Cyber Security Risk Assessment Framework for Cloud Customer and Service Provider

N. Sujata Kumari[1]*, Naresh Vurukonda[2]
Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India, 522502[1]
Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India, 522502[2]

***Abstract*—The rapid development of cloud computing demands an effective cybersecurity framework for protecting the sensitive information of the infrastructure. Currently, many organizations depend on cloud services for their operation, increasing the risk of cybersecurity. Hence, an intelligent risk assessment mechanism is significant for detecting and mitigating the cybersecurity threats associated with cloud environments. Although various risk assessment methods were developed in the past, they lack the efficiency to handle the dynamic and evolving nature of threats. In this study, we proposed an innovative framework for cybersecurity risk assessment in cloud customers and service providers. Initially, the historical cloud customer and service provider database was collected and fed into the system. The collected dataset contains historical security risks, network traffic, system behavior, etc., and the accumulated dataset was pre-processed to improve the quality of the dataset. The data pre-processing steps not only ensure quality but also transform the dataset into appropriate format for subsequent analysis. Further, a risk assessment module was created using the combination of deep recurrent neural network with krill herd optimization (DRNN-KHO) algorithm. In this module, the DRNN was trained using the pre-processed database to learn the pattern and interconnection between normal and abnormal network traffic. Subsequently, the KHO refines the DRNN parameters in its training phase, increasing the efficiency of risk assessment. This integrated module ensures adaptability to the system, leading to accurate prediction of evolving security threats. Then, a secure data exchange protocol was created for secure transmission between cloud customer and service provider. This protocol is designed by integrating artificial bee colony optimization with the elliptic curve cryptography (ABC-ECC). Thus, this collaborative framework ensures security in the cloud customer and service providers.**

Keywords*—Deep recurrent neural network; krill herd optimization; artificial bee colony optimization; elliptic curve cryptography

I. Introduction

In recent years, the significance of cloud computing (CC) has been rising, and it has received a growing interest in both business and scientific organizations [1]. The report from the National Institute of Standards and Technology (NIST) stated that the CC model offers configurable resources such as storage, networks, servers, applications, etc., in the most convenient and on-demand manner [2]. The CC delivers these resources with various kinds of service provider interaction by following a simple Pay-As-You-Go (PAYG) model. In the PAYG model, the cloud service consumers (CSC) demand the computing services for their business demands [3]. Consequently, the cloud service provider (CSP) delivers the demanded computing services to the CSC. In the CC model, the CSC has to pay only for the services they have utilized, reducing the cost and making the process more convenient [4]. In addition to this, the CC model has several advantages such as greater scalability, high availability, enhanced flexibility, well-documented, excellent reliability, etc., [5]. These advantages bring more benefits to the business organizations. In accordance with the report by Forbes, the CC has imposed a greater growth, and the CC market reached $411 Billion in 2020 [6]. Furthermore, the survey made by the LogicMonitor in 2020 illustrated that the cloud services landscape has gained 83% interest of the enterprises [7].

Although the CC brings more benefits to the organizations, it offers several issues. One among the issues is the cybersecurity threats, which degrades confidence and feasibility of the CC model [8]. From the perspective of CSC, the primary reason for confidence issues on CC is because of its multitenancy nature, sensitive data transformation, critical infrastructure and applications [9]. On the other hand, security is the primary issue from the perspective of CSP, which is mainly because of the cloud model's complexity. This complexity arises from the management of diverse security controls and measures [10]. Also, the security threats arise while transferring the data in the CC. These security threats induce distrust and fears in the business organizations, making them to redefine their decisions in adapting the CC model [11]. Despite how strongly the CC is secured, the business units suffer from trust issues on cloud models and remain uncertain about its economic feasibility [12].

While it's unrealistic to guarantee a zero-risk service, implementing an efficient security risk management algorithm can significantly enhance organizations' confidence in the CC model [13]. This risk management technique empowers organizations to make informed decisions about adopting this emerging technology [14]. The conventional risk management algorithms do not fit into the CC model because of the inherent assumptions made by them [15]. In recent decades, artificial intelligence (AI) techniques are widely used in different research fields because of its capacity to decide like humans [16]. The AI evolution paved a way for handling the security risks in the CC model. The utilization of AI techniques such as deep learning (DL) and machine learning (ML) techniques

provides an effective way of identifying the potential security risks in CC [17]. Although they offer better security compared to the conventional models, they face certain challenges like overfitting, large computational time, huge computational resource demand, lack of scalability, less adaptability, lower reliability, etc., [18]. Moreover, they cannot provide security to the sensitive data during transmission in the CC model [19]. This demands an optimal and reliable security risk management algorithm, which must potentially identify the cybersecurity risks, and provide greater protection to data in the cloud model [20]. By evaluating risks across different dimensions such as data sensitivity, regulatory compliance, access controls, and service-level agreements (SLAs), the framework enables both customers and providers to identify weak points and implement appropriate mitigation strategies. To resolve the above issues, we proposed a collaborative security risk assessment framework combining deep learning, meta-heuristic optimizations, and cryptographic algorithms. The main contributions of the work are described as follows.

- This study proposes a collaborative security risk assessment framework by integrating the efficiencies of deep learning, meta-heuristic optimizations, and cryptographic algorithms for effectively identifying the cybersecurity threats and securely transmitting the data in the cloud model.
- The developed framework creates a cybersecurity threat detection module named DRNN-KHO by integrating Krill Herd optimization and Deep Recurrent Neural Network, which is trained using the cloud security database to identify the normal and malicious activities in the cloud model.
- The study also developed a hybrid ABC-ECC model, which is designed by incorporating artificial bee colony optimization into the elliptic curve cryptographic algorithm for securely transmitting the sensitive data in the cloud model.
- The presented collaborative mechanism was implemented in the Python language, and the results are assessed and validated using metrics like attack detection accuracy, data confidential rate, computational time, encryption time, decryption time, etc.

The enduring sections of the article are organized as follows: Section II illustrates the literature survey, Section III depicts the system model and problem statement, Section IV explains the proposed algorithm, Section V analyzes the study results, and Section VI provides the study conclusion.

II. Related Works

A. Few Recent Studies Related to the Developed Framework are Described Below:

Lav Gupta et al. [21] presented a deep hierarchical stacked neural system for precise prediction of anomaly activities in the cloud environment. This study aims to resolve the security problems associated with the healthcare applications, as the malicious agents threaten the patient's life and health by changing their medical data flow into the healthcare networks. The experimental results depict that this DL model minimized the training time by 26.2%, and achieved better convergence. Moreover, this algorithm predicts the malicious agents with an average accuracy ranging from 93% to 95%. However, this framework offers limited generalization and is vulnerable to adversarial attacks.

Reem Al Saleh et al. [22] proposed a reputation-based trust evaluation algorithm by integrating Net Brand Reputation (NBR) with the deep learning approach for the cloud market. The DL algorithm used in this approach is CBiLSTM, which is the combination of Convolutional Neural Networks (CNN), and Bidirectional Long Short-Term Memory (BiLSTM). The implementation outcomes manifest that this algorithm obtained performances of 96.7% accuracy, 97% f-measure, 96.5% recall and 97.4% recall. In addition to this, this model consumed a minimum training time of 519ms. Although this approach offered improved performances, it is less scalable and interpretable.

R. Denis and P. Madhubala [23] developed a hybrid encryption algorithm for ensuring confidentiality in the cloud computing environment. This hybrid strategy combines Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman for secure data transmission. Also, an adaptive genetic algorithm was applied in the proposed work to ensure adaptability to the changing CC environment. This adaptiveness enables the system to respond to the emerging threats in the CC model. Finally, the simulation outcomes validate that this framework obtained data confidential rate of 0.95, and minimum running time of 5.4s. However, it faces complexity in managing keys, resulting in increased computational overhead.

Fursan Thabit et al. [24] developed an innovative lightweight cryptographic algorithm for improving the data security in the CC model. This approach is designed based on feistal and substitution permutation architectural approaches to enhance the encryption complexity. In addition, this approach obtained Shannon's theory of diffusion and confusion by including logical functions like XOR, shifting, XNOR, and swapping. The experimental outcomes suggest that this algorithm achieved a strong security level and less execution time compared to the existing cryptographic techniques. However, it is less adaptable for emerging threats.

P. Chinnasamy et al. [25] developed an efficient data security algorithm using hybrid cryptographic techniques for the CC model. This hybrid algorithm combines Elliptic curve cryptography (ECC) and Blowfish to provide a high level of data transfer and storage security. The main objective of this algorithm is to address the security challenges like confidentiality, availability, and integrity. This work was validated in the medical cloud environment, and the experimental data depicts that it achieved higher security and confidentiality than the symmetric and asymmetric algorithms.

