98694, 2023.
[35] S. Aziz, K. Munir, A. Raza, M. S. Almutairi, and S. Nawaz, "IVNet: Transfer learning based diagnosis of breast cancer grading using histopathological images of infected cells," *IEEE Access*, vol. 11, pp. 127880–127894, 2023.
[36] F. Rustam, A. Ishaq, M. S. A. Hashmi, H. U. R. Siddiqui, L. A. D. López, J. C. Galán, and I. Ashraf, "Railway track fault detection using selective MFCC features from acoustic data," *Sensors*, vol. 23, no. 16, p. 7018, Aug. 2023.

NISREAN THALJI received the bachelor's and master's degrees (Hons.) in computer science Yarmouk University, Jordan, Irbid, and the Ph.D. degree (Hons.) in computer science, specializing in artificial intelligence and data science from the prestigious University Malaysia Perlis (UNIMAP), Perlis, Malaysia. With a strong educational background, she is currently an Assistance Professor with Jadara University. She is an accomplished individual in the field of computer science. Her diverse expertise spans across various domains, including artificial intelligence, machine learning, deep learning, algorithm engineering, swarm intelligence, and natural language processing. Her research interests continue to thrive in these cutting-edge areas, further solidifying her reputation as a dedicated Scholar and a Researcher in the field.

ALI RAZA received the Bachelor of Science and M.S. degrees in computer science from the Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan, in 2021 and 2023, respectively. He has published several articles in reputed journals. His current research interests include data science, artificial intelligence, data mining, natural language processing, machine learning, deep learning, and image processing.

MOHAMMAD SHARIFUL ISLAM received the B.Sc. degree in computer science and telecommunication engineering from the Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh, in 2023. His current research interests include data science, machine learning, natural language processing, and image processing.

NAGWAN ABDEL SAMEE received the B.S. degree in computer engineering from Ein Shams University, Egypt, in 2000, and the M.S. degree in computer engineering and the Ph.D. degree in systems and biomedical engineering from Cairo University, Egypt, in 2008 and 2012, respectively. Since 2013, she has been an Assistant Professor with the Information Technology Department, CCIS, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include data science, machine learning, bioinformatics, and parallel computing. Her awards and honors include the Takafull Prize (Innovation Project Track), Princess Nourah Award in Innovation, Mastery Award in Predictive Analytics (IBM), the Mastery Award in Big Data (IBM), and the Mastery Award in Cloud Computing (IBM).

MONA M. JAMJOOM received the Ph.D. degree in computer science from King Saud University. She is currently an Associate Professor with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include artificial intelligence, machine learning, deep learning, medical imaging, and data science. She has published several research articles in her field.

• • •