IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2021, accepted December 21, 2021, date of publication January 4, 2022, date of current version January 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140342

A Deep Bidirectional LSTM-GRU Network Model for Automated Ciphertext Classification

EZAT AHMADZADEH, HYUNIL KIM, ONGEE JEONG, NAMKI KIM, AND INKYU MOON, (Member, IEEE)
Department of Robotics Engineering, DGIST, Hyeonpung-myeon, Dalseong-gun, Daegu 42988, South Korea

Corresponding author: Inkyu Moon (inkyu.moon@dgist.ac.kr)

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korean Government (MSIT) through the Research on AI-based Cryptanalysis and Security Evaluation under Grant 2020-0-00126.

ABSTRACT Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are a class of Recurrent Neural Networks (RNN) suitable for sequential data processing. Bidirectional LSTM (BLSTM) enables a better understanding of context by learning the future time steps in a bidirectional manner. Moreover, GRU deploys reset and update gates in the hidden layer, which is computationally more efficient than a conventional LSTM. This paper proposes an efficient network model based on deep BLSTM-GRU for ciphertext classification aiming to mark the category to which the ciphertext belongs. The proposed model performance was evaluated using well-known evaluation metrics on two publicly available datasets encrypted with various classical cipher methods and performance was compared against one-dimensional convolutional neural network (1D-CNN) and various other deep learning-based approaches. The experimental results showed that the BLSTM-GRU cell unit network model achieved a high classification accuracy of up to 95.8%. To the best of our knowledge, this is the first time an RNN-based model has been applied for the ciphertext classification.

INDEX TERMS Recurrent neural networks, bidirectional long short-term memory, gated recurrent unit, ciphertext classification, 1D-convolutional neural networks.

I. INTRODUCTION

With the increasing rate of data transfer over the internet, system security is becoming one of the most important issues for information exchange [1], [2]. The system security can be subdivided into cryptography and cryptanalysis. The purpose of a cryptosystem is to provide security by decorrelating the plaintexts and ciphertexts and making the plaintexts unreadable [3]–[5]. Cryptanalysis investigates the weakness of the cryptosystem to ensure system security. In cryptanalysis, the attacker tries to recover the original form of a secured message by analyzing hidden patterns in data or finding the secret key. One of the main ways to attack the cryptosystem is to analyze the hidden data patterns to reveal the main information of the ciphertext [1], [6], [7]. This is referred to as building a structured knowledge representation by extracting the features from the ciphertext normally using machine learning algorithms to process the information confined in the ciphertext. This technique includes machine learning, statistics, computational linguistics, and information retrieval [8]. Another way is to find the secret key to recover the original message. Ciphertext classification aiming to mark the category to which the ciphertext belongs can help an attacker reveal the subject of the information being exchanged. Ciphertext classification is a supervised learning task, where the machine learning algorithms are trained with a set of labeled data from different classes using features extracted from the document. On the other hand, different types of information with distinctive features can be involved in classification tasks.

An artificial neural network (ANN) is a commonly used method for many recognition, classification tasks, and cryptanalysis [9]–[11]. Many ANN-based and evolutionary approaches have been used in literature for cryptanalysis [12]–[17]. Convolutional neural networks (CNNs) are a powerful machine learning approach introduced several years ago. Recent advances in CNNs have demonstrated remarkable performance in different data processing tasks applied

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

mainly to medical image analysis [18], [19]. By network training on a set of annotated data, CNNs can extract the hidden pattern in data with remarkable accuracy. On the other hand, ANN and CNN-based methods are limited by the lack of sequential data processing ability considering unique characteristics of the ciphertext [20]–[22]. The long short-term memory (LSTM), recurrent neural network (RNN), and gated recurrent unit (GRU) cell units have been established for sequential data processing, which takes temporal features into account using memory cell units [20], [23]–[25]. Because the ciphertext sequence is decorrelated from the original plaintext, finding discriminative features between different categories requires a deep understanding of the context [26]. The bidirectional LSTM (BLSTM) provides a better understanding of context by learning in a bidirectional manner and learning the representations from future time steps [27], [28]. On the other hand, the GRU cell unit is computationally more efficient than conventional LSTM by deploying update and reset gates in hidden layers [29]. In our previous study, an attention-based LSTM was proposed to attack the classical cipher [30]. The attention mechanism improves the LSTM ability to save important information along the sequence.

Realizing the lack of deep learning applications on the ciphertext classification, this article presents a novel network model using BLSTM and GRU cell units to classify the ciphertext. The proposed network effectively captures time dependencies of the feature and the text features, which are essential for efficient cipher text classification. The present research evaluated three classical cipher methods (Caesar cipher, Vigenere cipher, and Substitution cipher). The efficiency of the proposed model was assessed using well-known evaluation metrics (accuracy, recall, precision, and F1 score) on two datasets: publicly available Brown corpus and company report datasets. The experimental results showed that the proposed model could effectively and quickly confirm the ciphertext category and achieve a high classification accuracy of 95.8%. In addition, we proposed a 1D CNN network model to evaluate the propsed network efficiency against the CNN model.

The main contributions according to the security and automated ciphertext classification challenges are as follows:

(1) A deep network model was designed based on BLSTM-GRU cell units for automated ciphertext classification, supporting different ciphertext lengths.

(2) In addition to features in the ciphertext document, this study focused on the temporal dependency from the input sequence using RNN-based cell units.

(3) The proposed method efficiency compared to various deep learning-based models was explored in the paper. The results showed that the proposed network model outperformed other deep learning-based approaches and was more efficient in automated ciphertext classification.

The remaining sections of this article are planned as follows. Section II briefly reviews RNN, LSTM, GRU, and classical cipher methods used in this experiment. Details of the proposed BLSTM-GRU and 1D CNN network models are presented in Section III. Section IV presents details of the experiments followed by results and discussions. Concluding remarks are given in the last section.

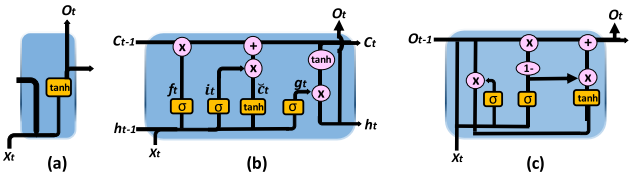

FIGURE 1. Recurrent neural networks and different cell units. (a) RNN model, (b) LSTM cell unit, (c) GRU cell unit.

