Computers, Materials & Continua
DOI:10.32604/cmc.2021.017199
Article

Tech Science Press

Cryptographic Based Secure Model on Dataset for Deep Learning Algorithms

Muhammad Tayyab[1,*], Mohsen Marjani[1], N. Z. Jhanjhi[1], Ibrahim Abaker Targio Hashim[2], Abdulwahab Ali Almazroi[3] and Abdulaleem Ali Almazroi[4]

[1]School of Computer Science and Engineering (SCE), Taylor's University Lake-Side Campus, Subang Jaya, 47500, Malaysia
[2]Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, 27272, UAE
[3]University of Jeddah, College of Computing and Information Technology at Khulais, Department of Information Technology, Jeddah, Saudi Arabia
[4]Department of Computer Science, Rafha Community College, Northern Border University, Arar, 91431, Saudi Arabia
[*]Corresponding Author: Muhammad Tayyab. Email: muhammadtayyab@sd.taylors.edu.my
Received: 23 January 2021; Accepted: 05 April 2021

Abstract: Deep learning (DL) algorithms have been widely used in various security applications to enhance the performances of decision-based models. Malicious data added by an attacker can cause several security and privacy problems in the operation of DL models. The two most common active attacks are poisoning and evasion attacks, which can cause various problems, including wrong prediction and misclassification of decision-based models. Therefore, to design an efficient DL model, it is crucial to mitigate these attacks. In this regard, this study proposes a secure neural network (NN) model that provides data security during model training and testing phases. The main idea is to use cryptographic functions, such as hash function (SHA512) and homomorphic encryption (HE) scheme, to provide authenticity, integrity, and confidentiality of data. The performance of the proposed model is evaluated by experiments based on accuracy, precision, attack detection rate (ADR), and computational cost. The results show that the proposed model has achieved an accuracy of 98%, a precision of 0.97, and an ADR of 98%, even for a large number of attacks. Hence, the proposed model can be used to detect attacks and mitigate the attacker motives. The results also show that the computational cost of the proposed model does not increase with model complexity.

Keywords: Deep learning (DL); poisoning attacks; evasion attacks; neural network; hash functions SHA512; homomorphic encryption scheme

1 Introduction

In modern machine learning (ML) and artificial intelligence (AI) models, learning algorithms provide numerous innovative features for daily life applications. Such advancements of ML algorithms have shown many results in real-world scenarios in solving the data-driven problems, such as prediction of a patient's data in the health care system [1], and system security logs for

security audit and unmanned aerial vehicles (UAVs) [2]. Deep learning (DL) has also achieved the highest maturity level and has been applied to numerous safety and security applications. The DL has been used in many crucial software applications, including the Internet of Things (IoT) [3], smart cities [4,5], modern education systems [6], surveillance models [7], vulnerability and malware detection [8], drone jets [9], robotics, and voice-controlled devices [10]. With the application of DL models, many real-life data-driven problems, such as speech recognition, medical image processing, automated cervical cancer detection [11], and others [12], can be easily solved. Many modern services and applications use data-driven approaches to automate their operation and provide different benefits to users [13]. In recent years, DL has yielded breakthroughs in many fields, such as learning algorithms [14]. However, DL algorithms have made a prosperous milestone in security-sensitive applications [15] and health-related prediction models [16], including detection of spam and malicious emails [17], fraud detection [18], and malicious intrusion detection [19]. The DL field includes a comprehensive range of different techniques, such as supervised learning algorithms, of which the most commonly used are neural networks (NNs) [20], support vector machine (SVM) [21], and decision trees [22]. However, it should be noted that originally, most well-known DL algorithms were not designed for adversarial environments, especially for security-critical applications [23].

The innovations and new features introduced by DL have caused many security problems, which can result in misclassification or wrong predictions of DL models. The most severe and challenging data security and privacy problems in DL models are caused by poisoning and evasion attacks. One of the most common causative attacks carried out during the training phase of learning algorithms is introducing carefully crafted "noise" or "poisoning" to training data. This attack is known as a poisoning attack, and it can mislead the learning process of a learning algorithm. Whereas, in the case of exploratory attacks that are known as evasion attacks, certain "good words" are injected into spam emails so that these spam emails will be labeled as non-spam emails and thus will bypass the spam detection system [24]. Therefore, a secure model is needed to provide data and model security in DL. To address this challenge, this study designs and develops a secure model for DL by introducing cryptographic functions for secure DL services. By using cryptographic functions, critical organizations, such as research centers in hospitals or fraud detection companies, can work securely with the end-users while ensuring security to all involved parties [25]. The proposed model follows the previous procedures but introduces hash function SHA512 [26] and homomorphic encryption (HE) scheme to provide data integrity and confidentiality in DL algorithms. The hash functions are used to provide data authenticity regardless of whether data are modified or remain in the original state. Since hash functions are one-way functions and generate a fixed length of alphanumeric values independent of the input string, the proposed model uses a property of digital signature of data to check data authenticity. The HE is used to provide data integrity and confidentiality in the learning process of a DL model [27]. The HE allows a DL model to perform mathematical operations over encrypted data but prevents input data from leaking information to a host model. Therefore, for an adversary, it is hard to break the security of HE.

1.1 Threat Model and Problem Statement

DL models play a vital role in classification and prediction tasks in different environments. Data used for training and testing of DL models are commonly gathered from numerous untrustworthy sources. Therefore, it is considered as a standard that a DL model should operate normally, consistent with outcomes, regardless of internal problems and complexity of the model. However, the primary motive of attackers is to obtain the information on data and a DL model by injecting

malicious data to subvert the normal working of the DL model. An attacker can manipulate the input data by inserting poisoned data that can divert predictions or lead to misclassification so that an intruder gains benefit. Hence, a secure model that can address security and privacy problems in DL models is urgently needed. The proposed secure model not only preserves data privacy but also provides model security with the help of common cryptographic schemes.

1.2 Problem Description

A universal threat, commonly known as a risk of data transmission, which can be caused either by side-channel attacks or by interception, is the main problem in DL algorithms. A strong cryptographic scheme provides strong measures against this threat by using encryption and signature schemes to secure data transmission through the network. However, it is hard to guarantee that data transmitted over the network have not been manipulated by attackers. There is a strong concern that an adversary can affect data and manipulate data for a DL model, for instance, via a poisoning attack. Also, an attacker can gain access to the DL model and subvert the training process, which can result in misclassification or wrong prediction. To overcome the mentioned threat, the HE scheme can be used to ensure data security and integrity. The HE allows operation on a ciphertext without decrypting the ciphertext, so a learning algorithm can use the ciphertext and perform prediction or classification. In this way, the third party does not have any access to the plain text, which guarantees data privacy.

