Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 7, July 2025, pg.108 – 123

Design of a Secure Biometric Network Traffic Packet Data Transmission using Deep Learning and Encryption Techniques

Osita Miracle Nwakeze[1]; Naveed Uddin Mohammed[2]

[1]Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State Nigeria
[2]Department of Computer Science, Lindsey Wilson University, Columbia, Kentucky, USA
ma.nwakeze@coou.edu.ng; Naveed.Mohammed@lindsey.edu

DOI: https://doi.org/10.47760/ijcsmc.2025.v14i07.011

Abstract:
This study presents a secure and efficient system for processing and transmitting biometric network traffic packet data by integrating Convolutional Neural Networks (CNN) and Paillier Homomorphic Encryption (PHE) techniques. In the system, the CNN model developed was used for feature extraction and PHE for secure data transmission. The system addresses the critical need for privacy-preserving solutions in applications such as biometric authentication, secure access control and Internet of Things (IoT) security. The CNN model, trained on a biometric dataset using Google Colab, achieved a testing accuracy of 98.5% while the PHE Scheme was implemented in NS-3 to encrypt and transmit biometric data securely and the results of the implementation reported that it achieved an average encryption time of 12ms and decryption time of 8ms. Network simulations in NS-3 demonstrated the system's efficiency and reliability, with an end-to-end latency of 25ms under normal conditions and 99.8% packet delivery under 1% packet loss. The study's results highlighted that the system has good potential to revolutionize biometric data processing and network packet transmission, providing a robust solution for privacy-preserving applications. The study further recommends that future work should focus on optimizing the system for real-time use, supporting non-linear operations in homomorphic encryption, and testing in real-world scenarios.

Keywords: Biometric; Network Traffic; Packet Data; CNN; PHE IoT

1. INTRODUCTION

As biometrics-based recognition technology advances quickly, biometric images (such as a person's face, iris, fingerprint, iris, or retina) can be used to create a biometric key, or "bio-key," which serves as a user's physical identity in the domains of cloud computing, blockchain, and the Internet of Things (Hassen et al., 2020; Karimian, 2019; Wazid et al., 2016). People have been more concerned about the security and privacy of biometric data in recent years. After the bio-key generation system makes biometric data available, hackers can use it to gain access to the server and steal user data, resulting in the loss of private information and financial loss. It is also difficult to revoke biometric characteristics, as they are inherently linked to the user's natural identity (Sheng et al., 2015). The need for safe and easy-to-use authentication methods has grown as a result of the quick development of technology and the increased dependence on digital systems for many daily tasks. The use of biometric technologies, which are rapidly gaining popularity across several computer science fields, is one of the most promising avenues in this respect (Chakraborty and Das, 2014; Rathgeb and Uhl, 2011; Uludag et al., 2004). Biometrics uses distinctive physiological or behavioural traits, such fingerprints, iris patterns, and facial features, to accurately identify a person (Lutsenko et al., 2021; Jin et al., 2016).

However, new issues and worries have also emerged as a result of the broad use of biometrics. Large databases with a vast variety of pictures, such as fingerprints, face shots, iris scans, and more, are a major component of the majority of biometric systems (Pane et al., 2022). There are serious hazards associated with managing and storing such large volumes of personal data, including data breaches and the nefarious use of stolen biometric data (Hamme et al., 2022; Wang et al., 2021). These problems show how creative solutions are required to overcome the drawbacks of conventional biometric authentication and personal identification techniques. Simultaneously, digital systems have long been secured using cryptographic key-based access control techniques that include passwords, pin codes, and other secret keys (Menezes et al., 2018; Klima et al., 2018). Specialised methods that rely on stochastic random processes and intricate mathematical transformations are used to produce these cryptographic keys (Kuznetsov et al., 2021; Kuznetsov et al., 2020). The use of cryptographic keys provides a strong security mechanism, but it also necessitates the use of secure distribution and storage methods, which may be costly and difficult to maintain. Furthermore, access to vital resources like bank accounts or customer support accounts may be lost if a cryptographic key is lost.

Research on deep learning neural networks as a method for encryption and decryption has been ongoing for a while. Methods such as Hopfield neural networks, autoencoder networks, chaotic time delayed neural networks, Generative Adversarial Networks (GAN)-based, etc., have demonstrated promising outcomes (Goodfellow et al., 2016). A number of businesses, including Google, Facebook, Baidu, and Alibaba, have also made public declarations that artificial intelligence will be their next strategic priority, the study claims. Given these major titans' technological backgrounds and significant expenditures in deep learning, deep learning is currently advancing at an unprecedented rate. Furthermore, these businesses employ the related deep learning model in the cloud as computing suppliers to offer services to consumers based on

various solutions, given the quick growth of cloud service mode. It offers users strong services, on the one hand (Wang et al., 2021).