II. METHODOLOGY

The current section in the article presents the details about the RNN, LSTM, and GRU cells units used to design the main constructing elements of the network model, followed by a review of different classical cipher techniques. This review is restricted to the above-mentioned state-of-the-art techniques because the primary focus was on ciphertext classification.

A. RECURRENT NEURAL NETWORKS (RNNs)

The RNN is a widely used specific neural network capable of sequence data processing that makes it suitable for learning algorithmic tasks. The RNN has been used for many natural language processing (NLP) applications. The main limitation of the RNN includes suffering from vanishing gradients in deep networks (see Fig. 1(a)) [31], [32]. For sequence data $(x_1, x_2, x_3, \ldots, x_t)$, the hidden state h_t of the RNN is calculated using the following equation:

$$h_t = f(h_{t-1}, x_t) \tag{1}$$

where f denotes the activation function.

B. LONG SHORT-TERM MEMORY (LSTM)

LSTM is a specific type of RNN model to solve the gradient vanishing problem of the RNN. LSTM is made up of three main gates. These three gates control the information flow in and out from LSTM structures to protect and control information: the forget gate, the input gate, and the output gate (see Fig. 1(b)) [31], [33]. The input gate stands for new information added to the cell state, the forget gate decides which information will be memorized or eliminated from the cell, and the output gate is for LSTM output. Sigmoid and tangent functions are mainly used in LSTM cells.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

$$i_t = \sigma(w_{xi}^T x_{(t)} + w_{hi}^T h_{(t-1)} + b_i) \tag{4}$$

$$f_t = \sigma(w_{xf}^T x_{(t)} + w_{hf}^T h_{(t-1)} + b_f) \tag{5}$$

$$o_t = \sigma(w_{xo}^T x_{(t)} + w_{ho}^T h_{(t-1)} + b_o) \tag{6}$$

$$g_t = \tanh(w_{xg}^T x_{(t)} + w_{hg}^T h_{(t-1)} + b_g) \tag{7}$$

$$c_t = f_t \otimes c_{(t-1)} + i_t \otimes g_t \quad (8)$$
$$h_t = o_t \otimes \tanh c_{(t)} \quad (9)$$

where σ is the sigmoid and *tanh* is the tangent activation function, respectively. i, f, o, c, and h are the input gate, forget gate, output gate, intermediate gate, and the cell memory output and $\otimes$ denotes element-wise multiplication; t represents time step and T represents the length of the window (the length of a sliding cutout of a time sequence of data); w denotes the layer weight representing input x, and b represents the threshold of the output gate.

C. GATED RECURRENT UNIT (GRU)

The GRU is a type of RNN structure with fewer gates compared to LSTM. In the GRU cell unit, the input and forget gates are controlled by one gate. Hence, the forget gate and input gate are combined into one gate, making the GRU simpler than LSTM [29]. For example, if $z_t = 1$, the entry of the new data for the input gate will be closed, and the forget gate is opened, whereas the mechanism acts vice versa when $z_t = 0$ (see Fig. 1(c)). The reset gate determines how to combine the new input with the previous memory to calculate the new state. The GRU differences from the LSTM are as follows:

$$r_t = \sigma(w_{xr}^T x_{(t)} + w_{hr}^T o_{(t-1)} + b_r) \quad (10)$$
$$z_t = \sigma(w_{xz}^T x_{(t)} + w_o^T z\, o_{(t-1)} + b_z) \quad (11)$$
$$o_t = z_t \otimes o_{t-1} + (1 - z_t) \otimes \tilde{o}_t \quad (12)$$

where r_t stands for reset gate; z_t represents the update gate; o_t is the output gate. $\otimes$ denotes element-wise multiplication; t represents the time step; T represents the length of the window; w denotes the layer weight representing input x, and b represents the threshold of the output gate.

D. CLASSICAL CIPHERS

The classical ciphers used in this experiment, including Caesar cipher, Vigenere cipher, and substitution cipher, are explained briefly below. To encrypt the original plaintext into unreadable ciphertext with a shift (or Caesar) cipher encryption method, each letter in the original message is replaced with a letter corresponding to a certain number of letters up or down in the alphabet. The number of possible shifts is limited to between 0 and 25 in the English language, which is equal to the number of English letters. The receiver decodes the ciphertext message by shifting each letter in the encrypted message back [34]. A Vigenere cipher is categorized as a poly-alphabetic cipher that encrypts a plaintext letter into a set of different letters using the Key with the total number of possible 26m keys. The substitution cipher deploys any permutation of the 26 letters as a key. Therefore, the total possible keys are $26! \approx 2^{88.4}$. Table 1 gives an example of the classical cipher methods.

E. WORD EMBEDDING

Word embedding is a set of language feature learning techniques in NLP converting word tokens to machine-readable vectors. *Word2vec* is a two-layer neural net that converts the text words into a vector. The input is a text corpus, and the output is a set of vectors. The advantage of *word2vec* is that it can train large-scale corpora to produce low-dimensional word vectors [35]. Given a sentence consisting of n words $(x_1, x_2, x_3, \ldots, x_{n-2}, x_{n-1}, x_n)$, every word x_i is converted into a real-valued vector, e_i, represented as

$$e_i = [w_1, w_2, w_3, \ldots., w_{n-2}, w_{n-1}, w_n] \in R^{n \times d} \quad (13)$$

where w is a word, and d is the size of the word embedding.

III. NETWORK ARCHITECTURE DESIGN

This section provides details of the proposed BLSTM-GRU network model and 1D CNN-based network model.

A. PROPOSED BLSTM-GRU NETWORK MODEL

First, the proposed network was tested using the LSTM network with three layers, and the results were evaluated. The parameter setting for each LSTM layer was selected experimentally. Subsequently, the LSTM layers were replaced with BLSTM and GRU cell units, and the network performance was evaluated. Table 2 lists the optimal hyperparameters setting of the proposed network model.