1.3 Contributions

The main contribution of this works is the design of a secure NN model for DL algorithms that can preserve data privacy and provide data security in DL models. The contribution of this work can be summarized as follows:

(1) A secure NN model against poisoning and evasion attacks during the training and testing phases of a DL model is developed to ensure data security.
(2) Two cryptographic functions, the hash function SHA512 and HE scheme, are used to provide authenticity, integrity, and confidentiality of data.
(3) The proposed model is evaluated based on accuracy, precision, attack detection rate (ADR), and computational cost.
(4) The proposed model helps to maintain high accuracy and precision while ensuring appropriate ADR and lower computational cost compared to the original NN model.

The rest of the paper is organized as follows. In Section 2, the related literature on security problems and security attacks in DL models, which can greatly affect DL models' performances, is presented. In Section 3, a detailed description of the proposed secure model, including methods, evaluation matrix, experimental setup, and data used for the implementation of the proposed secure model, is provided. In Section 4, the results and limitations of the proposed model are discussed. Finally, in Section 5, the main conclusions are drawn, and future work directions are given.

2 Related Work

In recent decades, DL has enhanced significantly in solving many problems in the AI field. However, this has caused challenging scientific problems, such as brain construction [28], and has faced various security challenges. These security and privacy challenges have a great impact on DL models during the prediction and classification processes [29]. This study considers two types

of active attacks, poisoning attacks and evasion attacks, which have been regarded as the most challenging security attacks in DL models [30]. To address the problems caused by these two attach types, a secure model is developed. In the following, a few recent studies on the mentioned attacks are presented.

2.1 Security and Privacy Issues in DL Algorithms

As mentioned above, DL provides innovative features in learning models in various fields. For model training, DL requires a large amount of sensitive data to achieve high accuracy in classification and prediction [31]. Data used for model training face a number of security and privacy issues. To address these security issues, secure and private AI (SPAI) was proposed by Carlini et al. [32]. The SPAI aims to provide data security and privacy and offers a mechanism to mitigate the effects of adversarial attacks. However, this scheme significantly increases model complexity and computational cost. Caminero et al. [33] proposed a model that limits the effects of adversarial attacks using simple operations of the HE scheme, but it makes a DL model complex.

One of the most severe security concerns in the DL field is data poisoning with adversarial examples, which can mislead a DL model. Ovadia et al. [34] developed an outlier detection-based model to reduce the effects of optimal poisoning attack on the ML model performance. However, this model may constrain the prediction decision boundary significantly. Generally, data used for model training should be obtained from secure sources, but in practice, this is not always the case [35,36]. The deep neural networks (DNNs) also face many security problems due to using adversarial examples that behave normally for observers. In recent years, there have been a large number of reported attacks in DNNs, which has affected the training and testing of DNN models [37,38]. Papernot et al. [39] proposed an efficient model against security attacks that can be constructed using a highly effective classifier with the help of adversarial examples of DNN data. However, this malicious classification can cause additional constraints in adversarial examples, especially in the computer vision field. The potential defense mechanisms against crafted adversarial examples have also been evaluated.

2.2 Security Attacks

With the development of the AI field, learning algorithms have been widely explored by adversaries, and poisoning and evasion attacks have been further improved to achieve their goal of changing the learning data [40]. The spam filter [41], DNNs [42], and classifier systems [43] are common DL areas that are strongly affected by poisoning and evasion attacks. In addition to other features, security has been considered as one of the most critical features of DL models. According to the related literature, two major types of active security attacks are poisoning and evasion attacks, and they can affect DL models' performances significantly. For instance, in the poisoning attack, an adversary is involved in the learning phase of a DL model and tends to subvert certain processes as normal processes, while in the evasion attack, an adversary is engaged to sabotage the classification of a DL model during the model testing phase. The data used by an attacker to initiate the mentioned attacks are known as adversarial data. An attacker can use different data to realize malicious activity depending on an attack scenario of a DL model [44]. Generally, there are two main types of attack scenarios. In the first type, all model settings, including parameters and values of hyper-parameters, are available to an attacker, and such an attack is known as a whitebox attack [45]; this attack has a very high success rate of getting information from a targeted model. In the second type, an adversary has limited knowledge and

has no information on the model and its parameters, and this attack is known as a blackbox attack; this attack has a very low success rate of getting the information from a target model.

2.2.1 Poisoning Attacks

In poisoning attacks, attackers intentionally insert malicious data or add malicious noise to the training data to divert the normal learning process or to mislead or misclassify the training data toward the wrong prediction. An attacker can generate malicious noise by interpreting the output pattern of a target model, which is known as poisoning attacks [46]. Several methods for poisoning attacks have been launched against traditional DL algorithms, such as SVM and LASSO.

2.2.2 Evasion Attacks

In evasion attacks, the primary objective of an adversary is to add additional noise to the test data by analyzing the output pattern of a target model. An attacker can also inject malicious queries into data to get wanted information, and once the attacker generates an output pattern similar to that of the target model, the attacker can replace the original data with malicious data. This can be difficult while evaluating security-sensitive applications. In this case, the classifier of the target model will become a malicious classifier, which will result in incorrect classification results. In the case of geo-metrics, the evasion attacks replace the test data with adversarial data to sabotage the normal training process.

3 Methodology

In this section, the proposed secure model that can preserve data privacy and security during the training and testing phases of a DL model is described in detail. First, the methods used in the proposed model are introduced. Then, the evaluation criteria are defined, and the evaluation matrix is categorized into two major parts, which are performance and evaluation of the operation of the proposed model. Finally, the experimental verification of the proposed model is conducted, and the results of the proposed model are compared with those of the conventional NN model.

3.1 Methods

In the proposed model, there are three major phases: Phase 1 that includes applying hash function SHA512 and HE, Phase 2 that includes decryption and verification of data, and Phase 3 that includes training of a DL Algorithm for classification and prediction. The phases of the proposed model are presented in Fig. 1.

