Indonesian Journal of Electrical Engineering and Computer Science
Vol. 35, No. 2, August 2024, pp. 971~978
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v35.i2.pp971-978

Hybrid encryption based on a generative adversarial network

Iqbal Amir[1], Hamizan Suhaimi[1], Roslina Mohamad[1], Ezmin Abdullah[1], Chuan-Hsian Pu[2]
[1]Wireless High Speed Network Research Group (WHiSNet), School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia
[2]Department of Electrical and Electronic Engineering, University of Nottingham Malaysia, Semenyih, Malaysia

Article Info

Article history:

Received Mar 10, 2024
Revised Apr 2, 2024
Accepted Apr 6, 2024

Keywords:

Adversarial attacks
Generative adversarial
Hybrid encryption
Message encryption
Neural network

ABSTRACT

In today's world, encryption is crucial for protecting sensitive data. Neural networks can provide security against adversarial attacks, but meticulous training and vulnerability analysis are required to ensure their effectiveness. Hence, this research explores hybrid encryption based on a generative adversarial network (GAN) for improved message encryption. A neural network was trained using the GAN method to defend against adversarial attacks. Various GAN training parameters were tested to identify the best model system, and various models were evaluated concerning their accuracy against different configurations. Neural network models were developed for Alice, Bob, and Eve using random datasets and encryption. The models were trained adversarially using the GAN to find optimal parameters, and their performance was analyzed by studying Bob's and Eve's accuracy and bits error. The parameters of 8,000 epochs, a batch size of 4,096, and a learning rate of 0.0008 resulted in 100% accuracy for Bob and 52.14% accuracy for Eve. This implies that Alice and Bob's neural network effectively secured the messages from Eve's neural network. The findings highlight the advantages of employing neural network-based encryption methods, providing valuable insights for advancing the field of secure communication and data protection.

Corresponding Author:

Roslina Mohamad
Wireless High Speed Network Research Group (WHiSNet), School of Electrical Engineering
College of Engineering, Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
Email: roslina780@uitm.edu.my

1. INTRODUCTION

Cryptography has been a significant tool for information security for millennia. By limiting unwanted access, cryptography guarantees the confidentiality, integrity, and validity of sensitive data [1]. One of the trickiest things about cryptography is keeping the encryption keys safe. Secret encryption keys are required for encrypting and decrypting messages [2]. The security of a message may be compromised if an attacker were to obtain the key, deduce the encryption algorithm, and then read or alter the message. As a result, cryptographers have been trying to develop more robust and secure encryption methods that are resilient to attacks from hackers and other criminals [3].

An encryption algorithm can also be made more secure by utilizing a neural network to strengthen the key generated by the encryption process [4]. The network comprises interconnected nodes or "neurons" arranged in layers designed for processing and analyzing complex data [5]. It uses a process called backpropagation to modify internal parameters and achieve the desired outcome. During this process, the network compares its expected output to the actual output and adjusts its parameters as needed. Training a neural network involves feeding it a collection of input data and allowing it to adjust its internal parameters to

produce the desired output [6]. Once trained, a neural network can accurately classify data or make predictions even if the input is noisy or incomplete [7].

A generative adversarial network (GAN) is a machine learning model consisting of a generator and a discriminator neural network [8]. The generator creates new examples, while the discriminator evaluates them. The goal of the generator is to produce data that mimics real data, and the goal of the discriminator is to differentiate between real and generated data. Together, these two networks are trained in a zero-sum game in which one agent's success equals the other's failure. The generator tries to fool the discriminator, and the discriminator tries to correctly classify the data as real or generated [9]. GANs are used for unsupervised learning, a type of machine learning by which a model learns to identify patterns in a dataset without labeled responses [10].

Recent developments in the field of encryption have seen the application of neural networks to improve the security and efficiency of encryption systems [11]–[13]. One of the most exciting applications is the generation of encryption keys [14]. Neural networks can be trained to generate secure and unpredictable encryption keys that are less likely to be guessed by attackers [15]. In addition, neural networks can help identify patterns in encrypted data that attackers can exploit and detect anomalies that may indicate a security breach. Neural networks are also being used to enhance the security of encryption techniques for secure communication by encrypting models and their parameters. Techniques like adversarial training encryption can process encrypted data without revealing sensitive data or the model itself [16].