Fig. 2 shows the overall structure of the proposed network model. The input layer of the proposed network model is a sequence of the ciphertext. The input layer is a sequence input layer to enter sequential ciphertext data into the network, followed by a word-embedding layer. The next is the BLSTM layer, followed by a dropout layer to prevent network overfitting. The BLSTM layer learns the dependencies and dynamics between sequence data in a bidirectional manner, which is important for learning discriminative features of data in each time step. The dropout layer is normally used in a deep learning-based method to prevent the network from overfitting. The dropout layer randomly drops out a certain number of neurons to improve the generalizability of the network. This prevents network overfitting. The second layer of the proposed model is a GRU cell unit with 200 hidden cell units that can extract contextual features with a lower computational cost than LSTM, followed by a dropout layer. Afterward, a conventional LSTM unit is used with 200 hidden units followed by a fully connected with 60 neurons. The last layer is a fully-connected layer (FC) with the number of neurons equal to the number of classes for each dataset. A Softmax function is used to generate the probability of each ciphertext class. The proposed method can fully characterize each ciphertext information based on the advantages of the high precision sequence labeling ability of the network model. The softmax function can be calculated as follows:

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \quad (14)$$

TABLE 1. Example of plaintext, ciphertext, and corresponding category encrypted with different classical cipher methods.

Plaintext	Ciphertext	Category and Encryption Method
We today are not entitled to excoriate honest men who believed Parker to be downright pernicious and who barred their pulpits against his demand to poison the minds of their congregations	ZH WRGDB DUH QRW HQWLWOHG WR HAFRULDWH KRQHVW PHQ ZKR EHOLHYHG SDUNHU WR EH GRZQULJKW SHUQLFLRXV DQG ZKR EDUUHG WKHLU SXOSLWV DJDLQVW KLV GHPDQG WR SRLVRQ WKH PLQGV RI WKHLU FRQJUHJDWLRQV	Category: Religion Encryption method: Caesar cipher with 3 shifts to the right
	FG IUJDE DNG VUI GVIWIRGJ IU GQSUNWDIG ZUVGCI TGV FZU VGRWGLGJ YDNOGN IU VG JUFVNWHZI YGNVWSWUPC DVJ FZU VDNNGJ IZGWN YPRYWIC DHDWVCI ZWC JGTDVJ IU YUWCUV IZG TWVJC UM IZGWN SUVHNGHDIWUVC	Category: Religion Encryption method: Substitution cipher Key: "DVSJGMHZWXORTVUYANCIPLFQEK" (a→D, b→V, c→S, d→J…..)
	AL CHXEF JKY RVC XHXPCEYH AX XRGVABUXL QHHIZC FYR DQH VISRXPIK YTLOLA MI FL MHQRYRZBX WNKHMJRHOW HWW QLV KTLVLM MBIPA IOPWRMM ENJBHWA QBM HLVTHH AX IIMZXG NLL VBHHZ XY NLLRK WSUPKYKHCBIRZ	Category: Religion Encryption method: Vigenere cipher. Key: DGIST

TABLE 2. The BLSTM-GRU network model parameter setting.

Parameter	Value
Drop out	0.2
Mini-batch	128
Optimizer	Adam
Word embedding dimension	300
Learning rate	0.001

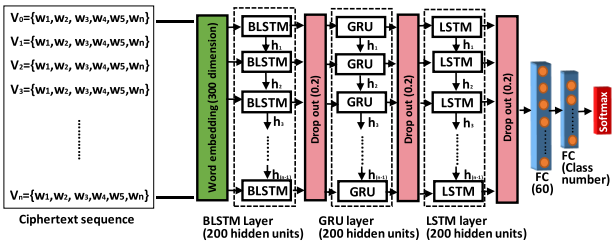

FIGURE 2. Proposed BLSTM - GRU network model for the ciphertext classification where the hidden blocks can be used as conventional LSTM or GRU cells.

where z stands for input vector and k is the number of classes; N denotes the total number of samples; i is the sample number.

The proposed model-training algorithm can be explained in the following steps.

B. PROPOSED 1D CNN-BASED MODEL

A 1D CNN-based model for ciphertext classification was proposed to evaluate the efficiency of the BLSTM-GRU model in comparison to CNN. CNN can capture hierarchical features of data via multiple consecutive convolution kernels [36]. CNN mainly consists of the following elements: 1) a set of convolutional filters, 2) an activation function, and 3) a max-pooling layer. A convolution layer is a fundamental

Algorithm 1 Proposed BLSTM-GRU Network Model

1. Input: the sequence of the ciphertext.
2. Output: corresponding class of the ciphertext.
3. Learning rate, batch size, embedding dimension.
4. BLSTM (hidden unit size, batch size)
5. Drop out Layer (0.2)
6. GRU (hidden unit size, batch size)
7. Drop out Layer (0.2)
8. LSTM (hidden unit size, batch size)
9. Drop out Layer (0.2)
10. FC (60 neurons)
11. FC (number of classes)
12. Softmax
13. Return Output

component of the CNN architecture that performs feature extraction, which is a combination of linear and nonlinear operations and activation functions as follows:

$$f = \sum_{i=1}^{N} (w_i x_i + b) \tag{15}$$

where i, w, b, x, and N are the input, layer weight, bias, input data, and the total number of samples, respectively.

Maximum pooling is referred to as a pooling operation that calculates the maximum values from each convolution filter. The results are downsampled or pooled feature maps that highlight the present feature in the patch calculates as follows:

$$\widehat{c} = \max(c) \tag{16}$$

where c stands for convolution layer values after the convolution operation.

The ReLU function is a nonlinear function applied to increase the nonlinearity of the CNN feature maps that can

TABLE 3. Parameters for tuning the proposed 1D CNN network model.

Pathway#	1	2	3
Kernel size	5	3	1
Dropout	0.2	0.2	0.2
Stride	2	2	2
Padding	0	0	0
Convolution filters	64/128	64/128	64/128

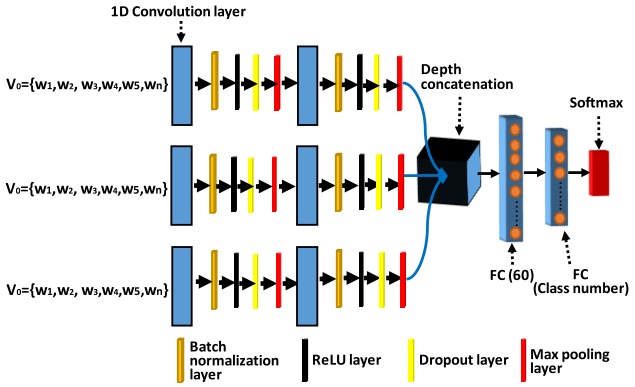

FIGURE 3. Proposed 1D CNN network model for the ciphertext classification.

be calculated as follows:

$$g(i) = \begin{cases} 0, & if\ i \leq 0 \\ i, & otherwise \end{cases} \tag{17}$$

Batch normalization is a method used to make artificial neural networks faster and more stable by calculating the mean and standard deviation of each input variable.