3.1.1 Phase 1

In Phase 1, the proposed secure model calculates the hash value by applying the hash functions, which is verified in Phase 3. The hash value can be considered as a digital signature and can be used for data verification. In this way, poisoning and evasion attacks can be easily detected, and it can be determined whether data have been compromised with additional noise.

The specific steps are as follows:

(a) Hash function SHA512 is applied to data get hash value H_0 and appended as a part of data attribute.
(b) Once the hash value is appended as part of data, it is encrypted using the HE encryption mechanism to ensure data privacy and stored to cloud storage for further processing.

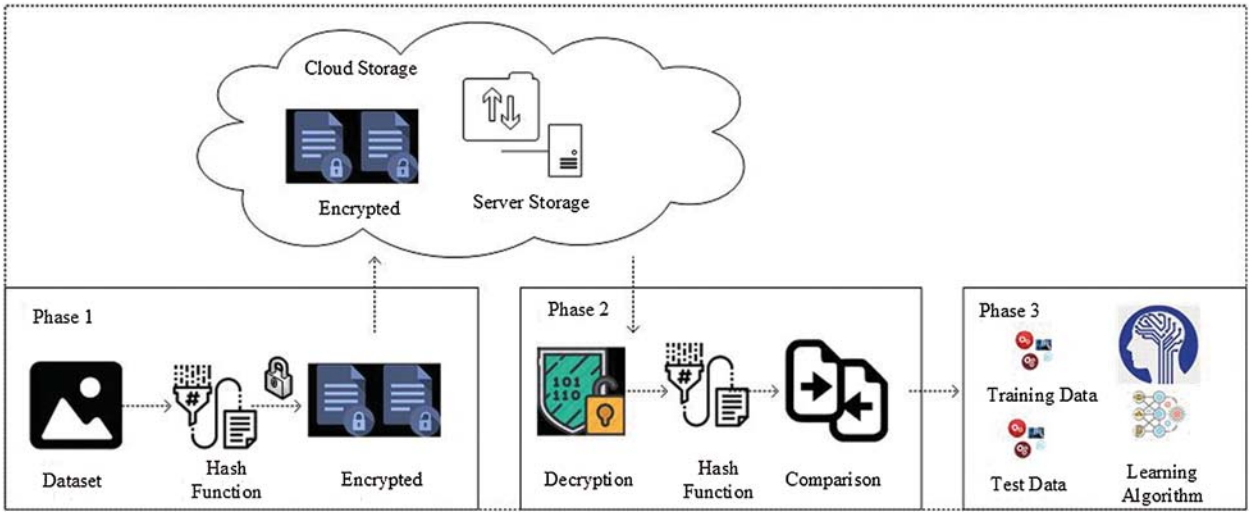


Figure 1: Phases of proposed model

Algorithm 1: Hash and Encrypt

Input: dataset D_0, *Key*
Output: D_{Hash}, $D_{Encrypted}$, *Upload*
1 Procedure HASH (D_0)
2 For $(i \leftarrow n)$ *do*
3 $H_0 = Hash(R_i[j])$ ∴ For each row of dataset
4 $D_{Hash} = D_0 \,||\, H_0$ ∴ Hash value in appended into the dataset
5 $D_{Encrypted} = Encrypt(D_{Hash}, Key)$ ∴ Using homomorphic encryption
6 return $D_{Encrypted}$
7 Upload the Encrypted data to cloud

3.1.2 Phase 2

In Phase 2, data are first retrieved from the cloud storage and then decrypted to obtain the original dataset. Next, the hash function SHA512 is again applied to the data, and the second hash value $H_1$is computed. It should be noted that while the second hash value is computed, the previous hash value is not used. Then, the hash values H_0 and H_1 are compared to evaluate data integrity.

The specific steps of Phase 2 are as follows:

(a) Encrypted data are retrieved from cloud storage.
(b) The HE is applied to the data to obtain the original data that have been outsourced.
(c) The second hash value H_1 is computed to check data integrity by comparison of this hash value with the previous hash value H_0.
(d) If the hash values match, the proposed model proceeds to Phase 3; otherwise, the model stops operation.

3.1.3 Phase 3

In Phase 3, after data verification in Phase 2, the proposed model performs data sampling, i.e., the data are split into training and test data.

The specific steps of Phase 3 are as follows:

(a) Split data into training and test data.
(b) Normalized data to obtain the image pixel values between +0.5 and -0.5 by using Eqs. (1) and (2) respectively.

$$train\_image = ((train\_image/255) - 0.5) \tag{1}$$

$$test\_image = ((test\_image/255) - 0.5) \tag{2}$$

(c) After data normalization, train the DL model with the training data.
(d) Test the trained DL model using the test data to evaluate the performance of the trained DL model.

Algorithm 2: Hash and Decrypt

Input: $D_{Encrypted}$, key
Output: Clean Dataset D_{clean} $Attack\_Rate$

Procedure $Decrpt\left(D_{Encrpted}, key\right)$
$D_{Decrypt} = Decrpt\left(D_{Encrpted}, key\right)$
For $\left(i \leftarrow n\right) do$
  $H_1 = Hash\left(R_i[j]\right)$ ∴. For each row of dataset
  $D_{Hash\_1} = D_1 \,||\, H_1$ ∴. Hash value in appended again into the dataset
If $(H_0 == H_1)$ ∴. Comparison of Two hash values for authentication
  $Rate = (False\_obs/Total\_obs)$ ∴. Computer Attack Detection rate
  Proceed toward phase 3
  $D_{clean} = \mathrm{R}\_Col\left(D_{hash\_1}\right)$ ∴. Remove the hash columns
  Return D_{clean}
Else
  Return "The dataset has been intruded maliciously"
  Return D_{clean}, $Attack\_Rate$

3.2 Evaluation Matrix

The evaluation of the proposed model is conducted using the evaluation matrix. The evaluation matrix is divided into two sub-categories, performance evaluation and execution evaluation. The most common evaluation metrics used in the state-of-the-art literature are used in the model evaluation process. The accuracy of predicting the correct labels as well as adversarial labels is also analyzed. The conventional NN model is used to further evaluate the proposed model via the comparison of the models on the same data. In addition to the prediction accuracy, the proposed model is evaluated based on precision and ADR. The performance evaluation procedure and parameters are described in the following.