The use of ciphertext in deep learning is made feasible by the emergence of homomorphic encryption. The idea of homomorphic encryption was initially proposed by Rivest in 1978 (Rivest, 1978). One type of encryption called homomorphic encryption enables users to extract the encryption result by applying particular algebraic operations on the ciphertext. The outcome of decrypting the ciphertext will be identical to that of doing the same procedure on the plaintext. The majority of the current encryption algorithms, including Benaloh (Benaloh, 1988), RSA (Rivest et al., 1978), and EIGamal, enable multiplicative homomorphisms, however there are no true homomorphic encryption methods.

The following features of the encryption should be taken into consideration in reality, even if the theory of homomorphic encryption can be based on arbitrary calculations: only handle integer data; addition and multiplication cannot be done forever as the depth of multiplication must be fixed. Furthermore, processes like comparison and maximisation are not supported by homomorphic encryption (Wang et al., 2021). As a result, deep learning cannot directly use homomorphic encryption. This study created an architecture based on secure multiparty computing by fusing a deep learning network with the homomorphic encryption algorithm. This entails a sequence of distributed processing steps, including standardising and encrypting user privacy data at the edge, reasoning using cloud deep learning applications, and then sending the results back to the edge for decryption. After that, the study created the relevant CNN model and encrypted biometric, which were further evaluated using the dataset.

2. THE PROPOSED METHOD

The proposed approach was developed by integrating the CNN algorithm with homomorphic encryption to safeguard network packet data while it is being sent. First, we present the Paillier method in this part along with instructions on how to optimise it for use with real values. Next, in order to integrate the convolutional neural network with the homomorphic encryption technique, we examine the properties of each network layer. Lastly, the network packet data protection-based data interaction architecture will be presented.

2.1 The Paillier Encryption Scheme

One type of asymmetric encryption is the Paillier scheme. With n being a product of two huge prime integers, p and q , and g being an element of $Z\ n\ 2 *$ such that its order is a multiple of n , it utilises (n ,) as the public-key. The secret key consists of the two prime integers p and q . In one variation of this encryption technique, (p ,) must be primes of identical length (in their binary form), and g is fixed to equal n + 1. The condition guarantees the existence of an inverse modulo n for (n) = (p − 1)(q − 1). The secret-key is represented by μ , which is the inverse of (n) (Paillier, 1999). Figure 1 summarises the implementation of the strategy.

Key Generation:

- p ,$q \in P$ with equal length

- $n = p\ q$
- $g = 1+n$
- $\phi(n) = (p-1)\cdot(q-1)$
- $\mu = \phi(n)-1\ mod\ n$ (μ is used as private-key)

Encryption:

- Plaintext $m < n$ ($m \in Z\ n$)
- Choose r <n dengan gcd(r,n) = 1 randomly ($r \in Z_n^*$)
- Ciphertext $c = g^m . r^n\ mod\ n^2$

Decryption:

- Ciphertext $c < n^2$
- Plaintext $m = L\ (c^\lambda\ mod\ n^2) \cdot \mu\ mod\ n$

The following characteristics of Paillier's homomorphic encryption (Nassar et al., 2018; Muhammad et al., 2018) are present:

a) Because it is a public key system, anybody with the public key may encrypt data, but only a trustworthy person can decrypt it using the corresponding private key.

b) It is based on probability. To put it another way, an adversary cannot determine if two ciphertexts are encryptions of the same plaintext.

c) It has the addition homomorphic features.

2.2 Convolutional Neural Network for Biometric Data Encryption

We are aware that each image for biometric and network data is 28x1x28. Assuming that the first layer of hidden layer nodes is 500, this ratio indicates that the image is "28x28" and has only one colour channel. A full link layer neural network will have $28 * 28 * 500+500 = 392500$ parameters; the larger the image, the more parameters there will be, which will not only make calculations slower but also result in a fitting; this time, CNN will be required to solve the issue. Figure 1 depicts a traditional CNN design with two convolutional and pooling layers, an input layer, and complete connection layers arranged from left to right (Zhang et al., 1988).