The proposed 1D CNN network consists of three parallel pathways that extract the features from the ciphertext. Each pathway consists of two 1D convolution layers, each followed by batch normalization, ReLU, and dropout layers (see Fig. 3). The extracted features are then fed to a max-pooling layer to reduce the data dimension. The number of convolutional filters in each of the pathways was 64 and 128, respectively. Consequently, the extracted features from each pathway are concatenated using a depth concatenation operation and fed into a fully connected layer. The Softmax function is used to generate the probability of each class. Table 3 lists the proposed 1D CNN parameters.

IV. RESULTS AND DISCUSSION

This section first introduces datasets used to evaluate the efficiency of the proposed network model for ciphertext classification. Second, the details of the experimental setup for network training, network training on different datasets, network performance analysis, the impact of hyperparameter tuning, and finally a discussion of the results are given.

A. DATASETS

Experiments are conducted on the two datasets to authenticate the effectiveness of the proposed network model. The datasets were divided randomly into training and test sets. The optimal amount of training and test datasets were obtained experimentally.

Dataset-1: The first dataset is the company reports dataset, containing documents related to different issues occurring during company operation. It consists of four hundred and eighty documents from four different classes, where each class represents a group of reports related to failure in different company sections.

Dataset-2: The second dataset is the Brown corpus, which consists of a collection of text samples from fifteen different classes. Both datasets are encrypted using three classical cipher methods (Caesar, Substitution, and Vigenere cipher). Table 4 lists the total amount of samples in each dataset and data distribution for training and test datasets. Fig. 4 shows the visualization of the class distribution of both datasets.

TABLE 4. Data distribution for the training and test datasets.

Name	Total data (sentence)	Training data	Test data	Categories
Brown corpus	57,340	80%	20%	15
Company report	480	80%	20%	4

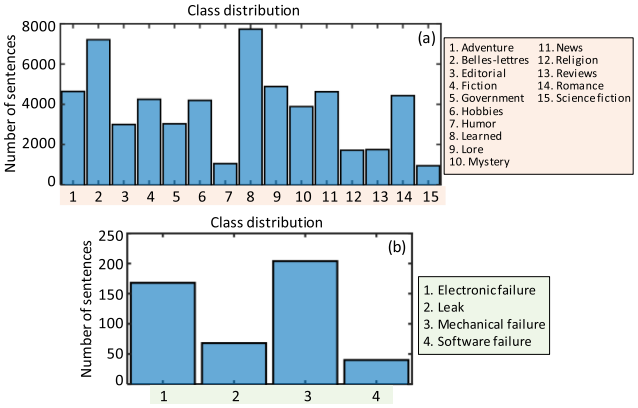

FIGURE 4. Visualization of the class distribution and the number of samples in the different datasets. (a) Brown corpus dataset, (b) company report dataset.

TABLE 5. System types of equipment.

Component	Description
CPU	Intel Cori7-7700
GPU	GTX1050
Language	Matlab
Memory	16 GB
System type	64-bit operating system
OS type	Window 10/64

B. EXPERIMENTAL SETUP

Table 5 lists the system configuration for training the proposed network models.

TABLE 6. Confusion matrix evaluation measures for the classification results.

		Predicted class	
		Positive	Negative
Actual class	Positive	True-positive (TP)	False-negative (FN)
	Negative	False-positive (FP)	True-negative (TN)

C. MODEL TRAINING

In this phase, the main work is the training of the proposed network models over the encrypted domain. The training procedure can speed up using Graphics Processing Unit (GPU). The proposed models were implemented using Matlab deep learning library, which can be executed on GPU. This will accelerate the training process 5 to 10 times. A stochastic gradient descent (SGD) training strategy subdivides the training dataset (called mini-batches) for each training epoch. A mini-batch size of 128 was considered for training the proposed method, which yielded better performance.

All optimal parameters were obtained experimentally. A dropout of 0.2 was used to prevent network overfitting. The Adam optimizer was used with a 0.001 learning rate and cross-entropy loss function [37]. The cross-entropy was defined by measuring the difference between the actual and predicted output of the model expressed as the following equation:

$$loss = -\sum_{i=1}^{N} (y_i \log \bar{y}_i) \tag{18}$$

where y stands for predicted probability by the network; $\bar{y}$ is the ground truth; i stands for the number of data; N is the total number of samples.

The training progress plot demonstrates the training accuracy per mini-batch. The training plots and corresponding cross-entropy loss for each mini-batch of both encrypted datasets against plaintext using the BLSTM-GRU network model are shown in Fig. 5(a) Brown dataset and (b) Company report dataset. The classifier accuracy using the proposed BLSTM-GRU network model oscillates between 92% to 100% for the Brown dataset and 95% to 100% for the Company report dataset. Similar results are found for plaintext classification accuracy. Figure 6 presents the training process of the proposed 1D CNN model on both datasets. The classifier accuracy oscillates between 72% to 81% for the Brown dataset and 77% to 85% for the Company report dataset.

D. PERFORMANCE ANALYSIS

The performance of the classification model was evaluated using a confusion matrix, which is a widely used method for measuring the classification accuracy of machine learning methods. The confusion matrix was calculated as listed in Table 6.