Many research papers on cryptography use neural networks instead of fixed encryption algorithms [16]–[18] to enable self-learning and increase the flexibility of encryption methods. However, this also poses challenges for clarity and explanation. One notable exception is the research in [17] that proposed a hybrid encryption model, combining the strength of the Advanced Encryption Standard with an adversarial network. This achieves a balance between traditional encryption methods and innovation.

The researchers in [19] introduced a method for protecting multi-party communication using adversarial training with neural networks. The GAN model was trained for 200 epochs using a batch size of 256 and a learning rate of 0.0008 with the Adam optimizer. However, training many neural networks is computationally expensive, and the best parameter values for this method are still under study. A GAN was used for secure communication [16]. Three neural networks—Alice, Bob, and Eve—competed in an adversarial training in which Alice encrypted messages to Bob while Eve tried to decode them. The network trained for 15,000 epochs with a batch size of 128 and a learning rate of 0.0002. The researchers acknowledged the high computational cost of training large neural networks for real-time communication.

Encryption protects sensitive data from unauthorized access and manipulation. However, adversarial attacks (i.e., malicious inputs that can exploit vulnerabilities in an encryption model) can compromise encryption [20]. These attacks can pose a severe risk for encryption, as attackers may be able to decipher or alter encrypted messages without detection. Neural network capabilities in symmetric encryption can be utilized to avoid such risk. However, neural network models are also sensitive to the choice of hyperparameters during training, such as the learning rate, number of hidden layers, and activation functions [21]. These hyperparameters affect the performance and generalization ability of models, and finding the optimal values is challenging. If the hyperparameters are not tuned properly, a model may suffer from overfitting, underfitting, slow convergence, or instability [22]. If a neural network encryption method is to perform optimally, it must be tested appropriately for potential vulnerabilities. To address this challenge, researchers and developers must analyze the encryption to prevent attackers from stealing data [23]. By doing so, a robust encryption model that ensures the confidentiality and integrity of encrypted information can be developed, thus safeguarding the information against unauthorized access.

The remainder of this paper is organized as follows: section 2 discusses the method used in this project, which involved developing a GAN model, assessing the performance of different parameters, and analyzing the system's effectiveness in securing a message. Section 3 presents the results and discussion. Finally, Section 4 provides the conclusions and recommendations for future works.

2. HYBRID ENCRYPTION METHOD

This research project phase describes the construction of a hybrid encryption technique that combines exclusive or (XOR) encryption with a GAN model. The first step involves creating a dataset of messages and keys for encryption and developing a model structure. A training process is then created that utilizes GAN training.

2.1. Generating messages and keys

The dataset generation process consists of creating a batch of random binary values, each with 16 bits, for both messages and keys. Each binary value is converted into -1 or 1 using a representation common

in neural network applications. This conversion process is essential for the neural network's operation. The messages and keys are the input data for training the neural network. This binary representation enables the network to learn and process the XOR operation's encryption and decryption patterns, enhancing the symmetric cryptosystem's overall training.

2.2. Developing XOR encryption and the GAN model

The purpose of this research was to combine XOR encryption and a GAN model. The model consists of three neural network models: Alice, Bob, and Eve. The diagram of the XOR encryption and GAN model integration is shown in Figure 1, where P is the message, K is the key, C is the ciphertext, C_{XOR} is the ciphertext after XOR encryption, P_{BOB} is the message created by Bob, and P_{EVE} is the message created by Eve. Messages and keys were encrypted with XOR to produce XOR ciphertext. Alice uses XOR ciphertext and a key to generate the ciphertext. Bob and Eve receive Alice's ciphertext, but only Bob has the key to recover the original message, while Eve tries to recover the message without the key. A GAN-based training session is performed to improve the model by minimizing the loss for better message reconstruction. Eve's loss is calculated by comparing the original and reconstructed messages. Bob's loss is calculated by comparing the original and reconstructed messages and taking Eve's loss as input to ensure that Eve's error is at least 50%.

Alice, Bob, and Eve consist of the convolutional neural network model shown in Figure 2. It comprises 32 input units, three hidden layers of two, four, and four channels with sigmoid activation, and one output layer of one channel with tanh activation. The output is the result of the convolutional neural network model. This model has three copies for Alice, Bob, and Eve. Alice uses the XOR output and keys to create ciphertext, Bob uses Alice's ciphertext and keys to reconstruct the message, and Eve uses only Alice's ciphertext to guess the message.