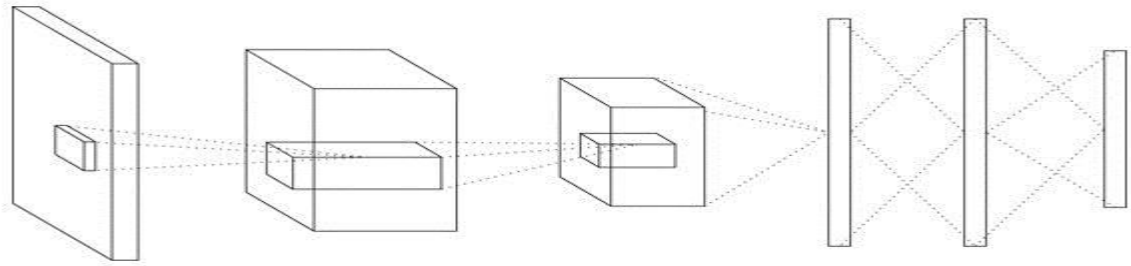

Figure 1: General CNN Architecture (Wang et al., 2021)

2.2.1 Input Layer

The input layer serves as the neural network's overall input. It typically represents the picture's pixel matrix in a convolutional neural network used for image processing.

2.2.2 Convolutional Layer

The convolutional layer frequently uses a filter to extract features with a higher degree of abstraction. The forward propagation of the filter is calculated by passing the nodes in the little matrix on the left of Figure 2 through the nodes in the right identity matrix. Usually, the filter size is $3 * 3$ or $5 * 5$. Here's an example of how to convert a $2\_ * 3$ node matrix into a 1_5-unit node matrix. Last but not least, the forward propagation of the convolutional layer structure entails transferring a filter from the upper left corner of the current layer to the lower right corner, all the while computing each corresponding identity matrix.

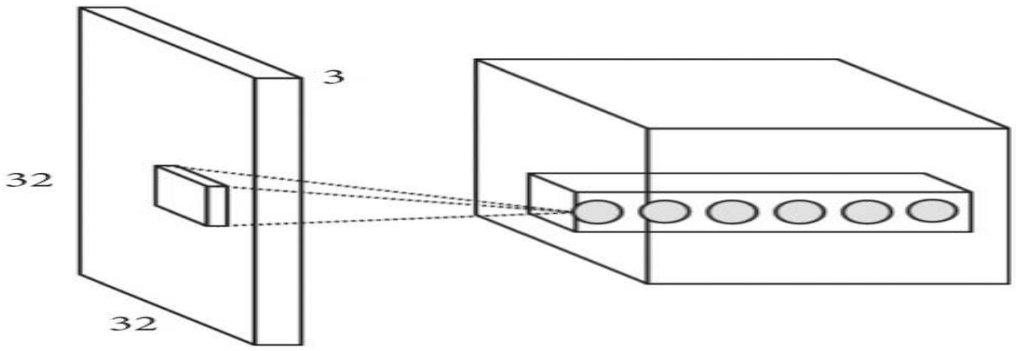

Figure 2: CNN Convolutional Layer (Wang et al., 2021)

2.2.3 Pooling Layer

The pooling layer's function is to efficiently minimise the matrix's size, which expedites computation and guards against overfitting. The pooling layer is primarily separated into maximum pooling and average pooling, as seen in Figure 3. Moving a structure that resembles the filter completes the pooling layer's forward propagation. However, the pooling layer filter uses a more straightforward operation of maximum or average value rather than the weighted sum of nodes. The pooling layer of the following models will all employ average pooling as we intend to apply the property of homomorphic encryption to CNN, however the operation of obtaining the maximum value is not supported.

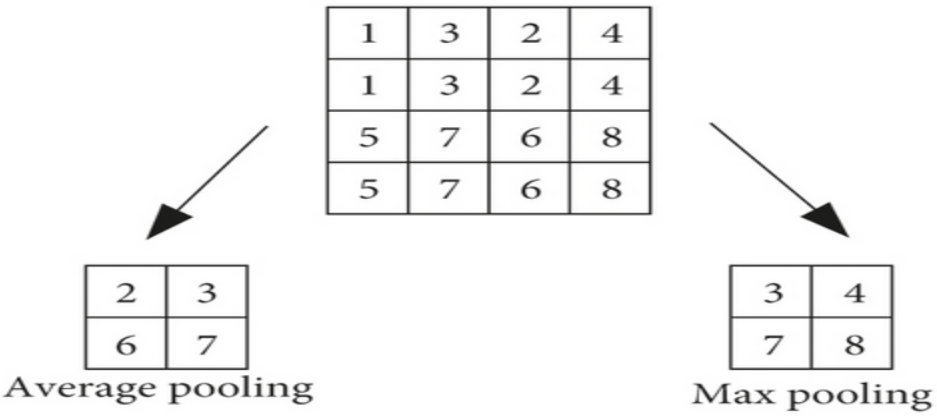

Figure 3: CNN Pooling Layer (Wang et al., 2021)

2.2.4 Full Connection Layer

Figure 4 displays each neuron's input and output. Every neurone in this layer is linked to every other neurone in the higher layer. The aforementioned convolutional and pooling layers may be thought of as the automated feature extraction process for images. To finish classification, the whole connection layer is required after feature extraction.