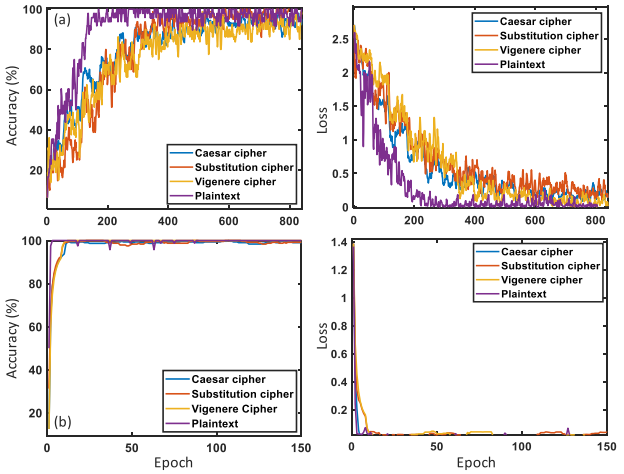

FIGURE 5. Accuracy and loss curves of the proposed BLSTM-GRU network model per epoch for each cipher method. (a) Brown corpus dataset, (b) company report dataset.

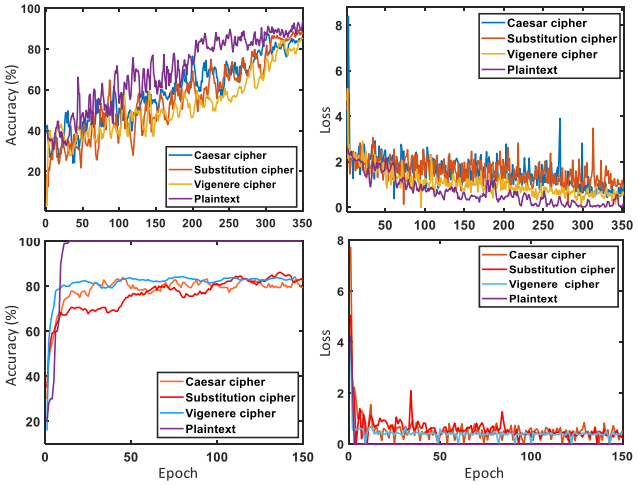

FIGURE 6. Accuracy and loss curves of the proposed 1D CNN network model per epoch for each cipher method. (a) Brown corpus dataset, (b) company report dataset.

True positive (TP) stands for true positive when network prediction is equal to the actual value, both one. True negative (TN) stands for true negative when the predicted value by the network and actual class are equal to 0. False positive (FP) represents the false positive when the actual class 0 and the predicted class is 1. False negative (FN) represents the false negative when the actual class is 1, and the predicted class is 0.

Fig. 7 presents the confusion matrix of the proposed BLSTM-GRU model performance for classifying the Vigenere encrypted text versus plaintext. The confusion matrix for Caesar and Substitution cipher encrypted text has a similar result to Fig. 7. The confusion matrix for simple plaintext without encryption shows higher accuracy than the encrypted text. Afterward, we evaluated the classification performance of the proposed BLSTM-GRU model using well-known metrics, such as the accuracy, precision, recall, and F1-Measure using equations (19), (20), (21), and (22),

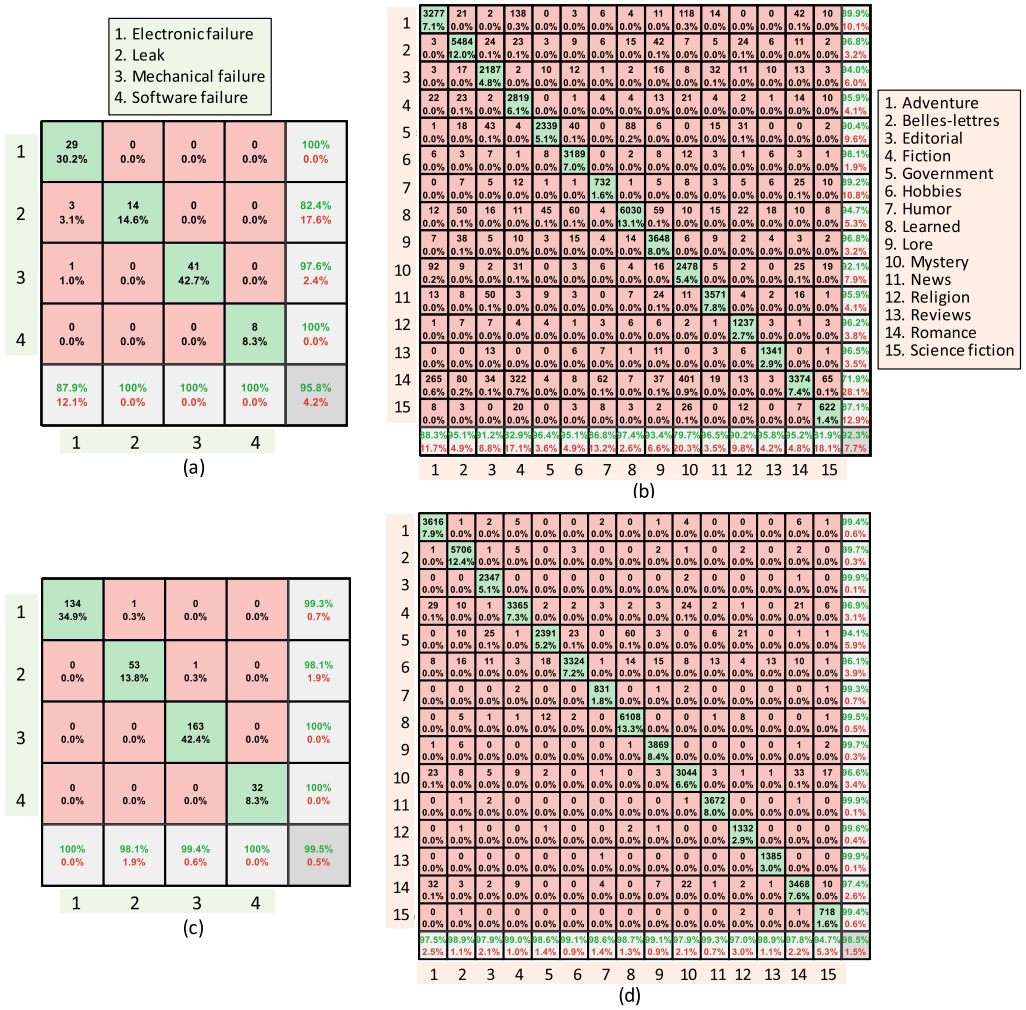

FIGURE 7. Confusion matrix of the ciphertext and corresponding plaintext classification and using proposed BLSTM-GRU network model. (a, b) Company report dataset and Brown corpus dataset encrypted using Vigenere cipher method, (c, d) confusion matrix for the corresponding plaintext classification.