Shaopeng Guan et al. [26] designed a distinct framework named Hadoop-assisted big data storage scheme for the CC environment. The primary concern of this work is to resolve the issues associated with the single encryption algorithm like low encryption efficiency, and unreliable data storage. The study utilizes the ECC encryption to encrypt the original data, which

ensures protection against security threats during transmission. Then, a homomorphic encryption was employed to ensure data integrity in the system. The experimental outcomes manifest that this algorithm improves the storage efficiency by 27.6%. Moreover, it offers greater protection to the sensitive information against cyberthreats. However, this framework cannot handle large volumes of data. Table I presents the literature survey.

TABLE I. LITERATURE SURVEY

Authors	Technique	Results	Merits	Demerits
Lav Gupta et al. [21]	Deep hierarchical stacked neural system	Reduced training time by 26.2%, obtained an average accuracy between 93% to 95%	High convergence, less training time, and greater detection accuracy	Limited generalization and vulnerable to adversarial attacks
Reem Al Saleh et al. [22]	NBR with CBiLSTM	Accuracy-96.7%, f-measure-97%, recall-96.5%, and recall-97.4%	High malicious data detection performances, and minimum training time	Less scalable and interpretable
R. Denis and P. Madhubala [23]	Hybrid encryption scheme (AES-RSA)	Data confidential rate-0.95. running time-5.4s	Provides greater protection to data during transmission	Key management complexity, and increased computational overhead
Fursan Thabit et al. [24]	Lightweight cryptographic algorithm	Execution time-6.4s	Strong security level and less execution time	Less adaptable for emerging threats
P. Chinnasamy et al. [25]	Hybrid cryptographic algorithm combining ECC and Blowfish	Data confidentiality-96%, security level-97%	Higher security and confidentiality	Highly complex, and resource intensive
Shaopeng Guan et al. [26]	Hadoop-assisted big data storage scheme	Improves cloud storage efficiency by 27.6%	Protects the sensitive data from cyberthreats	Cannot handle large volumes of data

III. SYSTEM MODEL WITH PROBLEM STATEMENT

Cloud computing is the process of delivering computing services such as databases, storage, servers, networking, analytics, software, and intelligence over the internet for offering flexibility and reliability, and fostering faster innovation. Although the CC model offers various benefits to the business institutions, it is prone to security challenges. This demands an effective security risk assessment framework to ensure confidentiality, integrity, and availability of the CC model [27]. A system security risk assessment model includes data collection module, preprocessing component, and risk assessment module (quantitative, statistical, DL or ML models). Fig. 1 presents the system model. The data acquisition component contains various data like network traffic, logs, system behavior, historical security risks, etc. In the preprocessing component, the collected database undergoes preprocessing steps to improve its quality for subsequent analysis. After preprocessing, a risk assessment module was created using either DL or ML models. These models are trained using the database to differentiate the patterns between the normal and malicious behavior. After the training process, it is validated for unseen data or real-time scenarios for risk evaluation.

Although these risk assessment models provide automatic prediction of malicious behavior, they often rely on threshold settings to differentiate normal and anomalies behavior. This may lead to incorrect predictions. Moreover, the conventional models require more computational resources for intensive training, making the system costly and complex. Also, they face challenges in providing scalability to handle large-scale cloud environments. In addition to this, the traditional security risk analysis algorithms can adapt to the emerging and evolving cyber threats, making it less reliable and effective for real-world scenarios. Moreover, only few studies concentrated on implementing the security measures on the CC model to ensure protection to the sensitive data during transmission and storage. To address the issues with the conventional security risk assessment techniques, and to bridge the research gap, we developed a collaborative framework, which uses deep learning and meta-heuristic optimization algorithms to precisely identify the security threats, and implements an optimized cryptographic algorithm for secure transmission of information in the cloud.

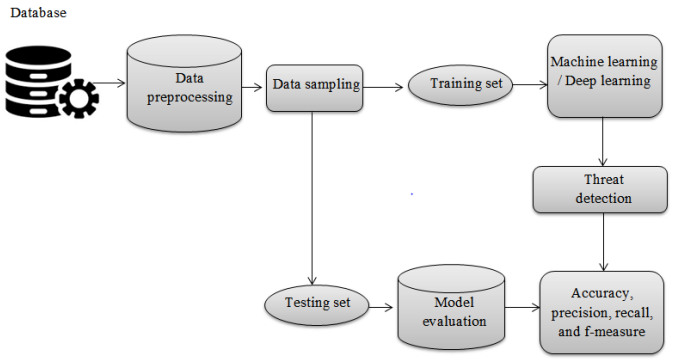


Fig. 1. System model.

IV. PROPOSED FRAMEWORK FOR SECURITY RISK ASSESSMENT

A novel collaborative security framework was proposed in this article by combining the benefits of deep learning, meta-heuristic optimizations, and cryptographic algorithms. The novelty of the study lies in the seamless and effective integration of these algorithms for ensuring security to the CC

model. The primary objective of the model is to identify the security threats/risks associated with the CC model, and to ensure secure data communication in the cloud. Firstly, a database containing network traffic, historical security risks, etc., was collected and imported into the system. Secondly, the accumulated database was preprocessed to handle the missing values, outliers, errors, etc. This process enhances the quality of the database, and makes it suitable for subsequent analysis. Then, an attack prediction module was created by integrating deep recurrent neural network and krill herd optimization.

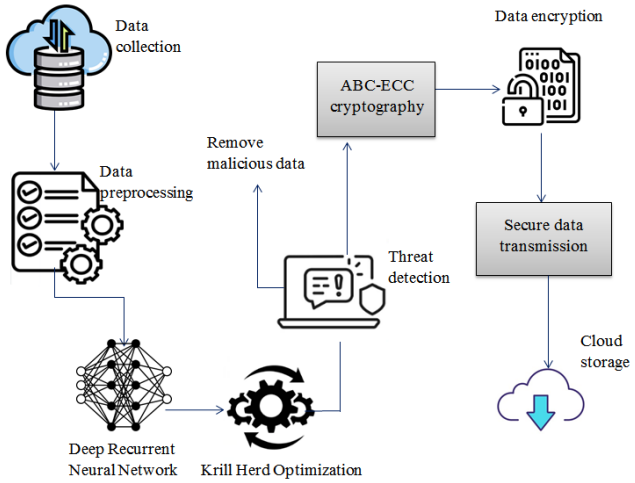


Fig. 2. Architecture of the proposed framework.

In this module, the DRNN was trained using the collected database to distinguish the patterns between the normal and malicious behavior, consequently we applied krill herd optimization for hyperparameter tuning and optimization. This combination enables the system to continuously learn the patterns and correlations between the normal and malicious traffic, ensuring adaptability to the emerging threats. After threat detection, the detected threats are eliminated from the system to offer a higher security level. Further, an optimized cryptographic algorithm was proposed to ensure security against the cyberthreats during the transmission process. The developed algorithm incorporates the efficiency of ABC into the ECC algorithm for optimally selecting the parameters for key generation, and distribution. This algorithm encrypts the data before transmission, making it unreadable for third parties. After encryption, the encrypted data is transferred to the cloud for storage, where the decryption step can be performed by the authenticated authorities and decode the data. Fig. 2 presents the architecture of the proposed strategy.

A. Data Collection

The proposed study commences with the collection of dataset relevant to cloud computing. The collected database may contain information like network traffic, logs, system characteristics, historical security risks, etc. The presented study utilized the publicly available security database named "DDoS SDN", and it is accessible at https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-dataset. This database contains 104,345 instances and 23 features, and it is developed for categorizing network traffic as either traffic or malicious within a cloud environment. The dataset is structured with a target variable named "label," containing binary values (0 or 1). Here 1 indicates malicious, and 0 defines benign. Out of the 23 features, 3 are categorical features, and 20 are numerical features, providing a wide range of information relevant to network traffic investigation. The size of the dataset is 12.56MB and it is available in csv format.

B. Preprocessing

Data preprocessing is an important step, which plays a significant role in improving the quality of the dataset. It includes steps like cleaning, transforming, scaling, etc., to make the database ready and reliable for further analysis. In the proposed work, we applied steps like handling missing values, feature scaling, and handling class imbalance to the raw database for making it effective for subsequent processes. Handling missing values indicates the process of detecting the null values in the database and replacing them with values determined using an imputation algorithm. In imputation, the missing value was replaced with the mean of the non-missing values of the dataset. The imputation process is mathematically represented in Eq. (1).

$$M_n = \frac{\sum_{i=1}^{m} D_i}{m} \tag{1}$$

Where M_n indicates the mean, m indicates the number of samples, and D indicates non-missing value. Then, feature scaling was performed to ensure that all the features in the dataset contribute equally to the analysis. This step prevents the dominance of a single feature in threat detection. Here, we applied a min-max scaling algorithm to rescale all features to a fixed range (0 and 1), and it is expressed in Eq. (2).

$$S_D = \frac{D - D_{mn}}{D_{mx} - D_{mn}} \tag{2}$$

Where S_D indicates the scaled value of the feature, D_{mn} denotes the maximum value of the feature, and D_{mx} represents the minimum value of the feature. Then, the database was checked for class imbalance. If there is imbalance in the class, then oversampling was done to address that. Finally, the database was split into training and testing to evaluate the performance of the proposed threat detection algorithm. The introduction of these steps not only enhances the dataset quality but also increases the speed of subsequent analysis, leading to enhanced and timely assessment of cybersecurity threats. This preprocessed dataset was fed into the DRNN-KHO for cybersecurity risk assessment.