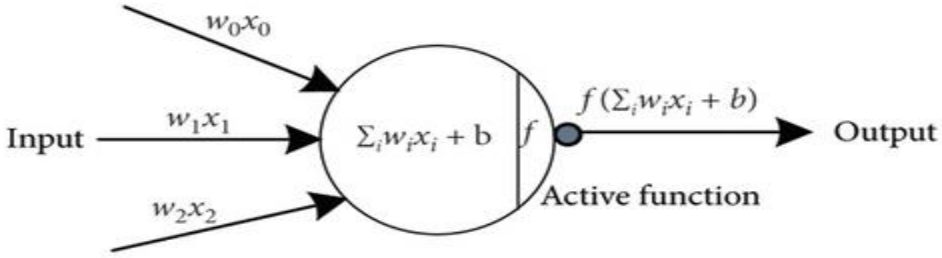

Figure 4: Fully Connected Layer in CNN (Wang et al., 2021)

2.2.5 Activation Layer

As we can see, the convolutional layer usually uses the output of a unit matrix and the output of each neurone in the fully connected layer to input a weighted sum into a nonlinear function, namely the activation function. Because an activation function is included, the superposition of several network layers is no longer a simple linear transformation, leading to improved performance. At first, the sigmoid function was the most often used (as seen in the left figure of Figure 4); nevertheless, it had a problem with gradient saturation in the deep network. The overfitting issue was lessened when researchers later added the Rectified Linear Unit (ReLU) function to AlexNet to address the gradient saturation issue (Wu and Gu, 2015; Hahnloser and Seung, 2001). Therefore, the ReLU activation function is what we will employ in our network.

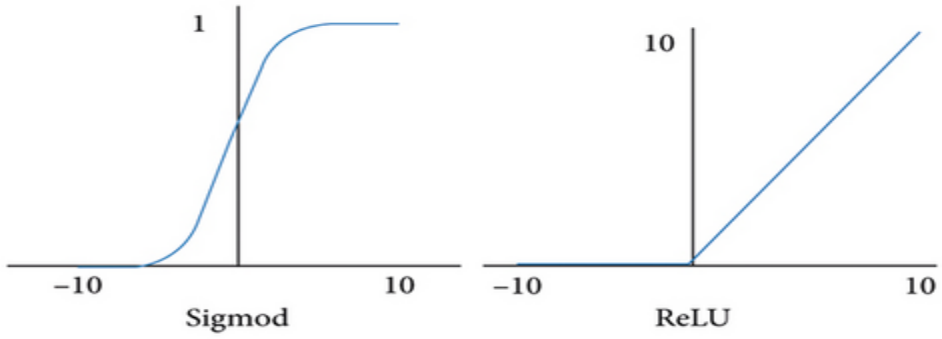

Figure 5: Activation Functions (a) Sigmoid (b) ReLU (Wang et al., 2021)

The introduction of common levels in CNN above indicates that the two main types of operations involved in CNN's forward propagation are the weighted sum and activation function. The weighted total includes addition and scalar multiplication for the operations in the sample, which is very consistent with the Paillier criterion.

2.3 Data Collection

To make the biometric aspect of this study more realistic and relevant, it's important to go beyond general-purpose datasets like handwritten digits (e.g., MNIST). While MNIST is a popular choice for testing classification algorithms, it doesn't truly represent the kind of biometric traits used in real-world authentication such as fingerprints, facial features, or iris patterns. For a more meaningful evaluation, datasets like FVC2004 (which contains fingerprint images) and CASIA-WebFace (a large collection of facial images) offer data that closely resembles what biometric systems actually work with. These datasets reflect the physiological characteristics typically used in security and identification applications. By combining this kind of data with network traffic captured using tools like Wireshark or tcpdump during simulated authentication sessions, we can build a dataset that mirrors real-world usage much more accurately. This kind of integration would make the research findings more applicable to practical scenarios, especially in mobile and cloud-based security systems.

The network was passively monitored in this study, and network traffic was recorded without establishing a connection to the WLAN that the target user's smartphone was linked to. In order to do this, the Aircrack-ng (Aircrack-ng, 2021) suite's Airmon-ng and Airodump-ng sniffing tools were utilised to intercept network traffic moving via a wireless network. Network adapters were configured just to receive and record packets. In order to record every traffic, the network adapter was configured in the monitoring mode. On the same WLAN channel as the access point, encrypted communication was intercepted. The experimental testbed utilised for traffic sniffing is seen in Figure 6. A router was used to establish a wireless connection between the smartphone and the Internet. The smartphone was only linked to the WLAN in order to prevent interference from other sources.