respectively. The classification accuracy (CA) stands for the overall percentage of the correctly predicted cipher text classes on the dataset that can be calculated as follows:

$$CA = \frac{TP + TN}{TP + FP + FN + TN} \tag{19}$$

The precision stands for a fraction of relevant predictions among all the predicted values that can be calculated as follows:

$$precision = \frac{TP}{(TP + FP)} \tag{20}$$

The recall is the ratio of correctly predicted occurrences among all instances on the dataset, which can be calculated as follows:

$$recall = \frac{TP}{(TP + FN)} \tag{21}$$

The imbalanced dataset may harm the actual network accuracy due to accuracy detection towards the majority of classes. Accordingly, the F1 measure was utilized to assess the detection performance for the proposed model, which can be calculated as follows:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{22}$$

By taking advantage of the combination of BLSTM and GRU for ciphertext classification, a single layer of BLSTM was sufficient [38]. Figure 8 presents the result of different evaluation metrics using the CNN and BLSTM-GRU network models on both datasets. The proposed method showed high classification accuracy for up to 15 different categories. Based on the experiments, the BLSTM-GRU network model works better than the CNN for ciphertext classification.

The CNN performance for large datasets with long sequence lengths was much lower than the BLSTM-GRU network. This is because the Brown corpus dataset contains sentences with a long sequence length, and the CNN cannot process long sequence data, whereas the BLSTM-GRU network shows a high-level ability for long sequence processing.

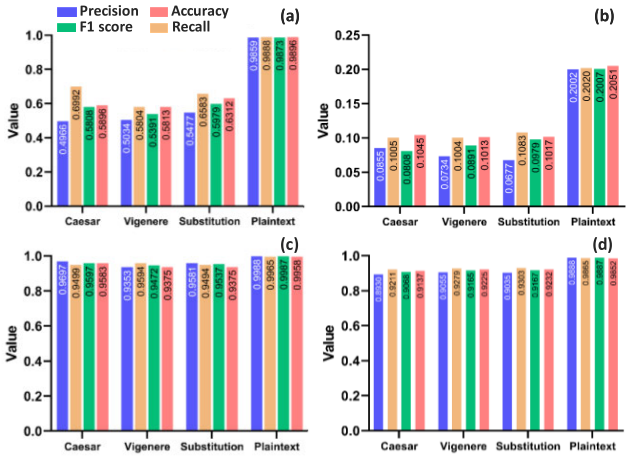

FIGURE 8. Performance comparison of CNN and BLSTM-GRU network models for ciphertext classification on different datasets. (a, b) CNN classification performance on the company report and Brown datasets, respectively. (c, d) BLSTM-GRU network model classification result on the company report dataset and Brown datasets, respectively.

TABLE 7. Impact of the word embedding dimension.

Accuracy	Embedding dimension	Epoch
~100	10	5
~100	100	4
~100	300	3

E. THE IMPACT OF HYPERPARAMETER TUNING

Hyperparameter tuning highly impacts the training processes that are assigned by users before training. The impact of hyperparameter tuning on network training was investigated using the company report dataset encrypted according to the Vigenere cipher method.

1) WORD EMBEDDING DIMENSION

Word embedding is a useful and popular tool in modern NLP, which is usually a linear or quadratic function of dimensionality. The word embedding dimension has a profound impact on the training time and computational costs. The smaller dimensionality of word embedding cannot capture all possible word relationships, whereas a very large embedding dimensionality leads to network overfitting and slows down training. Fig. 9(a) presents the experimental result for the impact of different word embedding dimensions on the training process, and Table 7 lists the quantitative results.

F. MINIBATCH SIZE

The mini-batch stochastic gradient descent (SGD) is a widely used technique for large-scale optimization problems for training machine learning models and deep learning models. The mini-batch refers to the amount of data used in every epoch to train the network. A, excessively large batch size slows the network convergence rate, while a too-small batch size makes the network fluctuate without achieving acceptable performance. Fig. 9(b) presents the experimental results for the impact of Mini-batch size on the training process, and Table 8 lists the quantitative results.

TABLE 8. Impact of the mini-batch size.

Accuracy	Mini-batch size	Epoch
~100	10	5
~100	100	3
~100	200	3

TABLE 9. Impact of hidden cell units.

Accuracy	Number of hidden units	Epoch
~100	10	6
~100	100	4
~100	200	4

G. IMPACT OF THE NUMBER OF LSTM HIDDEN UNITs

The number of hidden units in an LSTM refers to the dimensionality of the hidden states. Changing the number of hidden units affects the training of LSTMs. Fig. 9(c) shows the experimental result for the impact of hidden units on the training process, and Table 9 lists quantitative results.

H. DISCUSSION

This study introduced a deep learning-based network model for automated ciphertext classification with efficient performance using RNN-based cell units. Generally, the RNN-based model can store information along the sequence, which showed a better performance than other deep learning-based models by taking the temporal and spatial features into account [31]–[33]. Bidirectional LSTM enables a better understanding of context by learning future time steps in a bidirectional manner. Moreover, GRU deploys reset and update gates in the hidden layer, which is computationally more efficient than a conventional LSTM. In this paper, a network model based on the BLSTM-GRU cell units was proposed to recognize the ciphertext category automatically and accurately. In addition to features in the ciphertext document, this study focused on the temporal dependency from the input sequence using RNN-based cell units. Furthermore, to evaluate the efficiency of proposed BLSTM-GRU network model against CNN model, we proposed a 1D CNN-based network model for ciphertext classification. The proposed BLSTM-GRU method efficiency was compared with several other deep learning-based models, including proposed 1D CNN model. The results suggest the efficacy of the proposed BLSTM-GRU network model using different well-known evaluation metrics, including the F1 score, precision, and recall. The RNN-based model disadvantages can be expressed as long-term dependence problems, gradient fading, or gradient explosion problems. In this experiment,