C. DRNN-KHO for Threat Detection

Threat detection defines the process of classifying the normal and malicious network traffic. In the developed work, we proposed an innovative threat detection model by combining the efficiency of deep recurrent neural network, and krill herd optimization algorithm. Here, the DRNN acts as a classification module categorizing normal and malicious data,

while the KHO intends to refine its parameters to improve the overall threat detection process. The DRNN is an artificial neural network, which is mainly used in sequential data processing. The architecture of DRNN is similar to the conventional RNN structure with an additional dense layer. Generally, there are two different RNN architectures namely: Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) [28]. In the proposed work, we used a LSTM model for predicting the cybersecurity threats. The proposed DRNN contains input layer, recurrent layer (LSTM), dense layer, and output layer. The input layer of the system accepts the preprocessed database as input. This layer converts into a suitable format for further analysis, and forwards into the recurrent layer, where the model intends to learn the patterns and long-range dependencies within the data for identifying the normal and malicious data. The LSTM layer includes input gate, forget gate, candidate cell state, hidden state, cell state, and output gate. These cells and gates process the input sequence and capture the complex intricate patterns and correlations within the data, and they are expressed in Eq. (3), (4), (5), (6) and (7).

$$f_t = \sigma\left(W_f .\left[H_{t-1}.D_t\right] + B_f\right) \tag{3}$$

$$i_t = \sigma\left(W_i .\left[H_{t-1}.D_t\right] + B_i\right) \tag{4}$$

$$o_t = \sigma\left(W_o .\left[H_{t-1}.D_t\right] + B_o\right) \tag{5}$$

$$c_t' = \tanh\left(W_c .\left[H_{t-1}.D_t\right] + B_c\right) \tag{6}$$

$$H_t = H_t * \tanh\left(c_t\right) \tag{7}$$

Where f_t defines the forget gate, i_t denotes the input gate, c_t' indicates the candidate cell state, c_t represents cell state, o_t refers to the output gate, σ represents the sigmoid activation function, $\tanh$ denotes the hyperbolic tangent activation function. Further, W defines the weight matrices, and B represents the bias vectors. D_t indicates the input sequence at time step t, and H_{t-1} represents the hidden state at the previous time step. These gates perform certain functions to capture the long-term dependencies and intricate patterns within the sequential data for detecting the threats. The outcome of this layer is forwarded into the dense layer, which is typically a fully connected layer. In this layer, each neuron is interconnected with every neuron in the recurrent and output layer, making the system understand the difference between normal and malicious data. The dense layer is presented in Eq. (8).

$$ds_t = \sigma\left(W_{dh} .H_t + B_d\right) \tag{8}$$

Where W_{dh} defines the weight interconnecting the LSTM layer and dense layer, B_d denotes the bias vector, and σ represents the activation function. These learned patterns and extracted features are fed into the output layer, which produces the probability of being normal or malicious data. The resultant of the output layer is represented in Eq. (9).

$$y_t = \left(W_{ot} .ds_t + B_{ot}\right) \tag{9}$$

Where y_t defines the output at time step t, which indicates the probability of the input being normal or malicious. If the output value is 0, the system indicates it as “normal,” and “malicious.” defines the bias vector and indicates the activation function, which converts the learned features into probability values. At each iteration, this system learns to understand the difference between the normal and malicious data. The loss incurred by the model is presented in Eq. (10).

$$L_{oss} = -\frac{1}{m}\sum_{t=1}^{m}\left[a_t \log\left(y_t\right) + \left(1-a_t\right)\log\left(1-y_t\right)\right] \tag{10}$$

Where L_{oss} indicates the loss, m defines the number of training samples, and a_t represents the actual outcome. This loss function was minimized by adjusting its parameters like weight and bias. In the proposed work, the training loss incurred by the DRNN is optimized using the KHO algorithm by continuously iteratively adjusting its hyperparameters. The KHO is a nature-inspired algorithm developed based on the herding behavior of krills to solve the optimization problems. Here, the objective function of KHO is to maximize the prediction accuracy of DRNN by refining and fine-tuning its hyperparameters to its optimal range. This optimization algorithm was developed based on three main actions: (1) movement induced by other krill individuals, (2) foraging characteristics, and (3) random diffusion. The optimization process begins with the random initialization of the krill population in the search space [29]. Each krill individual in the population defines the hyperparameter values. After initialization, the fitness value of each individual was determined based on the defined objective function, which is represented in Eq. (11).

$$O_{bj} = \max\left(Ac\right) \tag{11}$$

Where O_{bj} indicates the objective function, and Ac denotes the threat detection accuracy of the DRNN model. The fitness of the parameter set will be high if the DRNN model achieved high threat detection accuracy and vice versa. After fitness evaluation, the parameter values are updated following the three actions mentioned above. These steps enable the system to explore the population space and find the optimal value. The parameter updation is mathematically expressed in Eq. (12).

$$p_v\left(t+1\right) = p_v\left(t\right) + \Delta t \frac{dp_v}{dt} \tag{12}$$

Where $p_v\left(t\right)$ indicates the parameter value at time t, $p_v\left(t+1\right)$ defines the updated parameters, and $\Delta t \frac{dp_v}{dt}$

represents the scale factor. Then, the fitness solution was determined for updated parameter sets. Finally, the updated fitness and old fitness was compared. If the updated fitness is greater than the old fitness, the updated parameter values are used for training. This process continues until reaching the maximum iteration count. Thus, the hybrid mechanism detects and classifies the normal and malicious data by learning the patterns within the data effectively. After threat detection, the malicious data is removed from the network for ensuring security and privacy to the data.

D. ABC-ECC for Secure Data Storage

After threat detection, we developed a hybrid cryptographic algorithm combining the efficiency of artificial bee colony optimization with the elliptic curve cryptography (ABC-ECC) for secure data transmission and storage in the cloud. Here, before transmitting the data to the cloud, we encrypt the data using the proposed ABC-ECC model, which prevents the intrusion of third parties during transmission. Elliptic curve cryptography is a public-key cryptographic algorithm developed based on the concept of elliptic curves over finite fields [30]. The elliptic curve is mathematically defined in Eq. (13).

$$y^2 + xy = x^3 + ux + v \tag{13}$$

Where x and y defines the coordinates of the curve, u indicates the curve slope, and v represents the constant term indicating the curve displacement. The important property of this curve is that we can define a rule for adding two points U and V on the curve to determine a third point W, which is also on the curve. This defined rule agrees with the normal addition properties. This forms the basis for operations like scalar multiplication, which are at the heart of ECC's security and efficiency. Consider a case where Alice and Bob want to share the information. Firstly, Alice selects a random integer A_r as its private key, and it is denoted as. Consequently, Bob also selects a random secret integer B_r as his private key, which is defined as. Before key generation and transmission, both Bob and Alice agree upon a non-secret (elliptic curve) and fixed curve point F_p (non-secret). Then, both Alice and Bob calculate their public key, which is represented in Eq. (14), and (15).

$$A_p = F_p A_r \tag{14}$$

$$B_p = F_p B_r \tag{15}$$

Where A_p and B_p define the public keys of Alice and Bob. They compute the public keys by multiplying their private keys with the fixed curve point. These keys are used for performing the encryption operations. If Alice wants to send the data to Bob, it computes a shared secret using her private key and Bob's public key. Subsequently, Bob also computes a shared secret using its private key and Alice's public key. After generating a shared secret, it is used to establish a secure communication channel between Alice and Bob, where both parties use their shared secret as the key for encrypting and decrypting messages exchanged between them. Although the ECC approach is more reliable and effective than conventional cryptographic techniques like Rivest-Shamir-Adleman, its efficiency and security relies on the selection of fixed curve points. Hence, we applied the Artificial Bee Colony algorithm for selecting the fixed curve point from the defined elliptic curve for improving the efficiency and security of the conventional ECC approach. The ABC is a meta-heuristic optimization algorithm developed based on the intelligent foraging characteristics of honey bee swarms. The ABC model contains three bee groups namely: onlookers, employed, and scouts. In the proposed work, we employed the employed bee phase for finding the optimal fixed curve point [31]. The initialization of the ABC population was expressed in Eq. (16).

$$F_{pi} = l_b + rand(0,1) * (u_b - l_b) \tag{16}$$

Where l_b and u_b defines the lower and upper bound of the parameter, respectively. After initialization, the employed bees search for new food sources. Here, the food sources represent the optimal fixed curve point. First they find the neighboring food source and determine its profitability (fitness solution). In case of fixed curve point selection, the fitness indicates its ability to improve the efficiency and security of ECC, which is mathematically represented in Eq. (17).

$$F'_{pi} = F_{pi} + \phi_{pi}(F_{pi} - r_{pi}) \tag{17}$$

Where r_{pi} indicates the randomly selected fixed curve point. Further, the bees explore the population to find the best solution. Further, the fitness solution was again evaluated to select the optimal solution. This process is an iterative process, and at each iteration, the ABC optimization finds the optimal fixed curve point; thereby improving the efficiency and security in data encryption. The encrypted data is then transferred into the cloud for storage and further analysis. Thus, the proposed collaborative framework detects the threats and offers security within the cloud network. The working of the proposed algorithm is described in pseudocode format in Algorithm 1.