Algorithm 3: Data Sampling and Model training

Input: D_{clean}, $Model_{Param} D_{train}$, D_{test}
Output: $Model_{Trained} Model_{Evaluated}$ Accuracy, Precision
1 Procedure $Data\_Sampling\,()$
2 $D_{train}, D_{test} = Data\_Sampling\,(D_{clean})$
3 $D_{train} = ((D_{train}/255) - 0.5)$ ∴ Data Normalization
4 $D_{test} = ((D_{test}/255) - 0.5)$ ∴ Data Normalization
5 $Model_{train} = Learning\_Model\,(D_{train})$
6 $Model_{Evaluated} = Evaluation\_Model\,(D_{test})$
7 $Accuracy = (tp + tn)\,/\,(tp + tn + fp + fn) * 100$
8 $precision = tp/\,(tp + fp)$
9 Return Accuracy, Precision

3.2.1 Performance Evaluation

a) Wall-Clock Running Time

The running time refers to the time a system requires to execute a certain program. This parameter is considered as a default parameter, as reported in the previous literature [47]. It depends on hardware, which means that it is directly dependent on the system configuration, including available memory space and computational power of the system. It is also dependent on the encryption scheme used in a model. The running time of the proposed model is $O(n^2)$, and it is computed asymptotically.

b) Hardware/Software Setup

The proposed model was experimentally verified using a PC with an Intel Core i5 3.5 GHz CPU and 16 Gb of RAM running on a Windows 10 operation platform. The HE library named *cryptography.fernet* [48] was used. The proposed cryptographic function was compared with the previous cryptographic functions.

3.2.2 Execution Evaluation

a) Accuracy

Accuracy has been commonly used as an evaluation metric of NN-based classification models. The accuracy is regarded as the most reliable metric, and it shows how well a model process input data. Typically, accuracy is expressed in percentage. The accuracy can be regarded as a fraction of predictions that a model predicted correctly [49,50]. The accuracy is given by Eqs. (3) and (4):

$$Accuracy\,(\%) = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \times 100 \tag{3}$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{4}$$

where tp denotes true positive, tn denotes true Negative, fp stands for false positive, and fn stands for false negative observations.

b) Precision

Precision is defined as a ration of the number of correct positively classified example to the number of all the positive label examples classified by the model [51,52]. The precision has a value between zero and one. The precision depicts how well the model behaves when it is exposed against adversarial data or any attack scenario. The precision of a model can be expressed as in Eq. (5):

$$precision = \frac{tp}{tp + fp} \tag{5}$$

where tp: True positive, and fp: False Positive

3.2.3 Attack Detection Rate

Attack Detection Rate (ADR), which represents the ratio of true positive and the total outcomes of the model. Given below is the representation of ADR in Eq. (6):

$$Attack\ \ Detection\ \ Rate = \frac{tp}{tp + fn} \tag{6}$$

where tp and fn are the representations of true positive and false negative [53].

3.3 Experiment

The proposed model is verified by experiments. However, since the proposed secure model uses two cryptographic functions to provide data privacy and security, a detailed explanation of cryptographic functions is given first.

3.3.1 Hash Function (SHA-512)

Hashing algorithms have been used in various fields, such as internet security and digital certificates. The hash functions play a vital role in the field of cryptography for providing digital security to online content [54]. Usually, hash functions take an arbitrary length of the input stream and generate a fixed-length hash value called the hash digest that consists of alphanumeric values and does not have any particular meaning. The output of hash functions should meet certain conditions, which are as follows:

(i) Uniform distribution: As the output of hash functions has a fixed length, and the input of hash functions can vary in length, different input values should not generate the same output stream.
(ii) Fixed length: The output of hash functions should have a unique value and fixed length.
(iii) Collision Resistance: Hash functions should generate similar output values for different inputs to make it difficult to distinguish two different hash values.

The proposed model uses the SHA-512 hash function. This function takes an arbitrary length as input data and generates a 512-bit long alphanumeric value as an output. Hash functions have been widely used as digital signatures for digital content. The proposed model uses hash functions to provide data authenticity. For instance, data can be modified by an attacker by adding malicious data to the original, clean data to obtain the information on learning algorithms. In such a case, a hash function can detect malicious activity on data.

The MNIST dataset [55] that contains pixel values of handwritten digits of 28 × 28 images was used in the experiment. The proposed model computed hash values of all features of a

single label, including the label, and appended to the data as additional features. This process was repeated for each label of the MNIST dataset.

3.3.2 Homomorphic Encryption (HE)

The HE scheme is used to ensure data security and maintain data integrity. The HE preserves the structure of a plaintext message, so different mathematical operations, such as addition and multiplication, can be conducted over the encrypted data that is commonly known as a ciphertext [56]. Similar to other security assurance schemes, the HE includes three functions denoted as *Gen, Enc*, and *Dec*, which are used for key generation, encryption, and decryption, and defined by Eqs. (7) and (8), respectively.

$$cipher\_text = Dec\,(s_k, plain\_text) \tag{7}$$

$$plain\_text = Enc\,(p_k, cipher\_text) \tag{8}$$

In 1978, Kaaniche et al. [57] used the HE for the very first time, and since then, it has been improved by many researchers. However, most of the encryption functions have certain limitations; for instance, in the Pailier cryptosystem, there is only the addition operation. This type of encryption is commonly known as somewhat homomorphic encryption (SHE) [58]. The first fully homomorphic encryption (FHE) was introduced in 2009 after the successful removal of additional noise in the HE. The FHE not only can support a circuit with an arbitrary depth but can also conduct multiple operations while performing encryption and decryption. However, this significantly increases the computation cost, which makes the FHE impractical for real-world applications. By introducing certain improvements into the original HE, the leveled homomorphic encryption (LHE) has been proposed, which makes the HE faster and reduces the computational cost. The LHE has the advantage of not using bootstrapping, thus allowing circuits to have a depth lower than a certain threshold. In terms of computational cost, if the number of steps is known, then the LHE can be used instead of the FHE. To summarize, using a limited number of operations, such as addition and multiplication, can decrease the computational cost and increase the efficiency of HE schemes, which has been used in the proposed NN model.

3.3.3 Model Setting

A simple NN was developed and denoted as the original NN model. The proposed model was a sequential model that consisted of two layers with 64 neurons having the ReLU activation function and one layer with 10 neurons having the softmax activation function. The initial model parameters were fine-tuned by the optimization using "adam" optimizer and "categorical_crossentropy" loss function for multiclass classification. The "categorical_crossestropy" was used as a loss function because it is very successful in the classification of multiple classes. The accuracy was used as an evaluation metric. The proposed model was developed using the MNIST dataset with a total of 60,000 images, of which 50,000 images were used for model training, and the remaining 10,000 images were used for model testing. In the model training process, the maximum number of epochs was set to five, and batch size was set to 32.