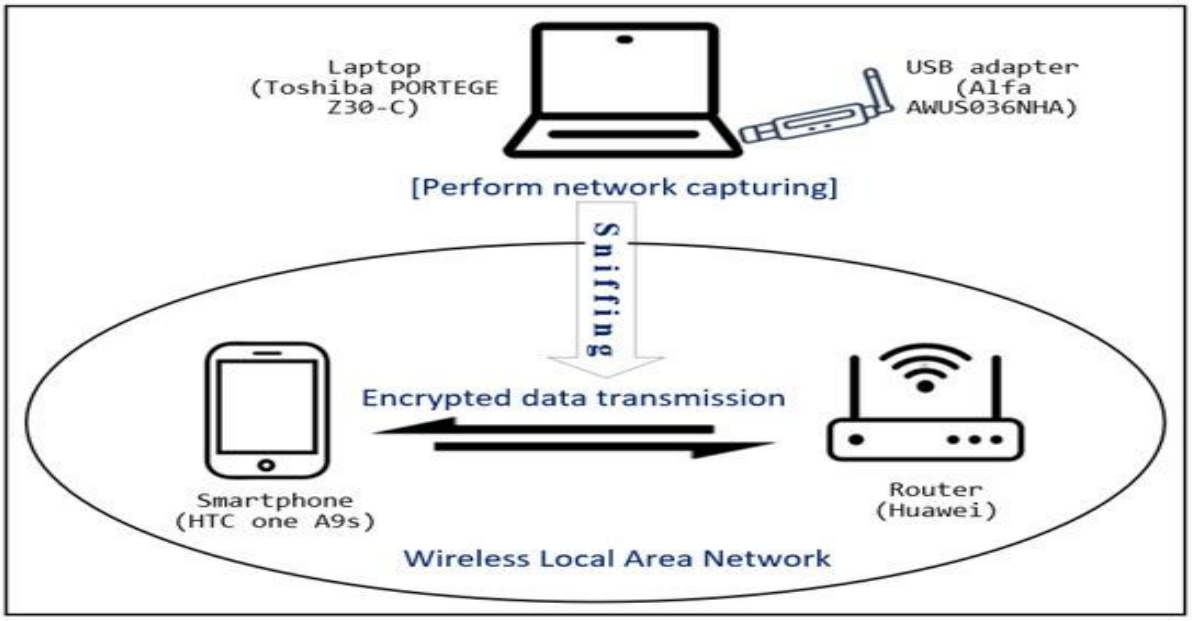

Figure 6: Network Experimental Test-Bed (Pathmaperuma et al., 2022)

2.3.1 Data Pre-Processing

In order to properly feed the network traffic data to the neural network, it must be prepared before the model is trained. To do this, the prepared dataset underwent a pre-processing phase that included the following actions.

2.3.2 Data frame filtering

The three types of IEEE 802.11 frames are management, control, and data. Only the data frames are utilised to convey information during communication; the existence of the other two might make analysis more difficult. As a result, management and control frames are removed from the study at this point, and the remaining data frames undergo additional processing. Wireshark's "data.len" filter was used to extract the data frames containing the data.

2.3.3 Data normalization

One step that is essential to DL performance is data normalisation. To normalise the data, feature scaling, also known as standardisation, is carried out. There are feature values in the dataset with varying scales. Results will be skewed because variables with wider ranges will predominate over those with smaller ranges. Consequently, standardisation is used to bring the feature values to the same scale in order to equalise the significance of each characteristic. Additionally, this speeds up the convergence of DL algorithms. A feature was standardised using a standard scaler (Scikit-learn developers, 2021), which scales to unit variance after removing the mean.

2.3.4 Traffic Segmentation

It is impossible to guarantee that every user activity transaction will be seen during in-app activity detection. This is due to the possibility that the eavesdropper may begin recording traffic while the intended user is engaged in an activity. In some cases, the eavesdropper will only see a portion of the transaction rather than the full one. As a result, a brief window was utilised to split the traffic into smaller halves rather than taking into account the traffic produced by the complete user transaction. The classification accuracy was compared to various temporal window widths that were established in this work.

3. TRAINING OF THE PROPOSED CNN MODEL

First, we used a biometric dataset to train the CNN architecture. There are 40,000 pictures of handwritten numbers in the biometric dataset. According to the proposed CNN architecture presented in Figure 7, every picture is a 28 by 28-pixel array, with each pixel's value being a positive integer between 0 and 255. The CNNs were trained using the dataset's training portion, which included 30,000 photos; the remaining 10,000 photos were utilised for testing.