Algorithm 1
Start {
Initialize the DDoS database;
Data preprocessing:
1. Handling missing values//*Apply imputation*
2. Feature scaling; //*Apply min-max scaling*
3. Handling class imbalance;// *Perform oversampling*
Data splitting;
Threat detection:
Define DRNN layers;
Initialize epoch count, weights;
Model training:
For each training epoch:
Extract the features;
Learn the patterns and correlations;
Determine output probability;

if (y_t =0) **Normal**;
else **Malicious**;
End for;
Eliminate threats;
Determine loss function;
Optimization:
Initialize KHO parameters (population size, maximum iteration);
Initialize the population;
Define objective function;
For each iteration:
Determine fitness solution;
Update parameter set;
Evaluate fitness for updated parameter sets;
if (new fitness> old fitness) **return updated parameters;**
else **return old parameters;**
End for;
Secure data transmission:
Initialize ECC parameters;
Select fixed curve point;
Select random integers;
Generate public keys;
Perform data encryption;
ABC optimization:
Initialize ABC population, maximum iteration;
Define objective;
For each iteration:
Determine fitness;
Exploration;
End for;
Return optimal fixed curve point;
Model evaluation;
}Stop

V. Results and Discussion

A collaborative security framework was proposed for assessing the cybersecurity risks within the cloud network. The proposed framework was modeled in the Pycharm tool, running in Windows 10 Operating system. The proposed framework was trained and validated using the DDoS dataset, and the results are determined in terms of accuracy, recall, f1-score, mean absolute error (MAE), mean square error (MSE), etc.

A. Training and Testing Performances

In this module, we discuss the training and testing performances of the proposed algorithm. Firstly, the database was split for training and testing purposes, and the performances are assessed as accuracy and loss for different learning rates (70% and 80%). The accuracy metric measures how quickly the proposed algorithm learns the patterns of normal and malicious traffic. The loss metric measures the deviation between the actual and predicted outcomes. Fig. 3 presents the training and testing performance of the developed algorithm for 70% learning rate. The training accuracy indicates how effectively the designed model fits into the training sequence and learns the patterns and correlations for distinguishing the normal and malicious traffic. The developed algorithm achieved greater training accuracy of 0.96, and 0.99 for 70% and 80% learning rates. This validates that the designed approach fastly learns and understands the pattern difference between the normal and threats.

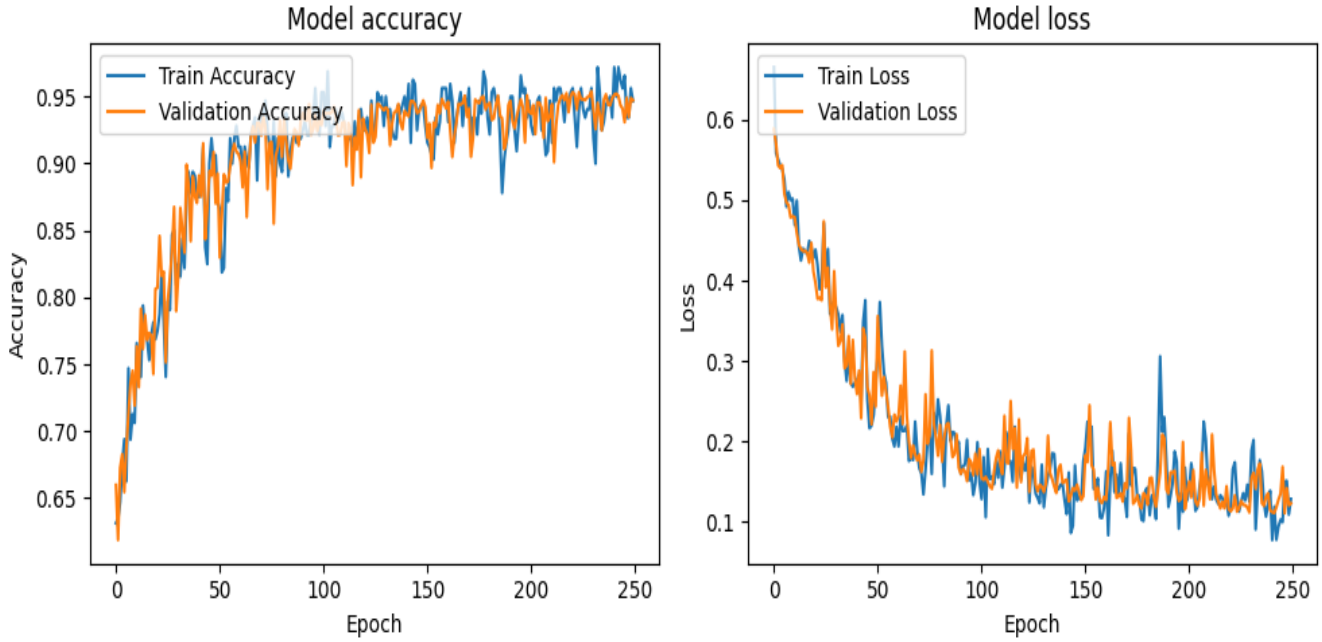


Fig. 3. Training and Testing performance of the proposed model for 70% learning rate.

Consequently, the testing accuracy was evaluated for 70% and 80% learning rates. The testing accuracy defines how effectively the developed model performs predictions on unseen data. The proposed algorithm achieved greater testing accuracy of 0.95, and 0.98 for 70% and 80% learning rates. These improved testing accuracy highlights that the proposed algorithm generalizes well on the new data, making it effective and reliable for real-time threat detection. Also, it is observed that the designed model obtained greater accuracy for 80% learning rate, which manifests that the model's performance increases with higher learning rate.

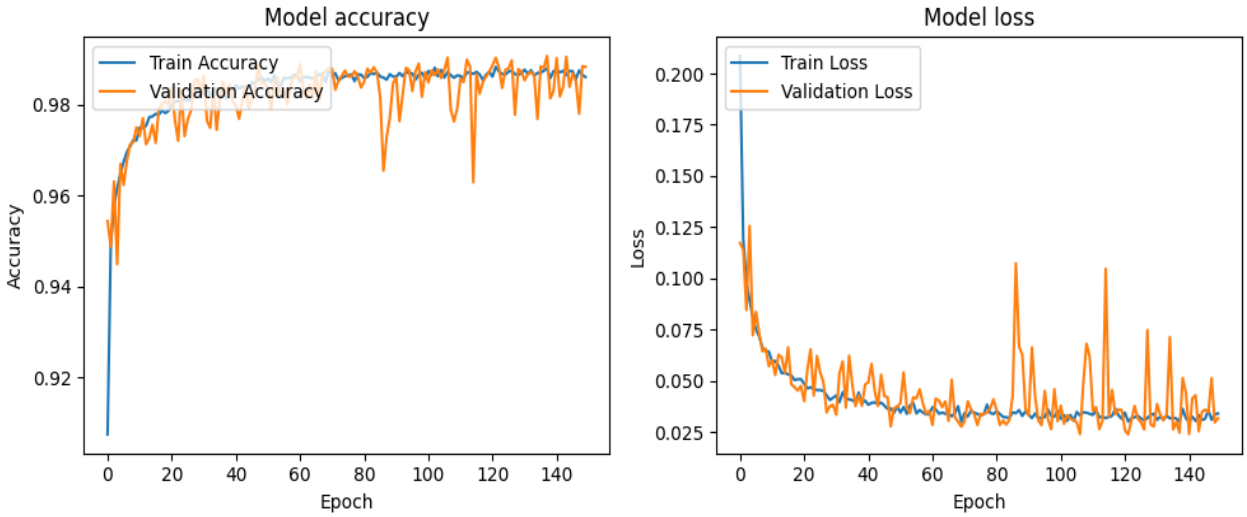
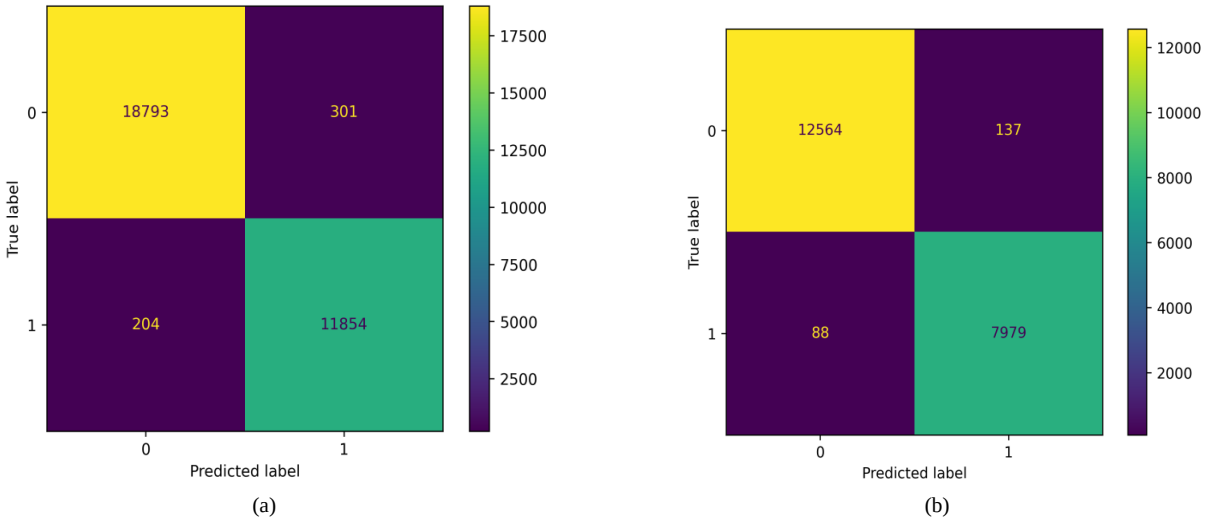