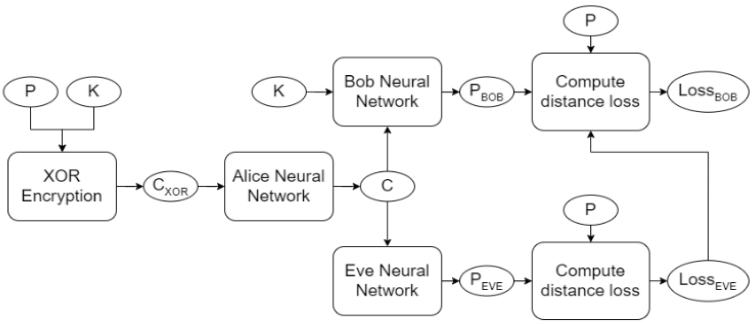


Figure 1. The diagram of the integration of XOR encryption with a GAN model [17]

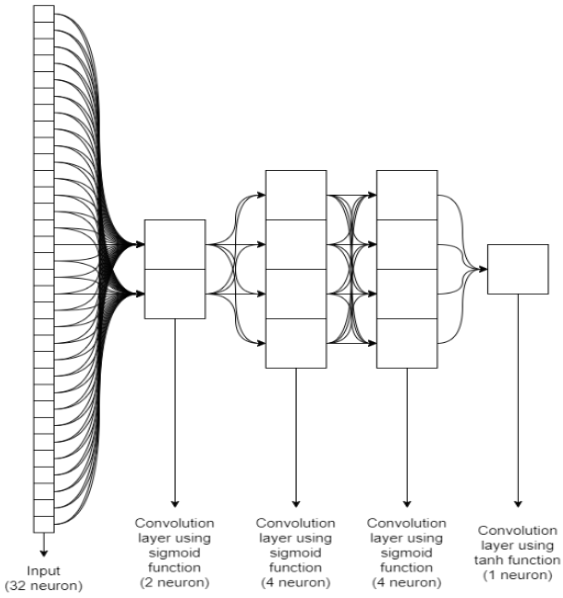


Figure 2. Diagram of the convolution neural network [16]

2.3. Training process using adversarial training

The training process optimizes the parameters of the neural network models through iterations. The optimization process uses the Adam optimizer separately for Eve and the combined Alice-Bob models [15]. The loss functions used during training include Eve's loss, which compares Eve's predictions and the original messages, and Alice-Bob's loss, which includes the reconstruction loss for Bob. The formula for the loss is shown at (1), where n is the number of batch sizes, X is the original message, and Y is the reconstructed message.

$$L1\ distance\ loss = \left(\frac{1}{n}\right) \times \sum |X - Y| \quad (1)$$

The absolute difference between the original message and the reconstructed message is taken for each element, and the mean of these differences is computed by summing and dividing by the number of batch sizes. It was ensured that Eve's output has at least 50% wrong bits by applying the formula shown in (2) with the loss computation in Alice-Bob loss, where N is the message bit length and Eve loss is the loss computed from Eve's *L1* destination loss.

$$Eve\ 50\%\ loss = \frac{\left(\frac{N}{2} - Eve\ loss\right)^2}{\left(\frac{N}{2}\right)^2} \quad (2)$$

In (2), N represents the total number of bits in the message. By subtracting Eve's loss from $N/2$ and squaring the result, the formula quantifies the deviation of Eve's output from the ideal scenario of 50% incorrect bits. Dividing this squared deviation by $(N/2)^2$ scales the loss value appropriately. The training modifies the model parameters to enhance the network's ability to communicate securely between Alice and Bob while making it difficult for Eve to eavesdrop. The parameter value for the model is examined in the project's second phase.

2.3.1. Configuring the parameter for model training

The three value adjustment training parameters in Table 1 are applied. For the first set, three epochs are selected (1,000, 4,000, and 8,000). These values are selected to capture a range of training scenarios, from relatively short durations to longer sessions. For the second set, the project uses different batch sizes, representing the number of samples processed in each training iteration, which is changed across three levels: 256, 1024, and 4096. Batch sizes in multiples of 128 or 256 optimize the Tensor Core design's performance [24]. The choice of these batch sizes is motivated by the need to explore the model's behavior under different computational loads. Lastly, the third set uses different learning rates. A common suggestion is to consider a learning rate of less than 1.0 and greater than 0.00001 [25]. In this project, the learning rate is tested at three values (0.0001, 0.0008, and 0.0016) to evaluate the model's sensitivity to the learning rate. Lower values suggest more accurate convergence but at the risk of slower learning; higher values indicate a risk of overshooting the optimal parameters.