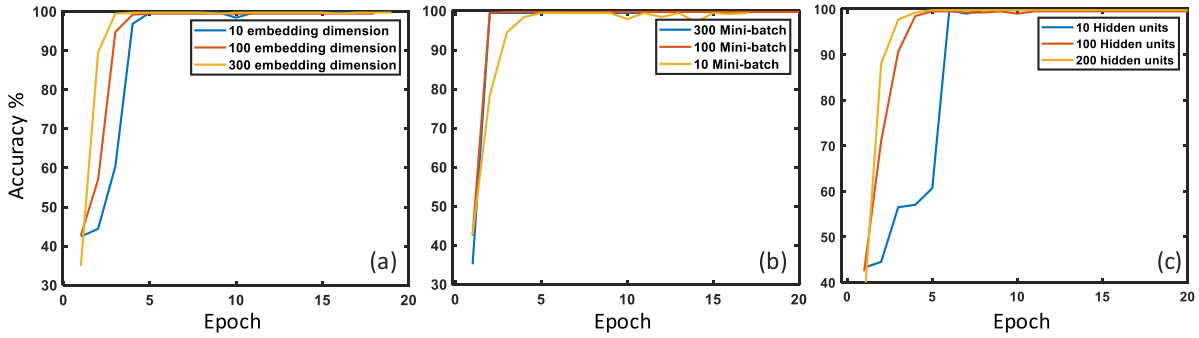

FIGURE 9. Impact of hyperparameter tuning on the network training performance. (a) Impact of the word-embedding dimension, (b) impact of the mini-batch size, (c) impact of the hidden cell units.

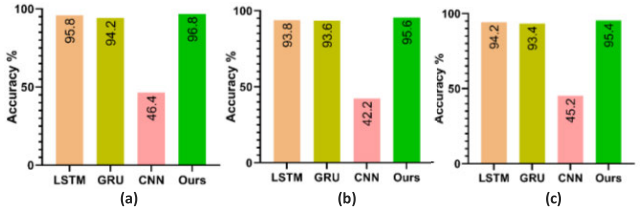

FIGURE 10. Performance comparison of different deep learning-based models for ciphertext classification. (a) Caesar cipher, (b) Substitution cipher, (c) Vigenere cipher.

the Adam optimization method was used to train the model, and the learning rate was set to 0.001. The dropout method is used to prevent overfitting with a factor of 0.2. In addition, this study investigated the impact of different hyperparameter tunings on network performance, including word embedding dimension, minibatch size, and the number of BLSTM-GRU cell units. The experimental results indicated that the network could converge faster using optimal hyperparameters. The effectiveness and performance of the proposed method were assessed by comparing the proposed method with some of the other deep learning-based models shown in Fig. 10.

V. CONCLUSION

This paper proposed a network model based on BLSTM and GRU network model, which has recently outperformed many deep learning approaches in the sequential data processing. The BLSTM showed a better understanding of the context by learning the future time steps in a bidirectional manner. The GRU cell unit deploys an update and reset gate, which is more efficient than the conventional LSTM model. Based on the experimental results, the method yielded high classification accuracy by deploying the bidirectional learning and which enabled the extraction of more distinctive features to predict the ciphertext classes better. The proposed method can classify ciphertext in modern ciphers, which are more complex, and the relationships and dependencies are more complex to discover distinctive features for accurate cipher text classification. The limitation of the proposed model can be expressed as long-term dependence problems in long ciphertext sequences, which causes the LSTM to lose important information along the sequence. Thus, future work will include an investigation of the network's ability to classify ciphertext encrypted with modern ciphers for long sequence.

REFERENCES

[1] S. Swapna, A. D. Dileep, C. C. Sekhar, and S. Kant, "Block cipher identification using support vector classification and regression," *J. Discrete Math. Sci. Cryptogr.*, vol. 13, no. 4, pp. 305–318, Aug. 2010.

[2] O. L. Usman and R. C. Muniyandi, "CryptoDL: Predicting dyslexia biomarkers from encrypted neuroimaging dataset using energy-efficient residue number system and deep convolutional neural network," *Symmetry*, vol. 12, no. 5, p. 836, May 2020.

[3] A. N. Gomez, S. Huang, I. Zhang, B. M. Li, M. Osama, and L. Kaiser, "Unsupervised cipher cracking using discrete GANs," 2018, *arXiv:1801.04883*.

[4] J. Chen and J. S. Rosenthal, "Decrypting classical cipher text using Markov chain Monte Carlo," *Statist. Comput.*, vol. 22, no. 2, pp. 397–413, Mar. 2012.

[5] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 720–729, Jan. 2018.

[6] P. Khadivi and M. Momtazpour, "Ciphertext classification with data mining," in *Proc. IEEE 4th Int. Symp. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2010, pp. 64–66.

[7] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, Nov. 2015.

[8] S. Mishra and A. Bhattacharjya, "Pattern analysis of cipher text: A combined approach," in *Proc. Int. Conf. Recent Trends Inf. Technol. (ICRTIT)*, Jul. 2013, pp. 393–398.

[9] A. N. Khan, M. Y. Fan, A. Malik, and M. A. Husain, "Cryptanalyzing Merkle–Hellman public key cryptosystem with artificial neural networks," in *Proc. IEEE 5th Int. Conf. Converg. Technol. (I2CT)*, Mar. 2019, pp. 1–7, doi: 10.1109/I2CT45611.2019.9033917.

[10] R. Focardi and F. L. Luccio, "Neural cryptanalysis of classical ciphers," in *Proc. 19th Italian Conf. Theor. Comput. Sci. (ICTCS)*, 2018, pp. 104–115.

[11] S. N. B. Bhushan and A. Danti, "Classification of compressed and uncompressed text documents," *Future Gener. Comput. Syst.*, vol. 88, pp. 614–623, Nov. 2018.

[12] H.-S. Shin, H.-Y. Kwon, and S.-J. Ryu, "A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in Twitter," *Electronics*, vol. 9, no. 9, p. 1527, Sep. 2020.

[13] A. J. Abd and S. T. F. Al-Janabi, "Classification and identification of classical cipher type using artificial neural networks," *J. Eng. Appl. Sci.*, vol. 14, no. 11, pp. 3549–3556, Nov. 2019.

[14] J. Luthra and S. K. Pal, "A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher," in *Proc. World Congr. Inf. Commun. Technol. (WICT)*, Dec. 2011, pp. 202–206.