3.4 Dataset

The handwritten numerical digits 0–9 of the MNIST dataset, which contained 28 × 28 grey-scale images of ten different classes, were used for model development (10-class classification task). This dataset has been used as a general dataset for the training and testing of many DL algorithms. It consists of 50,000 training images and 10,000 images test images. Although the

MNIST is a simple dataset, it has been the standard benchmark for homomorphic inference tasks [59] and has been used for classification and prediction tasks by many DL models.

4 Results and Discussion

This section presents the results of the proposed model. The accuracy, precision, and ADR metrics were used for verification of the proposed model. The proposed model was compared with the original NN model, which was denoted as a benchmark. Compared to the original model, the proposed model had an additional layer that included the cryptographic function to provide data security during model training and testing.

4.1 Experimental Correctness

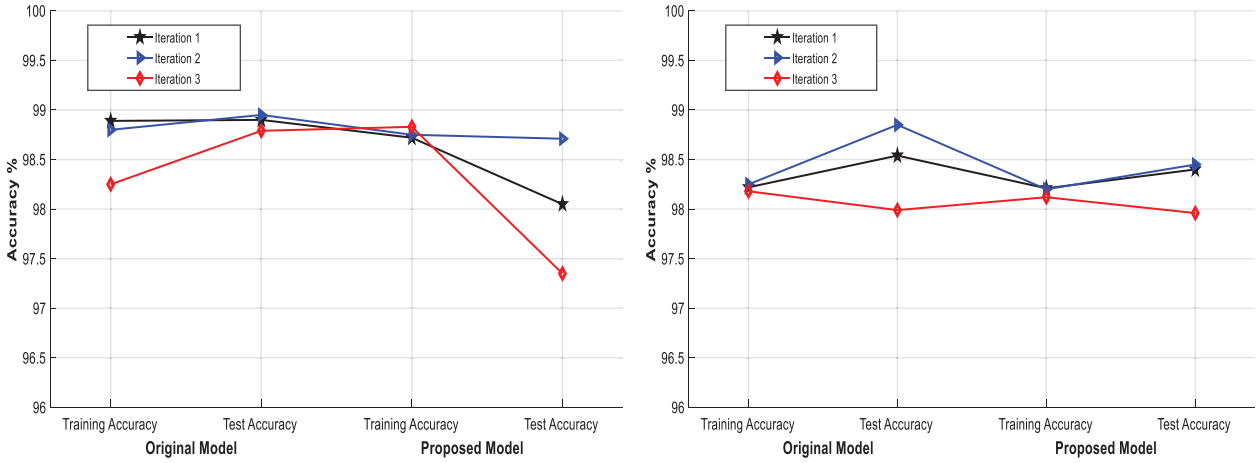
The model parameters were set so that the output after the decryption function must be correct. The encryption function was applied to both training and test data simultaneously. Based on the results, there was no accuracy loss on the plain text; the accuracy results were 98.89% for training and 98.90 for test data. The precision error, calculated compared to the decrypted outputs, was 0.05% [60]. The error was caused by the mathematical computation that could create variations in floating points for encryption and decryption functions. However, this error did not affect the accuracy of the proposed model significantly.

4.2 Accuracy Results

The proposed model was developed using the latest version of the Python programing language. The accuracy was computed for both the proposed model and the original NN model. The results showed that the accuracy of the proposed model was almost the same as that of the original NN model, having only a minor difference that was caused by the computational complexity, which was due to mathematical operations of encryption and decryption. This shortcoming can be overcome by reducing the number of operations in the encryption and decryption processes. We have mentioned the experimental correction in the previous section to elaborate on a minor difference in terms of accuracy. The main goal was to maintain high accuracy level of the proposed model while achieving a high attack detection rate. The accuracy difference between the two models is shown in Fig. 2a, where the two models achieved similar accuracies. Moreover, different scenarios were created based on the recent literature for evaluation of the proposed model's accuracy. Since the proposed model used the cryptographic function, HE for encryption and decryption, the DL model was used on an encrypted dataset. The DL model could be applied to an encrypted dataset because the HE allowed operations over encrypted data. In this regard, the accuracy of the proposed model was calculated when it was trained with an encrypted dataset, and the obtained result is presented in Fig. 2b, where, in this case, the accuracy of the proposed model showed a slight decrease because of mathematical computations and floating values.

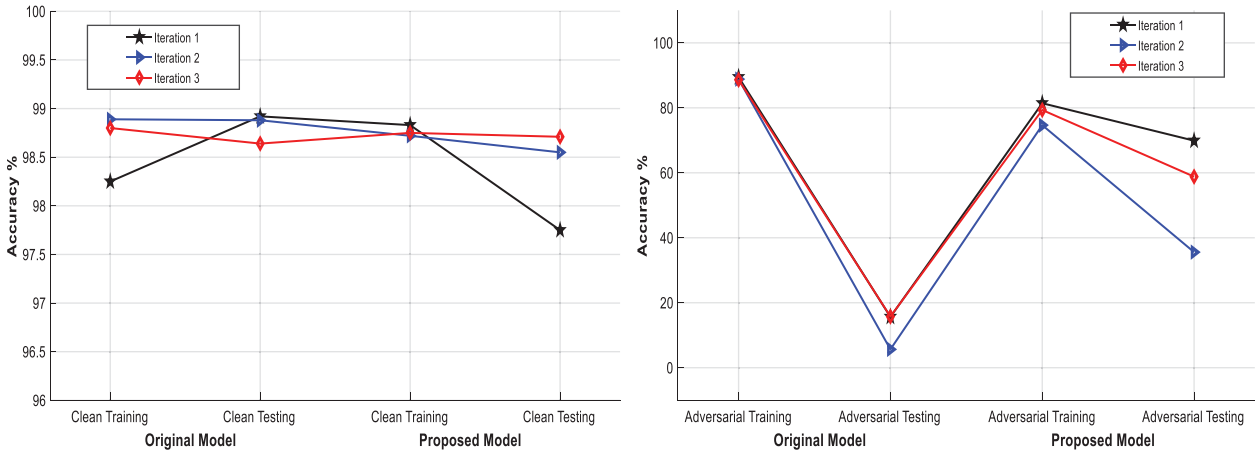
The proposed model was also tested under the attacker scenarios using adversarial data of FGSM attack (poisoning attack) and JSMA attack (evasion attack). The results of the proposed model when it was exposed to the poisoning and evasion attacks are presented in Figs. 3 and 4, respectively. The training and test results of the two models were compared for the cases without and with attacks. In the experimental scenario, the attacker injected malicious data to achieve high accuracy. The results confirmed that the attacker could achieve high accuracy by injecting equivalent adversarial data, but when the model was evaluated using the test dataset, the accuracy dropped slightly, as presented in Figs. 3a and 3b, and Figs. 4a and 4b, respectively. The accuracy comparison of the proposed model and the existing HME cryptographic techniques used in NN