Table 1. Values of parameters used for training

Parameter name	First set value adjustment	Second set value adjustment	Third set value adjustment
Epoch	1,000, 4,000, 8,000	8,000	8,000
Batch size	4096	256, 1,024, 4,096	4,096
Learning rate	0.0008	0.0008	0.0001, 0.0008, 0.0016

3. RESULTS AND DISCUSSION

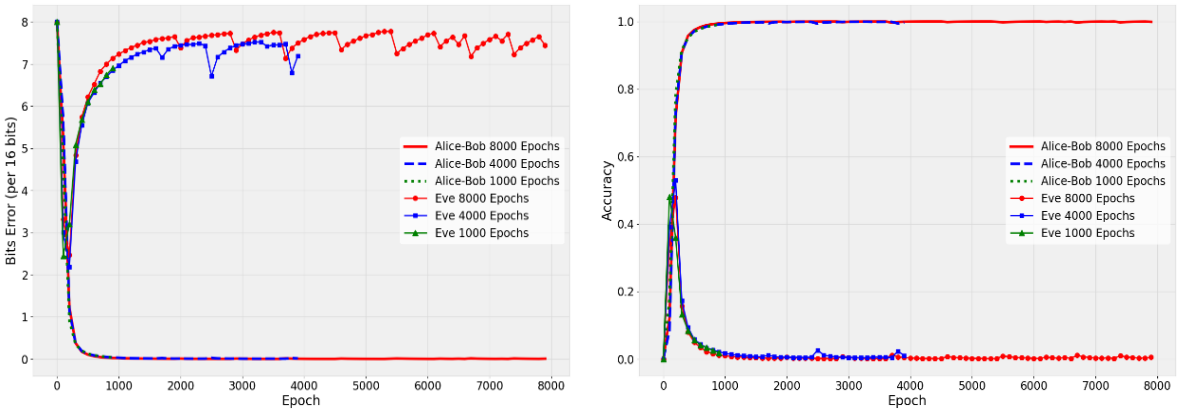
Figure 3 shows the results of the bits error and accuracy of Alice-Bob and Eve for 1,000, 4,000, and 8,000 epochs. As depicted in Figure 3(a), the Alice-Bob bits error consistently dropped to 0.3 on epoch 300 in all scenarios and then gradually approached 0 and was maintained until the final epoch, indicating successful learning and improved communication. At the same time, Eve's loss rapidly increased until 900 epochs, showing the system's resilience against decryption attempts by Eve. For the final bits error, the system achieved 7.446 for 8,000 epochs, 7.1965 for 4,000 epochs, and 6.9035 for 1,000 epochs.

Meanwhile, Figure 3(b) shows Alice-Bob's and Eve's accuracy for 1,000, 4,000, and 8,000 epochs. The message-decoding accuracy of Alice and Bob for all scenarios showed a consistent rise that reached approximately 1, indicating the model's skill in learning the encryption-decryption process until 700 epochs and maintaining their performance until the final epochs. Furthermore, Eve's accuracy for all scenarios

rapidly increased to 0.45 at 300 epochs and subsequently decreased, reaching almost 0 at 700 epochs, which was then maintained until the final epochs. These outcomes highlight the encryption scheme's effectiveness, especially with extended training. Across all scenarios, no significant difference was found in any of the graphs other than that representing the number of epochs.