[15] L. Liu, D. Jiang, T. An, and Y. Guan, "A plaintext-related dynamical image encryption algorithm based on permutation-combination-diffusion architecture," *IEEE Access*, vol. 8, pp. 62785–62799, 2020.
[16] S. A. Mokhov, "Cryptolysis: A framework for verification of optimization heuristics for the automated cryptanalysis of classical ciphers and natural language word segmentation," in *Proc. 8th ACIS Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, 2010, pp. 295–302.
[17] I. Polak and M. Boryczka, "Tabu search in revealing the internal state of RC4+ cipher," *Appl. Soft Comput.*, vol. 77, pp. 509–519, Apr. 2019.
[18] E. Ahmadzadeh, K. Jaferzadeh, S. Shin, and I. Moon, "Automated single cardiomyocyte characterization by nucleus extraction from dynamic holographic images using a fully convolutional neural network," *Biomed. Opt. Exp.*, vol. 11, no. 3, pp. 1501–1516, Mar. 2020.
[19] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes, "Big data analysis for brain tumor detection: Deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 87, pp. 290–297, Oct. 2018.
[20] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019.
[21] M. Usama, B. Ahmad, E. Song, M. S. Hossain, M. Alrashoud, and G. Muhammad, "Attention-based sentiment analysis using convolutional and recurrent neural network," *Future Gener. Comput. Syst.*, vol. 113, pp. 571–578, Dec. 2020.
[22] K. Kumari, J. P. Singh, Y. K. Dwivedi, and N. P. Rana, "Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization," *Future Gener. Comput. Syst.*, vol. 118, pp. 187–197, May 2021.
[23] F. Rundo, "Deep LSTM with dynamic time warping processing framework: A novel advanced algorithm with biosensor system for an efficient car-driver recognition," *Electronics*, vol. 9, no. 4, p. 616, Apr. 2020.
[24] J. Vicente-Gabriel, A.-B. Gil-González, A. Luis-Reboredo, P. Chamoso, and J. M. Corchado, "LSTM networks for overcoming the challenges associated with photovoltaic module maintenance in smart cities," *Electronics*, vol. 10, no. 1, p. 78, Jan. 2021.
[25] I. E. Livieris, N. Kiriakidou, S. Stavroyiannis, and P. Pintelas, "An advanced CNN-LSTM model for cryptocurrency forecasting," *Electronics*, vol. 10, no. 3, p. 287, Jan. 2021.
[26] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 794–802.
[27] H. M. Lynn, S. B. Pan, and P. Kim, "A deep bidirectional GRU network model for biometric electrocardiogram classification based on recurrent neural networks," *IEEE Access*, vol. 7, pp. 145395–145405, 2019.
[28] X. Luo and Z. Chen, "English text quality analysis based on recurrent neural network and semantic segmentation," *Future Gener. Comput. Syst.*, vol. 112, pp. 507–511, Nov. 2020.
[29] Y. Hao, Y. Sheng, and J. Wang, "Variant gated recurrent units with encoders to preprocess packets for payload-aware intrusion detection," *IEEE Access*, vol. 7, pp. 49985–49998, 2019.
[30] E. Ahmadzadeh, H. Kim, O. Jeong, and I. Moon, "A novel dynamic attack on classical ciphers using an attention-based LSTM encoder-decoder model," *IEEE Access*, vol. 9, pp. 60960–60970, 2021.
[31] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based LSTM with position context for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 20462–20471, 2019.
[32] L. Bai, L. Yao, X. Wang, C. Li, and X. Zhang, "Deep spatial–temporal sequence modeling for multi-step passenger demand prediction," *Future Gener. Comput. Syst.*, vol. 121, pp. 25–34, Aug. 2021.
[33] X. Chen, J. He, X. Wu, W. Yan, and W. Wei, "Sleep staging by bidirectional long short-term memory convolution neural network," *Future Gener. Comput. Syst.*, vol. 109, pp. 188–196, Aug. 2020.
[34] M. Russell, J. A. Clark, and S. Stepney, "Using ants to attack a classical cipher," in *Genetic and Evolutionary Computation* (Lecture Notes in Computer Science), vol. 2723. Berlin, Germany: Springer, 2003, pp. 146–147.
[35] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios, and R. Valencia-García, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," *Future Gener. Comput. Syst.*, vol. 114, pp. 506–518, Jan. 2021.
[36] S. Kumar, "MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things," *Future Gener. Comput. Syst.*, vol. 125, pp. 334–351, Dec. 2021.
[37] X. Li, D. Chang, T. Tian, and J. Cao, "Large-margin regularized softmax cross-entropy loss," *IEEE Access*, vol. 7, pp. 19572–19578, 2019.
[38] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.

EZAT AHMADZADEH received the M.S. degree in software engineering from the Science and Research Branch of Tehran, IAU University, Iran, in 2014, and the Ph.D. degree in computer engineering from Chosun University, South Korea, in 2020. He is currently a Postdoctoral Researcher with DGIST, Daegu, South Korea. His research interests include medical image analysis, artificial intelligence, AI-based cryptanalysis, deep learning, and machine learning.

HYUNIL KIM received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in information security from Kongju National University, South Korea, in 2014, 2016, and 2019, respectively. He is currently a Postdoctoral Researcher with DGIST, Daegu, South Korea. His research interests include AI-based cryptanalysis, artificial intelligence, deep learning, blockchain, and federated learning.

ONGEE JEONG received the B.S. degree in biomedical engineering from KonKuk University, South Korea, in 2019. She is currently pursuing the integrated M.S. and Ph.D. degrees with the Department of Robotics Engineering, DGIST. Her research interests include deep learning, information security, image processing, and digital holography.

NAMKI KIM received the B.S. degree in mechanical engineering from the Kumoh National Institute of Technology, South Korea, in 2021. He is currently pursuing the M.S. degree with the Department of Robotics Engineering, DGIST. His research interests include information security and deep learning.

INKYU MOON (Member, IEEE) received the B.S. degree in electronics engineering from Sungkyunkwan University, South Korea, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2007. From 2009 to 2017, he was a Faculty Member with the Department of Computer Engineering, Chosun University, South Korea. He joined DGIST, South Korea, in 2017, and is currently a Professor with the Department of Robotics Engineering. He has more than 100 publications, including peer-reviewed journal articles, conference proceedings, and invited conference papers. His research interests include information security and deep learning. He is a Senior Member of OSA.