Fig. 4. Training and testing performance of the proposed model for 80% learning rate.

The training loss quantifies the misclassification made by the proposed algorithm within the training sequence. It measures the deviation between the actual and the predicted results in the training sequence. The developed model obtained lower training loss of 0.1, and 0.03 for 70% and 80% learning rates. On the other hand, the testing loss was determined for 70% and 80% learning rates. The testing accuracy measures the model's generalization to unseen data by quantifying the difference between the actual and the predicted outcomes. The designed approach attained minimum loss of 0.11, and 0.4 for 70% and 80% learning rates. Fig. 4 presents the training and testing performance incurred by the developed algorithm for 80% learning. From this intensive model evaluation, it is clear that the proposed algorithm achieved better performances for increased learning rate. Also, it achieved greater accuracy and minimum loss in both train and test phases, highlighting its generalization ability and capacity to prevent the overfitting problem. This illustrates that the proposed technique effectively learns the patterns within the database, and predicts the normal and malicious data accurately.



(a) (b)

Fig. 5. Confusion matrix: (a) 70% learning rate, (b) 80% learning rate.

The confusion matrix is a performance evaluation tool deployed in deep learning to determine its efficiency in classification tasks. It is a table, which visualizes the classification performance of the model by summarizing the counts of correct and incorrect classifications made by the system on the sequence of data for which the actual values are known. Fig. 5 (a, b) presents the confusion matrix obtained for 70% and 80% learning rates. The cell in the matrix indicates the count of instances where the predicted class correlates with the actual class. This matrix is divided into four sections namely, true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). TP indicates the instances where the malicious traffic is correctly predicted, while TN represents the instances where the normal traffic is correctly predicted. On the other hand, FP defines the scenario when the normal traffic is incorrectly predicted as malicious, while FN denotes the scenario when the malicious traffic is incorrectly predicted as normal. By examining the confusion matrix, we determine the efficiency of the proposed model in threat detection and classification.

1) Cryptographic algorithm performances: In this module, the performance incurred by the proposed cryptographic algorithm (ABC-ECC) was examined in terms of encryption time, decryption time, success rate, and turnaround time. These performances are assessed by increasing the data size from 1 to 100000. Fig. 6 presents the assessment of cryptographic algorithm performances. Fig. 6 (a) presents the encryption time analysis. The encryption time measures the time taken by the proposed system for performing the encryption operation. The proposed algorithm obtained an average encryption time of 0.055, which illustrates that it consumes less time for encoding the dataset. Consequently, the decryption time of the system was determined and it is depicted in Fig. 6 (b). The proposed technique obtained an average decryption time of 0.03, which highlights that the system consumes minimum time for performing the decryption task.

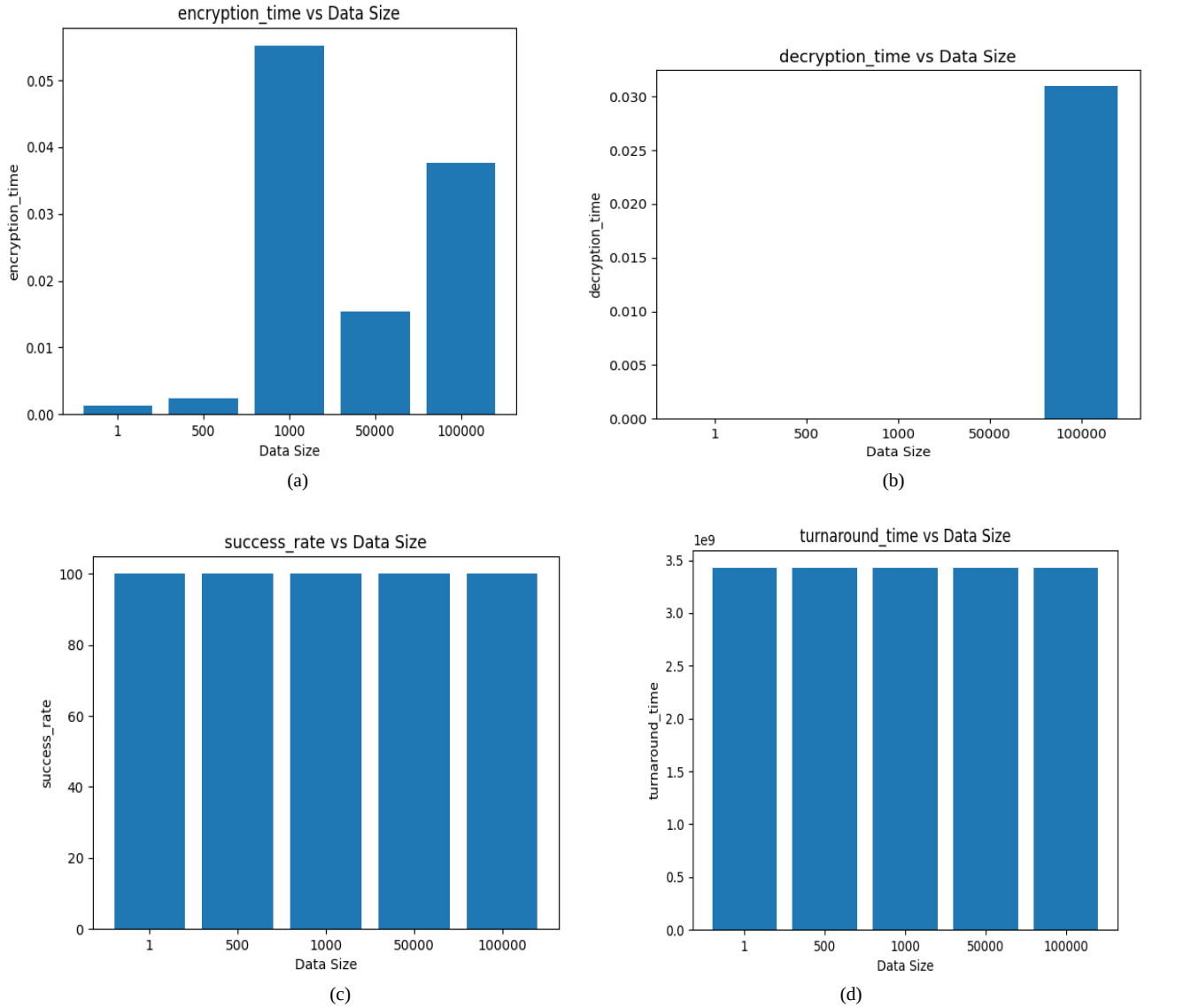


Fig. 6. Performance analysis: (a) encryption time, (b) decryption time, (c) success rate, and (d) turnaround time.