models to provide data security is presented in Tab. 1. As shown in Tab. 1, the accuracy of the proposed model was of the same level as those of the existing models, and a minor drop in the accuracy of the proposed model was due to additional mathematical operations in encryption and decryption. In addition, the proposed model has achieved better accuracy than the existing techniques. The results of the proposed model show that the accuracy is improved while the computational cost is not increased significantly.



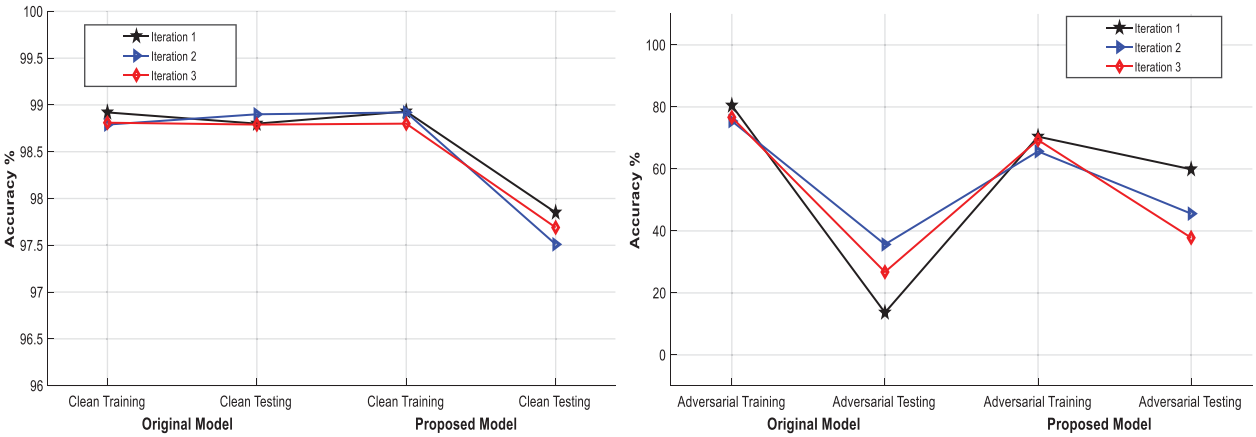
(a) (b)

Figure 2: Accuracy results (a) without encryption (b) with encryption



(a) (b)

Figure 3: Accuracy results (a) without FGSM (b) with FGSM (poisoning attack)



(a) (b)

Figure 4: Accuracy results (a) without JSMA (b) with JSMA (evasion attacks)

4.3 Precision Results

The precision is the ratio of totally positively predicted to the total prediction of the model. We have provided the precision values of proposed model and compared this value to the precision value of original model. In Tab. 2, we have provided the results, which shows that there is slightly difference between the values because of additional security layer.

4.4 Attack Detection Rate Results

The ADR results of the proposed model are given in Tab. 3. As mentioned above, the proposed model was tested using two widely-used attack types, FGSM (poisoning attack) [61] and JSMA (evasion attack) [62]. The results showed that the proposed model could detect the attacks efficiently while keeping the accuracy at a relatively high level; namely, the accuracy of the proposed model did not drop below the threshold level. To the best of the authors' knowledge, the proposed method has been the only method that can identify these attacks while achieving good accuracy. In the proposed model, the threshold level for critical systems was set to a fixed value, and it was assumed that the attacker's primary goal was to obtain the information on data as well as the DL model. Hence, if the ADR was greater than the threshold, the proposed model would terminate prediction or classification and return to the data sampling step. Analyzing this use case is very useful since it is very common in many critical systems, including health care systems and UAVs.

Table 1: Comparison with existing HE methods with proposed model

Parameters	Faster CryptoNets [59]	CryptoNets [59]	CryptoDL-1 [63]	CryptoDL-2 [63]	Proposed method using fernet
Model accuracy (%)	98.71	98.96	98.46	99.72	98.80
	98.65	98.95	98.52	99.62	98.90
	98.85	98.90	98.72	99.52	98.89

Table 2: Precision values of proposed model and original model

Parameter	Original model	Proposed model
Precision	0.96	0.96
	0.94	0.94
	0.97	0.98

Table 3: Attack detection rate of proposed model

Parameter	Proposed model
Attack detection rate (%)	99.1
	98.8
	98.9

4.5 Computational Cost

The computation cost is depended on the system configuration as well as the processing power of a machine. In the proposed model, although cryptographic functions are used, the overall computation has increased but not significantly. The computational cost is the only limitation of the proposed model, but it can be reduced by decreasing the number of operations and using a different optimization solution. The computational cost of the proposed model is defined by Eqs. (9)–(13):

$$PM = Hash + Encryption/Decryption + Hash + Model \tag{9}$$

$$= cn + n^2 + cn + cn\log n \tag{10}$$

$$= 2cn + n^2 + cn\log n \tag{11}$$

$$\begin{aligned} &= \Theta\left(n\log n + n^2\right) \\ &or \\ &= \Theta\left(n^2\right) \end{aligned} \tag{12}$$

$$PM = \Theta\left(n^2\right) \tag{13}$$

where PM represent the proposed model, cn denotes the cost, and Θ shows the tighter analysis of the proposed model. The computational cost of the proposed model is not greater than $\Theta\left(n^2\right)$, and the computational cost of the original model is $\Theta(n\log n)$. Hence, there is a slight increase in the computational cost of the proposed model compared to the original model.