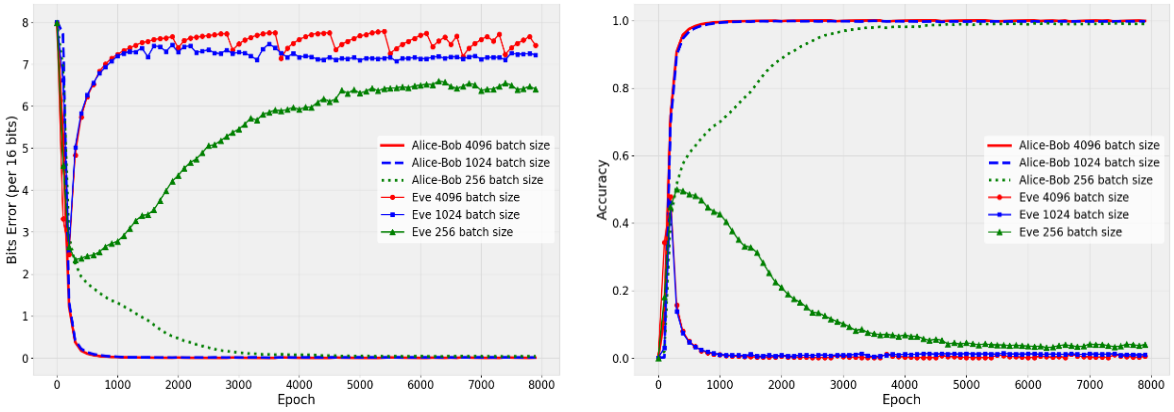
Figure 4 shows the results of the bits error and accuracy of Alice-Bob and Eve for batch sizes of 256, 1,024, and 4,096. In the first scenario (Figure 4(a)), with a batch size of 256, Alice-Bob's value quickly dropped to 1.9345 at 400 epochs and then slowly reached 0.1256 at 3,000 epochs, where it stayed. Meanwhile, Eve's value quickly fell to 2.3444 at 300 epochs and then gradually rose to 6.4084 at the final epoch of 8,000. In the second scenario, with a batch size of 1,024, Alice-Bob's value plummeted to 0.1345 at 500 epochs and remained constant. Eve's value also dropped to 2.5746 at 200 epochs and increased to 7.1922 at 1000 epochs, where it stayed. In the third scenario, with a batch size of 4,096, Alice-Bob's value sharply declined to 0.1724 at 400 epochs and did not change thereafter. Eve's value also sharply dropped to 2.4703 at 200 epochs, sharply rose to 7.6532 at 1,900 epochs, and then oscillated between 7.2 and 7.8.

Figure 4(b) shows the Alice-Bob and Eve accuracies for batch sizes of 256, 1,024, and 4,096. In the first scenario, with a batch size of 256, Alice-Bob's accuracy gradually rose to 0.9885 at 5000 epochs, where it stayed. Eve's accuracy quickly climbed to 0.4997 at 300 epochs and then slowly dropped to 0.0395 by the final epoch of 8,000. The second and third scenarios, with batch sizes of 1,024 and 4,096, indicate similar performances. Alice-Bob's accuracy swiftly reached 1 at 1,200 epochs and then remained constant. Eve's accuracy ascended to 0.45 at 200 epochs and then swiftly descended to 0.008 at 1,100 epochs, where it stayed. In summary, the batch size of 4,096 was the most effective for both Alice-Bob and Eve regarding bit error and accuracy.



(a) (b)

Figure 3. The results of Alice-Bob and Eve for 1,000, 4,000, and 8,000 epochs: (a) bits error and (b) accuracy

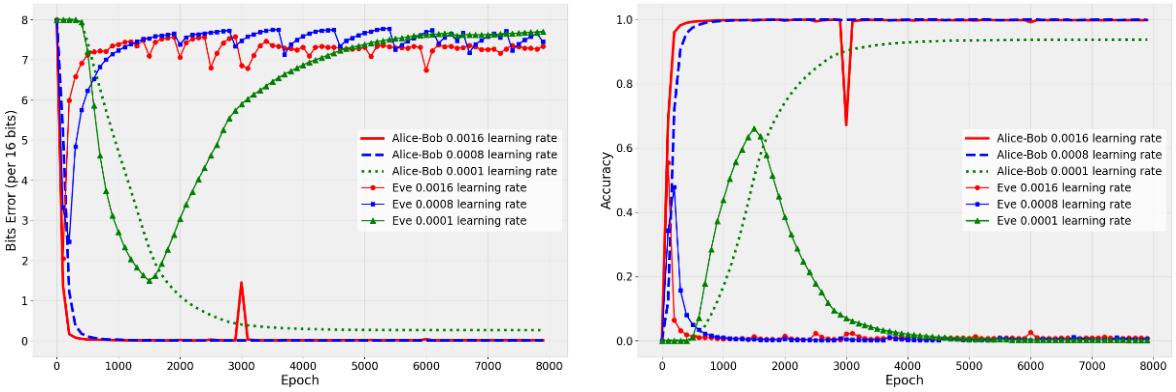


(a) (b)

Figure 4. The results of Alice-Bob and Eve for batch sizes of 256, 1,024, and 4,096: (a) bits error and (b) accuracy