Further, the success rate was determined to evaluate how the proposed system prevents the cyber security threats during transmission. Fig. 6 (c) presents the analysis of success rate over increasing data size. The proposed technique achieved a maximum success rate of 100%, highlighting that it effectively resists the attacks and protects the data. Finally, the turnaround time was assessed over increasing data sizes. The turnaround time indicates total time required to complete a cryptographic operation, including both encryption and decryption processes, in association with other tasks like key generation, initialization, and data processing. The developed model obtained a minimum turnaround time of 3.43E+09, demonstrating its computational efficiency and data processing speed.

B. Evaluation Metrics

In this module, we discuss the parameters used for examining the performance of the proposed algorithm. The parameters include accuracy, recall, f1-score, MSE, MAE, Normalized Mean Square Error (NMSE), success rate, encryption time, decryption time, and turnaround time. The definition for the parameters are described below.

a) Accuracy: Accuracy measures how effectively the developed model predicts the threats in the cloud model. It quantifies the correctly predicted positive and negative instances to the total instances, and it is formulated in Eq. (18).

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{18}$$

Where T_p T_n F_p, and F_n defines TP, TN, FP, and FN, respectively.

b) Recall: Recall measures the proportion of true positives that were correctly identified by the developed model. It is calculated using Eq. (19).

$$\text{Re}call = \frac{T_p}{T_p + F_n} \tag{19}$$

c) F1-score: F1-score: F1-score is the harmonic mean of precision and recall, providing a balanced measure between them. It is mathematically expressed in Eq. (20).

$$F - measure = \frac{2 * T_p}{\left(2 * T_p + F_p + F_n\right)} \tag{20}$$

d) Mean squared error: MSE measures the average squared difference between the predicted values and the actual values, and it is formulated in Eq. (21).

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - y\right)^2 \tag{21}$$

Where y defines the actual value, and y_i indicates the predicted value.

e) Mean absolute error: MAE measures the average absolute difference between the predicted values and the actual values. It is expressed in Eq. (22).

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|y_i - y\right| \tag{22}$$

f) Normalized mean square error: NMSE measures the relative mean squared error between the predicted values and the actual values, normalized by the variance of the actual values, and it is formulated in Eq. (23).

$$NMSE = 10\log_{10}\left[\frac{\sum_{i=1}^{m}\left|y_i - y\right|^2}{\sum_{i=1}^{m}\left|y_i\right|^2}\right] \tag{23}$$

g) Success rate: Success rate refers to the effectiveness of a cryptographic algorithm in resisting attacks. It measures how effectively the developed model prevents cyber security threats.

h) Encryption time: Encryption time measures the time taken by a cryptographic algorithm to encode the entire data.

i) Decryption time: Decryption time measures the time taken by a cryptographic algorithm to decode the original message from the encrypted data.

j) Turnaround time: Turnaround time indicates the total time required to complete a cryptographic operation, including both encryption and decryption processes.

These parameters provide comprehensive insights into different aspects of the performance of the proposed algorithm, encompassing accuracy, efficiency, security, and reliability.

C. Comparative Analysis

In this section, the performances incurred by the proposed model were compared and validated with the existing techniques such as Convolutional Neural Network (CNN) [32], Deep Neural Network (DNN) [33], Recurrent Neural Network (RNN) [34], and Support Vector Machine (SVM) [35].

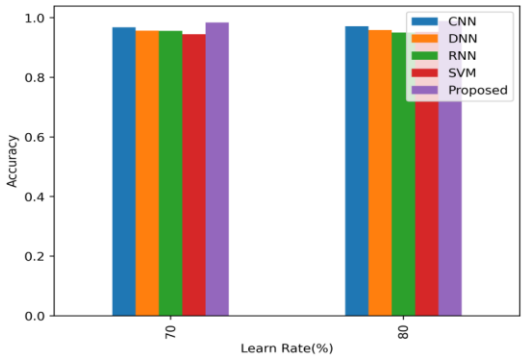
1) Comparison of threat detection performance: In this subsection, we compare the threat detection performance of the developed model with the conventional techniques such as CNN, DNN, RNN, and SVM. The performances are evaluated in terms of metrics like accuracy, f1-score, recall, MAE, MSE, and NMSE at 70% and 80% learning rates. Fig. 7 presents the comparative assessment. Fig. 7 (a) presents the accuracy comparison. The above-stated existing techniques and the proposed DRNN-KHO algorithm obtained an approximate accuracy of 0.96, 0.95, 0.955, 0.94, and 0.983, respectively at 70% learning rate, while the above techniques achieved an approximate accuracy of 0.97, 0.958, 0.95, 0.952, and 0.989, respectively at 80% learning rate. From this analysis, it is clear that the proposed algorithm achieved improved accuracy than the existing techniques. The improvement of accuracy signifies its efficiency in detecting and classifying the normal and

malicious traffic within the network. Consequently, we evaluated the MSE with the above-stated existing techniques. The above-mentioned existing algorithms and the developed approach obtained MSE of 0.03, 0.04, 0.044, 0.05, and 0.01, respectively at 70% accuracy. On the other hand, these approaches earned a MSE of 0.02, 0.041, 0.049, 0.047, and 0.01, respectively at 80% learning rate. The significant reduction of MSE by the designed model highlights that it accurately classifies normal and malicious traffic. Fig. 7 (b) presents the comparison of MSE.

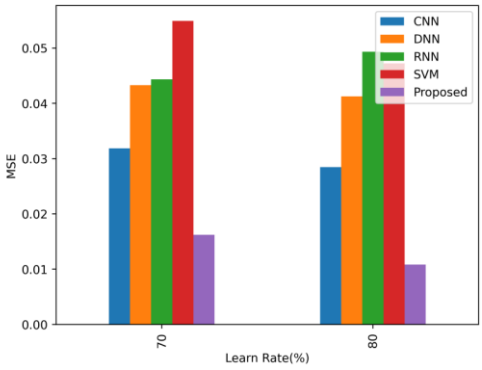
TABLE II. PERFORMANCE COMPARISON AT 70% LEARNING RATE

Metrics	CNN	DNN	RNN	SVM	PROPOSED
MSE	0.031876	0.043304	0.044331	0.054924	0.016211
MAE	0.031876	0.043304	0.044331	0.054924	0.016211
NMSE	0.015315	0.020805	0.021299	0.026389	0.007789
Recall	0.968206	0.956729	0.956029	0.945172	0.983659

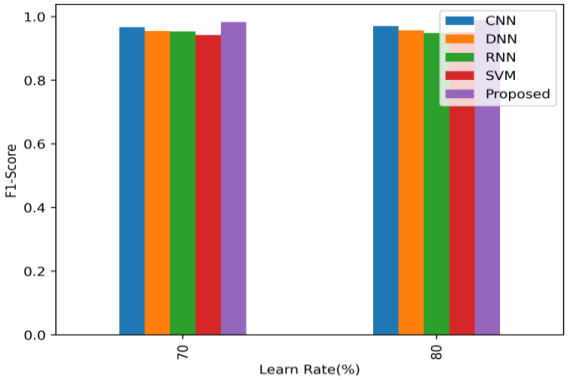
On the other hand, the f1-score was compared with the existing techniques. The f1-score determines the balanced classification performance considering both positives and negatives. These conventional models and the algorithm obtained f1-score of 0.96, 0.954, 0.953, 0.942, and 0.982, respectively at 70% learning rate, while these algorithms obtained f1-score of 0.966, 0.954, 0.953, 0.942, and 0.982, respectively, at 80% learning rate. The significant improvement made by the developed model highlights its efficiency in balanced classification performance than the conventional techniques. Fig. 7(c) depicts the comparison of f1-score. Subsequently, the recall was also evaluated and compared with the existing techniques. The above-stated existing algorithms and the proposed approach earned recall rate of 0.968, 0.956, 0.956, 0.945, and 0.983, respectively at 70% learning rate, while these models obtained recall of 0.970, 0.956, 0.948, 0.950, and 0.988, respectively at 80% learning rate. The enhancement of the recall made by the developed algorithm highlights its efficiency in classifying the network traffic. Fig. 7(d) presents recall comparison. Table II presents the comparative analysis of classification performance at 70% learning rate.