5 Conclusion

The DL has become one of the research hotspots because of its decision-based problem-solving nature in daily-life applications. In the DL model design, security and privacy concerns are the main challenges. Namely, an attacker can consciously add noise to the data, which can result in misclassification or wrong prediction. Therefore, it is important to address security and

privacy problems before designing and applying a DL model. In this study, two major types of attacks, poisoning and evasion attack, are considered, and a secure NN model that can provide data security is proposed. The proposed model uses two cryptographic functions, the hash function SHA512 and the HE schemes, to maintain integrity, confidentiality, and authenticity of data. The results have been calculated in terms of accuracy, precision, ADR, and computational cost. The result has provided an accuracy of 98% and a precision of 0.97 level as compared with the original benchmark model. The proposed model is verified by the experiments, and the experimental results show that the proposed model can achieve high ADR even under a larger number of attacks. Moreover, although the proposed model has additional operations, the computational cost of the proposed model is still in the acceptable range. Therefore, the proposed model can resolve privacy and security problems in the case of poisoning and evasion attacks. In future work, other types of security attacks, such as model extraction and model inversion attack, will be considered to further evaluate the robustness of the proposed model.

Acknowledgement: The authors would also like to thank the Taylors University for their support in conducting the experiment.

Data Availability: The data used to support the findings of this study are available from the corresponding author upon request.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Shickel, P. J. Tighe, A. Bihorac and P. Rashidi, "Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, 2018.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[3] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman and A. S. Ahmed, "Internet of things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, pp. 1674–1703, 2020.

[4] A. N. Muhammad, A. M. Aseere, H. Chiroma, H. Shah, A. Y. Gital *et al.,* "Deep learning application in smart cities: Recent development, taxonomy, challenges and research prospects," *Neural Computing and Applications*, vol. 1, pp. 1–37, 2020.

[5] M. Bilal, R. S. A. Usmani, M. Tayyab, A. A. Mahmoud, R. M. Abdalla *et al.,* "Smart cities data: Framework, applications, and challenges," in *Handbook of Smart Cities*, Ch. 36, 1st ed., London, United Kingdom: Springer, pp. 1–29, 2020.

[6] R. S. A. Usmani, A. Saeed and M. Tayyab, "Role of ICT for community in education during COVID-19," in *ICT Solutions for Improving Smart Communities in Asia*, 1st ed., Pennsylvania, USA: IGI Global, pp. 125–150, 2021.

[7] J. Xu, "A deep learning approach to building an intelligent video surveillance system," *Multimedia Tools and Applications*, vol. 1, pp. 1–21, 2020.

[8] M. Bilal, M. Marjani, M. I. Lali, N. Malik, A. Gani *et al.,* "Profiling users' behavior, and identifying important features of review helpfulness," *IEEE Access*, vol. 8, pp. 77227–77244, 2020.

[9] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem *et al.,* "Big IoT data analytics: architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed *et al.,* "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[11] Z. Alyafeai and L. Ghouti, "A fully-automated deep learning pipeline for cervical cancer classification," *Expert Systems with Applications*, vol. 141, pp. 112951–112991, 2020.

[12] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[13] S. Ali, Y. Hafeez, N. Jhanjhi, M. Humayun, M. Imran *et al.,* "Towards pattern-based change verification framework for cloud-enabled healthcare component-based," *IEEE Access*, vol. 8, pp. 148007–148020, 2020.

[14] M. Bilal, A. Gani, M. I. U. Lali, M. Marjani and N. Malik, "Social profiling: A review, taxonomy, and challenges," *Cyberpsychology, Behavior, and Social Networking*, vol. 22, no. 7, pp. 433–450, 2019.

[15] D. K. Alferidah and N. Jhanjhi, "A review on security and privacy issues and challenges in internet of things," *International Journal of Computer Science and Network Security*, vol. 20, no. 4, pp. 263–286, 2020.

[16] S. K. Saini, V. Bansal, R. Kaur and M. Juneja, "ColpoNet for automated cervical cancer screening using colposcopy images," *Machine Vision and Applications*, vol. 31, no. 3, pp. 1–15, 2020.

[17] A. Baccouche, S. Ahmed, D. Sierra-Sosa and A. Elmaghraby, "Malicious text identification: Deep learning from public comments and emails," *Information: An International Interdisciplinary Journal*, vol. 11, no. 6, pp. 312, 2020.

[18] H. Najadat, O. Altiti, A. A. Aqouleh and M. Younes, "Credit card fraud detection based on machine and deep learning," in *Proc. IEEE, 2020 11th Int. Conf. on Information and Communication Systems*, Irbid, Jordan, pp. 204–208, 2020.

[19] K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi and K. Vijayakumar, "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, pp. 5242–5248, 2021.

[20] G. Ahmad, S. Alanazi, M. Alruwaili, F. Ahmad, M.-A. Khan *et al.,* "Intelligent ammunition detection and classification system using convolutional neural network," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2585–2600, 2021.

[21] A. Al-Dhamari, R. Sudirman and N. H. Mahmood, "Transfer deep learning along with binary support vector machine for abnormal behavior detection," *IEEE Access*, vol. 8, pp. 61085–61095, 2020.

[22] S. Jia, P. Lin, Z. Li, J. Zhang and S. Liu, "Visualizing surrogate decision trees of convolutional neural networks," *Journal of Visualization*, vol. 23, no. 1, pp. 141–156, 2020.

[23] S. Gerasimou, H. F. Eniser, A. Sen and A. Cakan, "Importance-driven deep learning system testing," in *Proc. IEEE/ACM, 2020 42nd Int. Conf. on Software Engineering*, Seoul, South Korea, pp. 702–713, 2020.

[24] Y. Wang, B. Liu, H. Wu, S. Zhao, Z. Cai *et al.,* "An opinion spam detection method based on multi-filters convolutional neural network," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 355–367, 2020.

[25] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman *et al.,* "Lightweight authenticated-encryption scheme for Internet of Things based on publish-subscribe communication," *IEEE Access*, vol. 8, pp. 60539–60551, 2020.

[26] M. Lim, A. Abdullah, N. Jhanjhi and M. K. Khan, "Situation-aware deep reinforcement learning link prediction model for evolving criminal networks," *IEEE Access*, vol. 8, pp. 16550–16559, 2019.

[27] K. Huang, X. Liu, S. Fu, D. Guo and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 1–15, 2019.

[28] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung *et al.,* "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168, 2013.