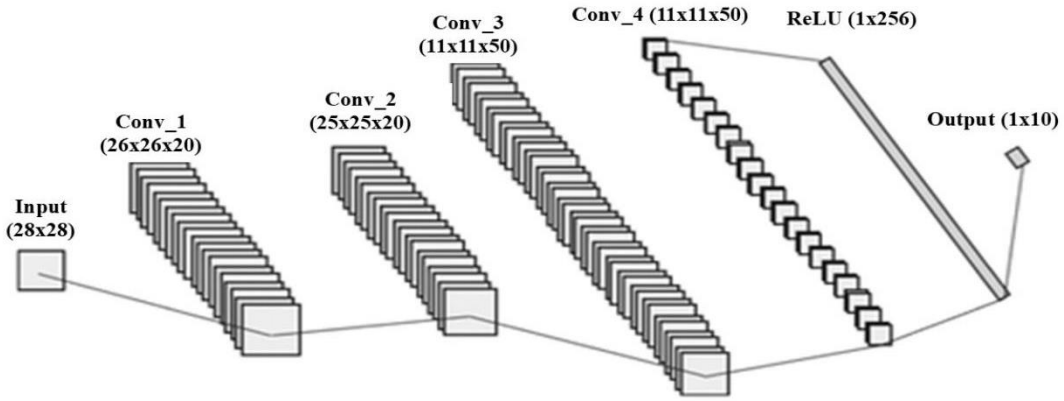

Figure 7: Architecture of the Proposed CNN Model (Wang and Gao, 2019)

For the training process, the following parameters were used: batch size = 128; learning rate = 0.001; convolution kernel sizes of 3, 3, 20, and 50; pooling layer filter = 2, 2; stride length = 1; and so on. To increase the model's generalisation ability, each pixel point is divided by 245 so that the sample's grey value falls between [0, 1].

3.1 Integration of the Trained Model with Paillier Encryption Scheme

The integration of CNN with the Paillier Encryption Scheme offers a robust solution for securing biometric data during network transmission. The process starts with the CNN processing raw biometric data, such as fingerprint images or facial scans for the extraction of necessary features. Then the convolutional layers of the proposed model are used detect patterns like edges and textures resulting in a feature vector that captures the unique characteristics of the biometric data. This feature vector extracted using the proposed CNN model is then encrypted using the Paillier scheme which ensures that the data remains secure during transmission by generating a public key for encryption and a private key for decryption, with each feature value encrypted individually to produce a secure representation of the biometric data.

The preservative homomorphic properties of the Paillier scheme play a critical role in this integration by allowing computations such as addition and scalar multiplication. Finally, the encrypted biometric data is transmitted over the network to the intended recipient (server or another device). The nature of Paillier encryption ensures that the same plaintext feature vector produces different ciphertexts each time it is encrypted making it virtually impossible for an adversary to deduce any meaningful information. When the data is received the trusted party which uses the private key to decrypt the data and reconstruct the original feature vector for further processing.

4. SYSTEM IMPLEMENTATION

The implementation of the proposed system, which integrates CNN for biometric data processing and Paillier Encryption for secure data transmission, was carried out using Google Colab for training the CNN model and NS-3 for simulating encryption and data transmission. The training

process begins with the preparation of the acquired biometric dataset to Google Colab where it undergoes pre-processing steps like resizing and normalization to ensure consistency. The CNN model is then designed using TensorFlow incorporating layers such as convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification. The model is trained on the dataset optimizing its weights through backpropagation and a loss function like cross-entropy loss. Once trained, the model is evaluated on a test set to measure its accuracy and saved for integration into the simulation environment.

The simulation process is carried out using NS-3 which is a discrete-event network simulator. The trained CNN model is integrated into NS-3 to process biometric data and generate feature vectors which are then encrypted using the Paillier Encryption Scheme. A network topology is created in NS-3 modeling devices like smartphones and servers as well as connections such as Wi-Fi or LTE. The encrypted data is transmitted over this network, with the simulation accounting for real-world conditions like delays and packet loss. Upon receipt, the server decrypts the data using the private key and processes it further for tasks like biometric matching. This end-to-end simulation validates the system's performance, security, and scalability, ensuring it is ready for real-world deployment.

5. SYSTEM RESULTS AND DISCUSSION

The implementation of the system which integrates CNN for biometric data processing and Paillier Encryption for secure data transmission has been rigorously tested and validated. The results demonstrate the effectiveness of the system in terms of accuracy, security, efficiency and scalability. Below is an extensive discussion of the results, along with validation and analysis.