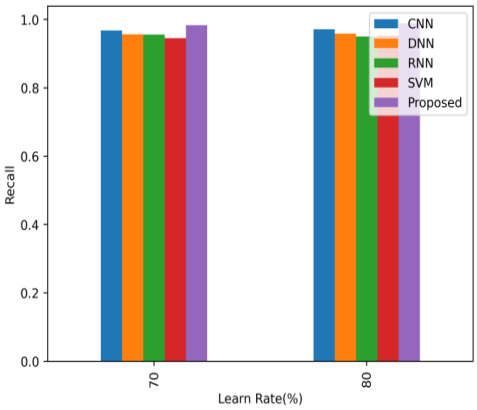
(a)



(b)



(c)



(d)

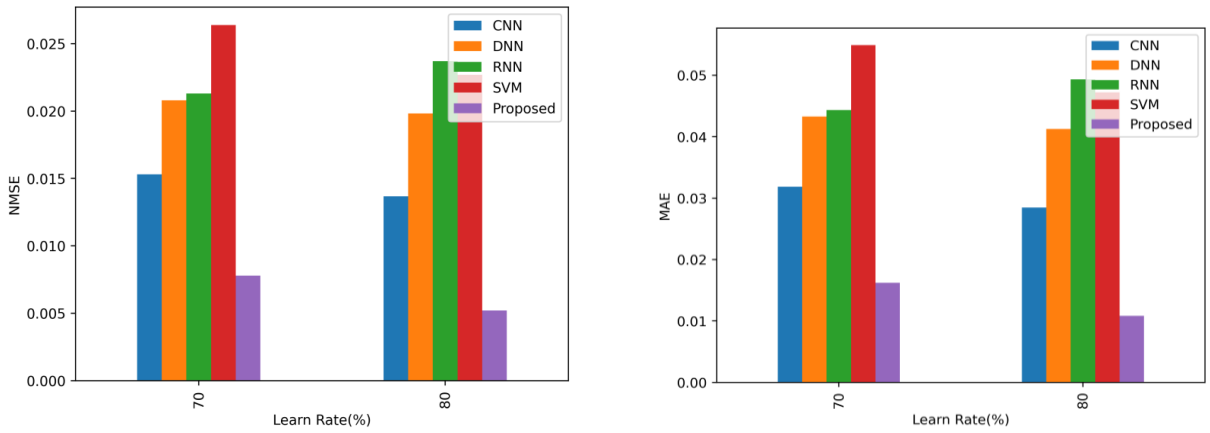


Fig. 7. Comparative analysis: (a) accuracy, (b) MSE, (c) F1-score, (d) recall, (e) NMSE, and (f) MAE.

TABLE III. PERFORMANCE COMPARISON AT 80% LEARNING RATE

Metrics	CNN	DNN	RNN	SVM	PROPOSED
MSE	0.028473	0.041281	0.049339	0.04722	0.010834
MAE	0.028473	0.041281	0.049339	0.04722	0.010834
NMSE	0.01368	0.019834	0.023705	0.022687	0.005205
Accuracy	0.971527	0.958719	0.950661	0.95278	0.989166
F1-Score	0.970089	0.9567	0.948259	0.950446	0.988611
Recall	0.971578	0.958868	0.950477	0.952343	0.989152

Furthermore, the MAE, and NMSE performances of the developed algorithm were compared and evaluated with the conventional techniques. Fig. 7 (e, f) presents the comparison of MAE and NMSE. These metrics measure the error in classification. The existing techniques and the developed approach obtained MAE of 0.03, 0.04, 0.044, 0.05, and 0.01, respectively at 70% learning rate, while these techniques earned NMSE of 0.01, 0.02, 0.021, 0.026, and 0.007, respectively at 70% learning rate. On the other hand, these models obtained MSE of 0.03, 0.04, 0.044, 0.05, and 0.01, respectively, and these algorithms achieved NMSE of 0.01, 0.02, 0.021, 0.026, and 0.007, respectively at 80% learning rate. From this analysis, it is clear that the developed algorithm achieved minimum MAE and NMSE compared to the existing techniques. This highlights the model's efficiency of accurately classifying the threats. In addition, it manifests that the designed model provides minimum error, and lower false positives and negatives. Table III presents the comparison of classification performance of different models at 80% learning rate. From this comprehensive comparative evaluation, it is clear that the designed model achieved better performances compared to the existing techniques, highlighting its efficiency of accurately classifying normal and malicious network traffic.

2) Comparison of cryptographic algorithm performance: In this section, we compare and evaluate the performance of the cryptographic algorithms with the existing algorithms like RSA [36], Advanced Standard Encryption (ASE) [37], ECC [38], and Diffie Hellman Key Exchange (DHKE) [39]. The performances are evaluated in terms of success rate, encryption time and decryption time, and they are evaluated for increasing data size from 1 to 100000.

The encryption time measures the time taken by the proposed algorithm for encoding the entire dataset. The existing techniques such as RSA, ASE, ECC, and DHKE obtained an average encryption time of 0.30, 0.41, 0.23, and 0.15, while the proposed algorithm achieved a minimum encryption time of 0.055. This illustrates that the designed algorithm quickly encrypts the dataset. Fig. 8 (a) presents the comparison of decryption time. Consequently, the decryption of above-stated existing techniques and the proposed are evaluated, and it is displayed in Fig. 8 (b). The decryption time defines the time taken by the developed algorithm for decoding the original message from the encrypted data. These models obtained decryption time of 0.30, 0.89, 0.35, 0.25, and 0.0309, respectively. From this analysis, it is clear that the proposed approach obtained less decryption time than others, highlighting its efficiency and speed in decrypting the data.

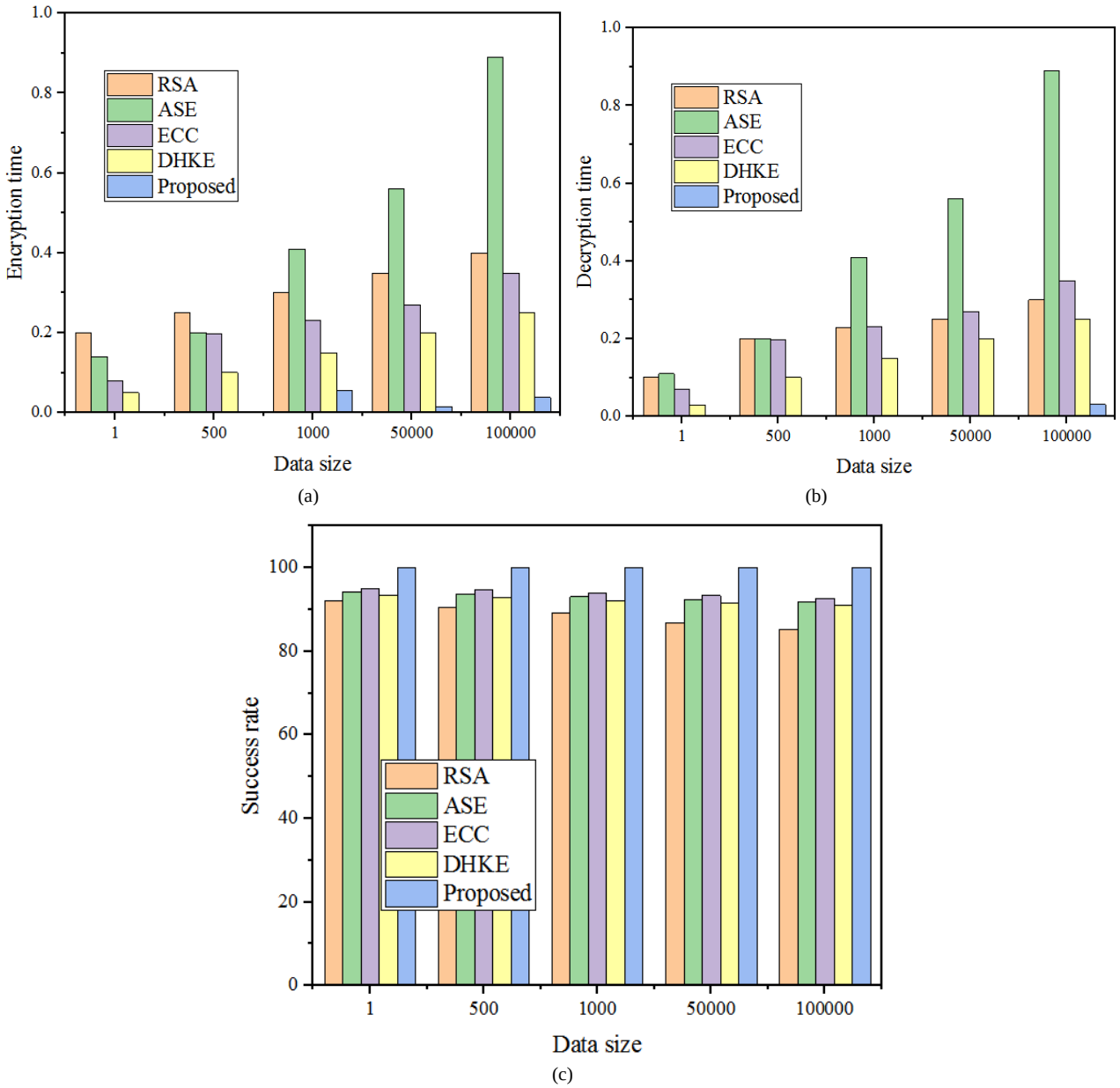


Fig. 8. Comparative analysis: (a) encryption time, (b) decryption time, and (c) success rate.