[29] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," *Journal of Chemical Information and Modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[30] G. Nguyen, S. Dlugolinsky, V. Tran and Á.L. García, "Deep learning for proactive network monitoring and security protection," *IEEE Access*, vol. 8, pp. 19696–19716, 2020.

[31] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico *et al.,* "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, pp. 1254806–1254816, 2015.

[32] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. ACM 10th Workshop on Artificial Intelligence and Security*, Texas, USA, pp. 3–14, 2017.

[33] G. Caminero, M. Lopez-Martin and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, vol. 159, pp. 96–109, 2019.

[34] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley *et al.,* "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Proc. 33rd Conf. on Neural Information Processing Systems*, Vancouver, Canada, pp. 13991–14002, 2019.

[35] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE, 2018 Security and Privacy Workshops*, San Francisco, CA, USA, pp. 1–7, 2018.

[36] N. A. Ghani, S. Hamid, I. A. T. Hashem and E. Ahmed, "Social media big data analytics: A survey," *Computers in Human Behavior*, vol. 101, pp. 417–428, 2019.

[37] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik *et al.,* "Practical black-box attacks against machine learning," in *Proc. ACM on Asia Conf. on Computer and Communications Security*, New York, United States, pp. 506–519, 2017.

[38] F. Altaf, S. Islam, N. Akhtar and N. K. Janjua, "Going deep in medical image analysis: Concepts, methods, challenges and future directions," *IEEE Access*, vol. 7, pp. 99540–99572, 2019.

[39] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE, 2016 Symp. on Security and Privacy*, San Jose, USA, pp. 582–597, 2016.

[40] Z. Guan, L. Bian, T. Shang and J. Liu, "When machine learning meets security issues: A survey," in *Proc. IEEE, 2018 Int. Conf. on Intelligence and Safety for Robotics*, Shenyang, China, pp. 158–165, 2018.

[41] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou *et al.,* "DUP-Net: Denoiser and upsampler network for 3D adversarial point clouds defense," in *Proc. IEEE Int. Conf. on Computer Vision*, Seoul, South Korea, pp. 1961–1970, 2019.

[42] T. Huang, Q. Zhang, J. Liu, R. Hou, X. Wang *et al.,* "Adversarial attacks on deep-learning-based SAR image target recognition," *Journal of Network and Computer Applications*, vol. 162, no. 1, pp. 102632, 2020.

[43] X. Cao and N. Z. Gong, "Mitigating evasion attacks to deep neural networks via region-based classification," in *Proc. 33rd Annual Computer Security Applications Conf.*, New York, United States, pp. 278–287, 2017.

[44] Y. Li, H. Li, G. Xu, T. Xiang, X. Huang *et al.,* "Toward secure and privacy-preserving distributed deep learning in fog-cloud computing," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11460–11472, 2020.

[45] T. Huang, Q. Zhang, J. Liu, R. Hou, X. Wang *et al.,* "Adversarial attacks on deep-learning-based SAR image target recognition," *Journal of Network and Computer Applications*, vol. 162, pp. 102632–102944, 2020.

[46] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb and S. Mahmood, "Cyber security threats and vulnerabilities: A systematic mapping study," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 3171–3189, 2020.

[47] F. Boemer, A. Costache, R. Cammarota and C. Wierzynski, "nGraph-HE2: A high-throughput framework for neural network inference on encrypted data," in *Proc. of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pp. 45–56, 2019.

[48] W. Liang, D. Zhang, X. Lei, M. Tang, K.-C. Li *et al.,* "Circuit copyright blockchain: Blockchain-based homomorphic encryption for IP circuit protection," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 1, 2020.

[49] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert *et al.,* "Deep learning for cellular image analysis," *Nature Methods*, vol. 16, no. 1, pp. 1233–1246, 2019.

[50] P. Pławiak, M. Abdar and U. R. Acharya, "Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring," *Applied Soft Computing*, vol. 84, pp. 105740, 2019.

[51] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[52] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen *et al.,* "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods," *Applied Soft Computing*, vol. 86, pp. 105836, 2020.

[53] A. Lasisi, R. Ghazali and T. Herawan, "Application of real-valued negative selection algorithm to improve medical diagnosis," in *Applied Computing in Medicine and Health*. Ch. 11, Sec. 11, 1st ed., Morgan Kaufmann, Waltham (MA): Elsevier, pp. 231–243, 2016.

[54] P. Kaplesh, "Cryptography security services: Network security, attacks, and mechanisms," in *Impact of Digital Transformation on Security Policies and Standards*. Ch. 5, 1st ed., Pennsylvania, USA: IGI Global, pp. 63–79, 2020.

[55] K. Cheng, R. Tahir, L. K. Eric and M. Li, "An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 13725–13752, 2020.

[56] A. Shafee and T. A. Awaad, "Privacy attacks against deep learning models and their countermeasures," *Journal of Systems Architecture*, vol. 1, pp. 101940–101949, 2020.

[57] N. Kaaniche, M. Laurent and S. Belguith, "Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey," *Journal of Network and Computer Applications*, vol. 171, no. 1, pp. 102807–102839, 2020.

[58] A. Wood, K. Najarian and D. Kahrobaei, "Homomorphic encryption for machine learning in medicine and bioinformatics," *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–35, 2020.

[59] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig *et al.,* "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. on Machine Learning*, New York, USA, pp. 201–210, 2016.

[60] Y. Lee, J.-W. Lee, Y.-S. Kim and J.-S. No, "Near-optimal polynomial for modulus reduction using l2-norm for approximate homomorphic encryption," *IEEE Access*, vol. 8, pp. 144321–144330, 2020.

[61] L. Gao, Q. Zhang, J. Song, X. Liu and H. T. Shen, "Patch-wise attack for fooling deep neural network," in *Proc. European Conf. on Computer Vision*, Glasgow, UK: Springer, pp. 307–322, 2020.

[62] A. U. H. Qureshi, H. Larijani, M. Yousefi, A. Adeel and N. Mtetwa, "An adversarial approach for intrusion detection systems using Jacobian Saliency Map Attacks (JSMA) Algorithm," *Computers*, vol. 9, no. 3, pp. 58, 2020.

[63] Q. Lou and L. Jiang, "SHE: A fast and accurate deep neural network for encrypted data," in *Proc. 33rd Conf. on Neural Information Processing Systems*, Vancouver, Canada, Springer, vol. 1, pp. 10035–10043, 2019.