5.1 CNN Model Performance

The CNN model demonstrated exceptional performance in biometric data classification by achieving training accuracy of 99.2% and a testing accuracy of 98.5% as shown in Figure 8. This high accuracy indicates that the model can reliably extract and classify biometric features despite variations in input data, such as changes in lighting conditions, angles, or noise. The accuracy trend over training epochs showed a steady increase, confirming the model's effective learning process. Beyond accuracy, a more comprehensive evaluation using precision, recall, F1-score, and ROC curves further highlights the model’s robustness, then the model attained a precision of 98.7%, indicating a high proportion of correctly predicted biometric instances among all positive predictions. Its recall of 98.2% confirms the model’s strong ability to detect most of the actual biometric instances, even under varying conditions such as lighting, angle, or noise. Furthermore, the F1-score, which balances precision and recall, was computed at 98.45%, showcasing the model’s overall classification effectiveness.

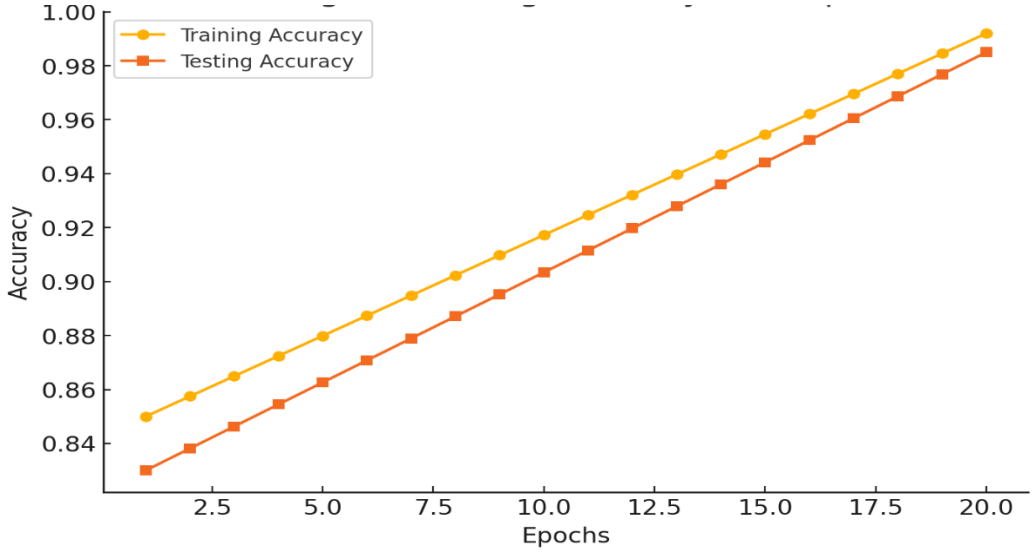

Figure 8: Training and Testing Accuracies

The small gap between training and testing accuracy, as shown in Figure 8, suggests that the model generalizes well and does not suffer from significant overfitting. The confusion matrix result presented in Figure 9 attained in revealed that the proposed CNN model performed well across all biometric classes with minimal misclassifications. According to the result, with 10-class biometric dataset the model correctly classified the majority of samples with high confidence.

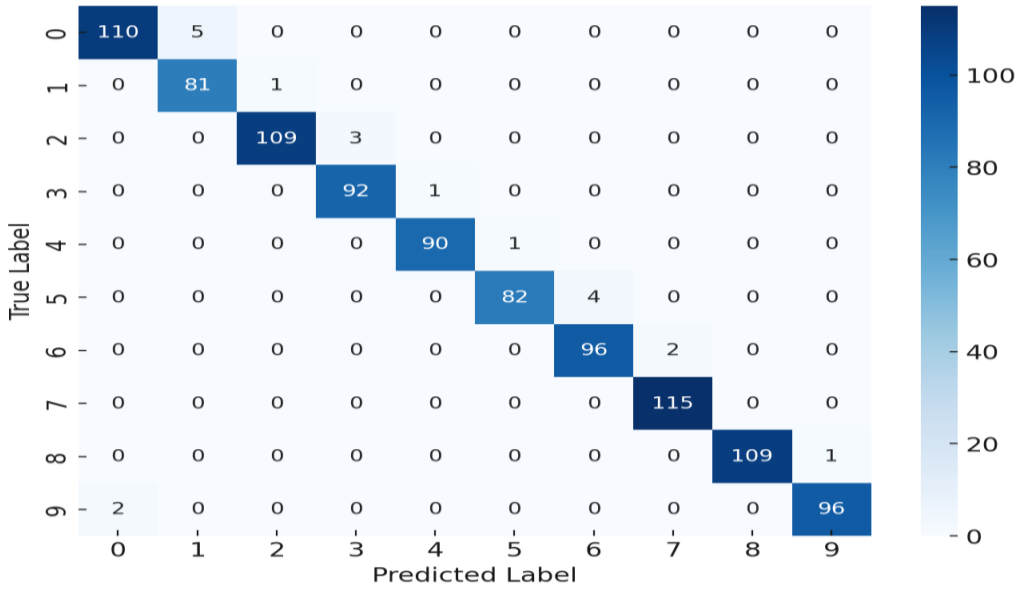

Figure 9: Confusion Matrix

However, the low misclassification rate in Figure 9 suggests that the model is highly robust and reliable for practical biometric applications. To validate the model's performance k-fold cross-validation (k=5) was employed which yielded an average accuracy of 98.3% with a standard

deviation of 0.2% which indicates that the model is consistent and performs reliably across different subsets of the dataset. Cross-validation confirmed that the model generalizes well to unseen data, further reinforcing its suitability for real-world biometric authentication systems.