Finally, the success rate was also determined and compared with the existing techniques. The above-mentioned existing techniques and the proposed algorithm obtained an average success rate of 92.11, 94.21, 95.05, 93.52, and 100, respectively. The significant improvement of the success rate by the proposed algorithm manifests its effectiveness in preventing the intrusion during data transmission. Fig. 8 (c) presents the comparison of success rate. This intensive analysis of system performances suggest that the developed algorithm resists the security attacks and provides greater level of security to the data during transmission.

VI. Conclusion

In this study, we proposed a collaborative framework for ensuring security in the cloud environment. The objective of the work is to assess the cyber threats and provide secure data transmission within the cloud network. The developed model was validated using the publicly available DDoS database, which is preprocessed and trained into the DRNN-KHO module for cyber attack detection. Consequently, we developed a hybrid cryptographic algorithm by integrating ABC into the ECC algorithm for securely transmitting the information into the cloud. The developed framework was modeled and implemented in the Pycharm tool, and the performances are assessed as accuracy, recall, f1-score, MSE, MAE, NMSE, success rate, etc. The experimental results highlight that the developed approach obtained greater accuracy of 0.989, higher recall of 0.989, improved f1-score of 0.988, reduced MSE of 0.010, lower MAE of 0.0108, and minimum NMSE of 0.005, respectively in threat detection. Consequently, the implementation of ABC-ECC algorithm suggests that it achieved 100% success rate, 0.037s encryption time, 0.0309s

decryption time, and 3.43E+09 turnaround time. Finally, we made a comparative assessment with the existing techniques, and it validated that the proposed algorithm achieved better performances than others.

Compliance with Ethical Standards

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent

Informed consent does not apply as this was a retrospective review with no identifying patient information.

Funding: Not applicable

Conflicts of interest Statement: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material:

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Code availability: Not applicable

REFERENCES

[1] Golightly, Lewis, et al. "Adoption of cloud computing as innovation in the organization." International Journal of Engineering Business Management 14 (2022): 18479790221093992.

[2] Mohamed, A., Hamdan, M., Khan, S., Abdelaziz, A., Babiker, S. F., Imran, M., & Marsono, M. N. (2021). Software-defined networks for resource allocation in cloud computing: A survey. Computer Networks, 195, 108151.

[3] Dwivedi, S. K., Yadav, J., Ansar, S. A., Khan, M., & Pandey, D. (2023, November). An investigation with emerging issues and preventive measures: Cloud computing perspective. In AIP Conference Proceedings (Vol. 2821, No. 1). AIP Publishing.

[4] Hentschel, R., Bley, K., & Lange, F. (2023, November). A Performance-Based Assessment Approach for Cloud Service Provider Selection. In Conference on e-Business, e-Services and e-Society (pp. 250-264). Cham: Springer Nature Switzerland.

[5] Al-Marsy, A., Chaudhary, P., & Rodger, J. A. (2021). A model for examining challenges and opportunities in use of cloud computing for health information systems. Applied System Innovation, 4(1), 15.

[6] Malik, Latesh, and A. Sandhya. Computing Technologies and Applications. United States: CRC Press, 2021.

[7] Landoll, Douglas. The security risk assessment handbook: A complete guide for performing security risk assessments. CRC press, 2021.

[8] El Kafhali, S., El Mir, I., & Hanini, M. (2022). Security threats, defense mechanisms, challenges, and future directions in cloud computing. Archives of Computational Methods in Engineering, 29(1), 223-246.

[9] Masood, Arooj, Demeke Shumeye Lakew, and Sungrae Cho. "Security and privacy challenges in connected vehicular cloud computing." IEEE Communications Surveys & Tutorials 22.4 (2020): 2725-2764.

[10] Parast, Fatemeh Khoda, et al. "Cloud computing security: A survey of service-based models." Computers & Security 114 (2022): 102580.

[11] Kotak, J., Habler, E., Brodt, O., Shabtai, A., & Elovici, Y. (2023). Information Security Threats and Working from Home Culture: Taxonomy, Risk Assessment and Solutions. Sensors, 23(8), 4018.

[12] Ullah, Raja Muhammad Ubaid. "An empirical study of establishing guidelines for evaluation and adoption of secure and cost effective cloud computing." (2023).

[13] Nandy, T., Noor, R. M., Kolandaisamy, R., Idris, M. Y. I., & Bhattacharyya, S. (2024). A review of security attacks and intrusion detection in the vehicular networks. Journal of King Saud University-Computer and Information Sciences, 101945.

[14] Shah, Varun, and Sreedhar Reddy Konda. "Cloud Computing in Healthcare: Opportunities, Risks, and Compliance." Revista Espanola de Documentacion Cientifica 16.3 (2022): 50-71.

[15] Sen, Amartya, and Sanjay Madria. "Analysis of a cloud migration framework for offline risk assessment of cloud service providers." Software: Practice and Experience 50.6 (2020): 998-1021.

[16] Sarker, Iqbal H. "AI-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems." SN Computer Science 3.2 (2022): 158.

[17] Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. Electronics, 11(2), 198.

[18] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. Electronics, 9(7), 1177.

[19] Adee, Rose, and Haralambos Mouratidis. "A dynamic four-step data security model for data in cloud computing based on cryptography and steganography." Sensors 22.3 (2022): 1109.

[20] Kure, Halima Ibrahim, Shareeful Islam, and Haralambos Mouratidis. "An integrated cyber security risk management framework and risk predication for the critical infrastructure protection." Neural Computing and Applications 34.18 (2022): 15241-15271.

[21] Gupta, L., Salman, T., Ghubaish, A., Unal, D., Al-Ali, A. K., & Jain, R. (2022). Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach. Applied Soft Computing, 118, 108439.

[22] Al Saleh, Reem, Maha Driss, and Iman Almomani. "CBiLSTM: A hybrid deep learning model for efficient reputation assessment of cloud services." IEEE Access 10 (2022): 35321-35335.

[23] Denis, R., and P. Madhubala. "Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems." Multimedia Tools and Applications 80.14 (2021): 21165-21202.

[24] Thabit, Fursan, et al. "A new lightweight cryptographic algorithm for enhancing data security in cloud computing." Global Transitions Proceedings 2.1 (2021): 91-99.

[25] Chinnasamy, P., et al. "Efficient data security using hybrid cryptography on cloud computing." Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020. Springer Singapore, 2021.

[26] Guan, Shaopeng, et al. "Hadoop-based secure storage solution for big data in cloud computing environment." Digital Communications and Networks 10.1 (2024): 227-236.

[27] El Kafhali, Said, Iman El Mir, and Mohamed Hanini. "Security threats, defense mechanisms, challenges, and future directions in cloud computing." Archives of Computational Methods in Engineering 29.1 (2022): 223-246.

[28] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101, 102031.

[29] Wei, Cheng-Long, and Gai-Ge Wang. "Hybrid annealing krill herd and quantum-behaved particle swarm optimization." Mathematics 8.9 (2020): 1403.

[30] Ullah, S., Zheng, J., Din, N., Hussain, M. T., Ullah, F., & Yousaf, M. (2023). Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey. Computer Science Review, 47, 100530.

[31] Jacob, I. Jeena, and P. Ebby Darney. "Artificial bee colony optimization algorithm for enhancing routing in wireless networks." Journal of Artificial Intelligence 3, no. 01 (2021): 62-71.

[32] Nguyen, Minh Tuan, and Kiseon Kim. "Genetic convolutional neural network for intrusion detection systems." Future Generation Computer Systems 113 (2020): 418-427.

[33] Ramaiah, M., Chandrasekaran, V., Ravi, V., & Kumar, N. (2021). An intrusion detection system using optimized deep neural network architecture. Transactions on Emerging Telecommunications Technologies, 32(4), e4221.

[34] Nayyar, S., Arora, S., & Singh, M. (2020, July). Recurrent neural network based intrusion detection system. In 2020 international conference on communication and signal processing (iccsp) (pp. 0136-0140). IEEE.

[35] Bhati, Bhoopesh Singh, and Chandra Shekhar Rai. "Analysis of support vector machine-based intrusion detection techniques." Arabian Journal for Science and Engineering 45.4 (2020): 2371-2383.

[36] Wahab, O. F. A., Khalaf, A. A., Hussein, A. I., & Hamed, H. F. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. IEEE access, 9, 31805-31815.

[37] Altigani, Abdelrahman, Shafaatunnur Hasan, Bazara Barry, Shiraz Naserelden, Muawia A. Elsadig, and Huwaida T. Elshoush. "A polymorphic advanced encryption standard–a novel approach." IEEE Access 9 (2021): 20191-20207.

[38] Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 12, 547-566.

[39] Mitra, S., Das, S., & Kule, M. (2021). Prevention of the man-in-the-middle attack on Diffie–Hellman key exchange algorithm: A review. In Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020 (pp. 625-635). Springer Singapore.