Figure 5 shows the results of the bits error and accuracy of Alice-Bob and Eve for learning rates of 0.0001, 0.0008, and 0.0016. Figure 5(a) shows that for the first scenario, with a learning rate of 0.0001, Alice-Bob's bits error slowly dropped to 0.2574 at 8,000 epochs. Eve's bits error quickly fell to 1.4933 at 1,500 epochs and then slowly rose to 7.700 at 8,000 epochs. In the second scenario, with a learning rate of 0.0008, Alice-Bob's bits error sharply declined to 0.0669 at 600 epochs, where it stayed. Eve's bits error sharply decreased to 2.4703 at 200 epochs, sharply increased to 7.6532 at 1,900 epochs, and then varied between 7.2 and 7.8. In the third scenario, with a learning rate of 0.0016, Alice-Bob's bits error swiftly dropped to 0.0733 at 300 epochs and remained constant, except for a spike of 1.446 at 3,000 epochs. Eve's bits error swiftly decreased to 2.0473 at 100 epochs, rapidly increased to 7.1239 at 500 epochs, and then fluctuated between 7 and 7.6.

Figure 5(b) shows the accuracy of Alice-Bob and Eve for learning rates of 0.0001, 0.0008, and 0.0016. In the first scenario, with a learning rate of 0.0001, Alice-Bob's accuracy slowly rose to 0.9021 at 3,000 epochs and stayed at that level. Eve's accuracy quickly climbed to 0.6615 at 1,500 epochs and then dropped to 0.0107 at 4,500 epochs, where it remained. In the second scenario, with a learning rate of 0.0008, Alice-Bob's accuracy quickly reached 0.9747 at 500 epochs and did not change after that. Eve's accuracy ascended to 0.4778 at 200 epochs and swiftly descended to 0.0214 at 700 epochs, where it stayed. In the third scenario, with a learning rate of 0.0016, Alice-Bob's accuracy sharply increased to 0.9817 at 300 epochs and remained constant, except for a downward spike of 0.6710 at 3000 epochs. Eve's accuracy sharply rose to 0.5537 at 100 epochs, sharply fell to 0.0120 at 500 epochs, and then remained constant. According to the results, the learning rate of 0.0008 was the most effective for both Alice-Bob and Eve regarding bit error and accuracy.



(a) (b)

Figure 5. The results of Alice-Bob and Eve for learning rates of 0.0001, 0.0008, and 0.0016: (a) bits error and (b) accuracy

The performance and security of the hybrid encryption method were analyzed based on XOR encryption and a GAN model under different hyperparameters. The best hyperparameters for both the training and validation phases were 8,000 epochs, a batch size of 4,096, and a learning rate of 0.0008, which generated the highest accuracy for Alice-Bob on the new validation dataset. The worst hyperparameters were 8,000 epochs, a batch size of 4,096, and a learning rate of 0.0001, which resulted in low accuracy for Alice-Bob. These results show that the hybrid encryption method based on XOR encryption and a GAN model effectively and securely encrypts messages, as it can prevent unauthorized access and ensure confidential communication. The results also indicate that the model's performance depends on the configuration of hyperparameters.

4. CONCLUSION

Integrating XOR encryption with a GAN neural network was highly effective in encrypting messages, offering strong security against unauthorized access. The best configuration consisted of training for 8,000 epochs using a batch size of 4,096 and a learning rate of 0.0008. These parameters reliably achieved an impressive accuracy of 100% on Alice-Bob. In comparison, Eve achieved an accuracy of 52.1423%, highlighting this configuration's success in enabling secure communication and protecting

confidential information from unwanted parties. It is suggested that future research examine the scalability of the model and its performance on larger datasets. It can also be beneficial to consider practical situations in which the system will be used and potential malicious attacks to enhance its ability to deal with real-world challenges.

ACKNOWLEDGEMENTS

This research is fully supported by a Fundamental Research Grant Scheme (FRGS) grant from the Ministry of Higher Education (MOHE), FRGS/1/2021/ICT09/UITM/02/1. The authors fully acknowledged MOHE and Universiti Teknologi MARA for the approved fund, making this vital research viable and effective.

REFERENCES

[1] A. M. Qadir and N. Varol, "A review paper on cryptography," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, 2019, pp. 1-6, doi: 10.1109/ISDFS.2019.8757514.
[2] C. Beaver, "Adventures in cryptology: Exploration-worthy project topics," *PRIMUS*, vol. 34, no. 1, pp. 13-31, 2024, doi: 10.1080/10511970.2023.2214924.
[3] H. Harkat, L. M. Camarinha-Matos, J. Goes, and H. F. T. Ahmed, "Cyber-physical systems security: A systematic review," *Computers & Industrial Engineering*, vol. 188, p. 109891, 2024, doi: 10.1016/j.cie.2024.109891.
[4] P. Saraswat, K. Garg, R. Tripathi, and A. Agarwal, "Encryption algorithm based on neural network," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1-5, doi: 10.1109/IoT-SIU.2019.8777637.
[5] Y. Qin and F. Liu, "Convolutional neural network-based polar decoding," in *2019 2nd World Symposium on Communication Engineering (WSCE)*, 2019, pp. 189-194, doi: 10.1109/WSCE49000.2019.9040920.
[6] E. Abdullah, K. Dimyati, W. N. W. Muhamad, N. Izzati Shuhaimi, R. Mohamad, and N. M. Hidayat, "Deep learning based asymmetrical autoencoder for PAPR reduction of CP-OFDM systems," *Engineering Science and Technology, an International Journal*, vol. 50, p. 101608, 2024, doi: 10.1016/j.jestch.2023.101608.
[7] A. E. Muñoz-Zavala, J. E. Macías-Díaz, D. Alba-Cuéllar, and J. A. Guerrero-Díaz-de-León, "A literature review on some trends in artificial neural networks for modeling and simulation with time series," *Algorithms*, vol. 17, no. 2, p. 76, 2024, doi: 10.3390/a17020076.
[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53-65, 2018, doi: 10.1109/MSP.2017.2765202.
[9] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588-598, 2017, doi: 10.1109/JAS.2017.7510583.
[10] H. Ahn and S. Chung, "Deep learning-based anomaly detection for individual drone vehicles performing swarm missions," *Expert Systems with Applications*, vol. 244, p. 122869, 2024, doi: 10.1016/j.eswa.2023.122869.
[11] W. Yao *et al.*, "Dynamics analysis and image encryption application of Hopfield neural network with a novel multistable and highly tunable memristor," *Nonlinear Dynamics*, vol. 112, no. 1, pp. 693-708, 2024, doi: 10.1007/s11071-023-09041-1.
[12] C. Wang, D. Tang, H. Lin, F. Yu, and Y. Sun, "High-dimensional memristive neural network and its application in commercial data encryption communication," *Expert Systems with Applications*, vol. 242, p. 122513, 2024, doi: 10.1016/j.eswa.2023.122513.
[13] A. S. Nadhan and I. Jeena Jacob, "Enhancing healthcare security in the digital era: Safeguarding medical images with lightweight cryptographic techniques in IoT healthcare applications," *Biomedical Signal Processing and Control*, vol. 88, p. 105511, 2024, doi: 10.1016/j.bspc.2023.105511.
[14] R. M. Jogdand and S. S. Bisalapur, "Design of an efficient neural key generation," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 1, pp. 60-69, 2011, doi: 10.5121/ijaia.2011.2105.
[15] C. Ni and S. C. Li, "Machine learning enabled industrial IoT security: challenges, trends and solutions," *Journal of Industrial Information Integration*, vol. 38, p. 100549, 2024, doi: 10.1016/j.jii.2023.100549.
[16] M. Coutinho, R. de Oliveira Albuquerque, F. Borges, L. García Villalba, and T.-H. Kim, "Learning perfectly secure cryptography to protect communications with adversarial neural cryptography," *Sensors*, vol. 18, no. 5, pp. 1306-1306, 2018, doi: 10.3390/s18051306.
[17] Y. Zhao, S. Zhang, Q. Tu, and X. Li, "A hybrid AES encryption for IoT using adversarial network," in *2021 7th International Conference on Computer and Communications (ICCC)*, 2021, pp. 2096-2100, doi: 10.1109/ICCC54389.2021.9674568.
[18] K. Sooksatra and P. Rivas, "An adversarial neural cryptography approach to integrity checking: Learning to secure data communications," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534000.
[19] I. Meraouche, S. Dutta, S. K. Mohanty, I. Agudo, and K. Sakurai, "Learning multi-party adversarial encryption and its application to secret sharing," *IEEE Access*, vol. 10, pp. 121329-121339, 2022, doi: 10.1109/ACCESS.2022.3223430.
[20] J. W. Lee *et al.*, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30039-30054, 2022, doi: 10.1109/ACCESS.2022.3159694.
[21] E. Degirmenci, I. Ozcelik, and A. Yazici, "Effects of Un targeted adversarial attacks on deep learning methods," in *2022 15th International Conference on Information Security and Cryptography (ISCTURKEY)*, 2022, pp. 8-12, doi: 10.1109/ISCTURKEY56345.2022.9931786.
[22] Y. Karaki and N. Ivanov, "Hyperparameters of multilayer perceptron with normal distributed weights," *Pattern Recognition and Image Analysis*, vol. 30, no. 2, pp. 170-173, 2020, doi: 10.1134/S1054661820020054.
[23] F. Ozkaynak, "Role of NPCR and UACI tests in security problems of chaos based image encryption algorithms and possible solution proposals," in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 621-624, doi: 10.1109/UBMK.2017.8093481.
[24] Y. Kochura *et al.*, "Batch size influence on performance of graphic and tensor processing units during training and inference phases," in *Advances in Computer Science for Engineering and Education II*, 2020, pp. 658-668, doi: 10.1007/978-3-030-16621-2_61.
[25] Y. Wu *et al.*, "Demystifying learning rate policies for high accuracy training of deep neural networks," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1971-1980, doi: 10.1109/BigData47090.2019.9006104.

BIOGRAPHIES OF AUTHORS

Iqbal Amir received a diploma in Electrical Engineering (Electronic) from Universiti Teknologi MARA, Pasir Gudang, in 2018. He later finished B.Eng. degree in Electronics Engineering from Universiti Teknologi MARA, Shah Alam in 2024. His previous research includes analyzing the integration of encryption with machine learning. He can be contacted at email: 272iqbalamir272@gmail.com.

Hamizan Suhaimi received his Master of Science in Electrical Engineering from Universiti Teknologi MARA (UiTM), UiTM Shah Alam. He is a Doctor of Philosophy student at the School of Electrical Engineering, UiTM Shah Alam, Malaysia. He graduated with a Diploma in Electrical Engineering (Electronic) in 2014 and obtained 1st class bachelor's degree (Honor) in Electronic Engineering in 2018. His research interest is mainly in computer networks, especially secure polar codes, channel precoding-based message authentication, and secure fifth generation (5G) networks. He can be contacted at 2021654596@student.uitm.edu.my.

Roslina Mohamad obtained a B. Eng. degree in Electrical Engineering and M. Eng. Science degree from Universiti Malaya, Kuala Lumpur, in 2003 and 2008. She later received a Ph.D. in Aerospace Engineering (Deep Space and Wireless Communications Algorithms) from Universiti Putra Malaysia in 2016. Since 2006, she has worked at the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, as a senior lecturer. She is the head of wireless high-speed network (WHiSNet) research interest group. Her research interests include computing algorithms and digital signal processing for deep space communication, channel coding, information-theoretic security, computation theory, internet of things, and wireless communication. She can be contacted at email: roslina780@uitm.edu.my.

Ezmin Abdullah is a senior lecturer from School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA (UiTM), Selangor, Malaysia since 2017. She graduated from Hirosaki University, Japan in 2006 with her bachelor's degree, majoring in Electronics and Information Systems Engineering. She received her master's, major in telecommunication systems and Ph.D. degree in electrical engineering from UiTM in 2013 and 2017 respectively. During her service in UiTM, she pursue her post-doctoral studies at Universiti Malaya in 2021 on the topic of energy issues in the radio access network for 5G and beyond. Her research interest mainly wireless communications, OFDM systems, channel coding, energy management, and the internet of things (IoT). She can be contacted at email: ezmin@uitm.edu.my.

Chuan-Hsian Pu received his B.Eng degree in Electronics and Computing from Nottingham Trent University, UK in 2001 and M. Eng in Telecommunications from Multimedia University, Cyberjaya, in 2005. Later, he received his Ph.D. from UTAR in Telecommunications in 2020. He is currently an assistant professor in the Department of Electrical and Electronic Engineering, at the University of Nottingham Malaysia. His current research interests include channel coding, signal processing, artificial intelligence, and image processing. He can be contacted at email: puchuan.hsian@nottingham.edu.my.