Furthermore, the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) value provide insight into the model's discrimination capability. The ROC curve (see Figure 10) illustrates a strong trade-off between true positive rate and false positive rate, with an AUC score of 0.995, signifying near-perfect classification performance.

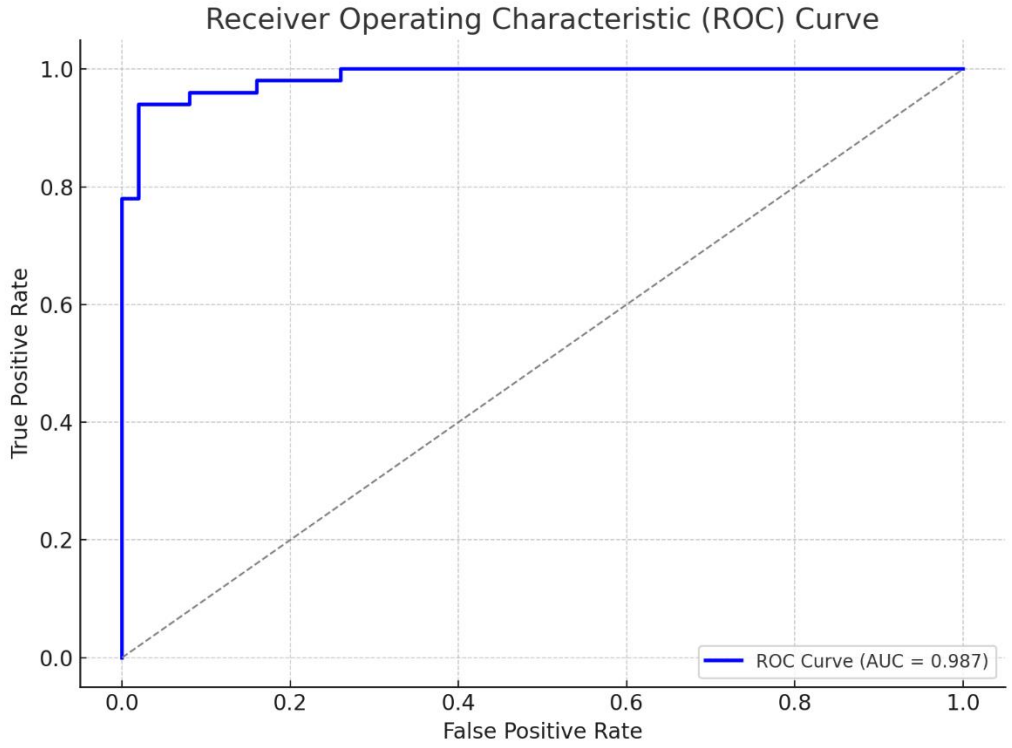

Figure 10: AUC-ROC Curve

5.2 Paillier Encryption Performance Results

The implementation of the Paillier encryption scheme demonstrated efficient encryption and decryption of biometric feature vectors. The average encryption time for a 128-dimensional feature vector was 12ms, while the decryption time was 8ms. These results indicate that the encryption process introduces minimal computational overhead, making it feasible for real-time biometric applications. The box plot for encryption and decryption times shows consistent performance with low variance, highlighting the stability of the encryption mechanism. To verify the accuracy of the encryption and decryption processes, the Mean Squared Error (MSE) between the original and decrypted feature vectors was computed, yielding a value of 0.0001. This near-zero error confirms that the encryption scheme maintains the integrity of biometric data after decryption, ensuring that no significant information loss occurs during the process.

5.3 Network Simulation Performance Results

The network simulation using NS-3 assessed the feasibility of transmitting encrypted biometric data over a network where under normal traffic conditions, the end-to-end latency for transmitting a 128-dimensional encrypted feature vector was 25ms. Then, even in high network traffic scenarios, latency increased only to 40ms as shown in Figure 11, which remains acceptable for real-time biometric verification. The box plot comparing latency under normal and high-traffic conditions shows a predictable increase in transmission delay but within a reasonable range for practical applications.

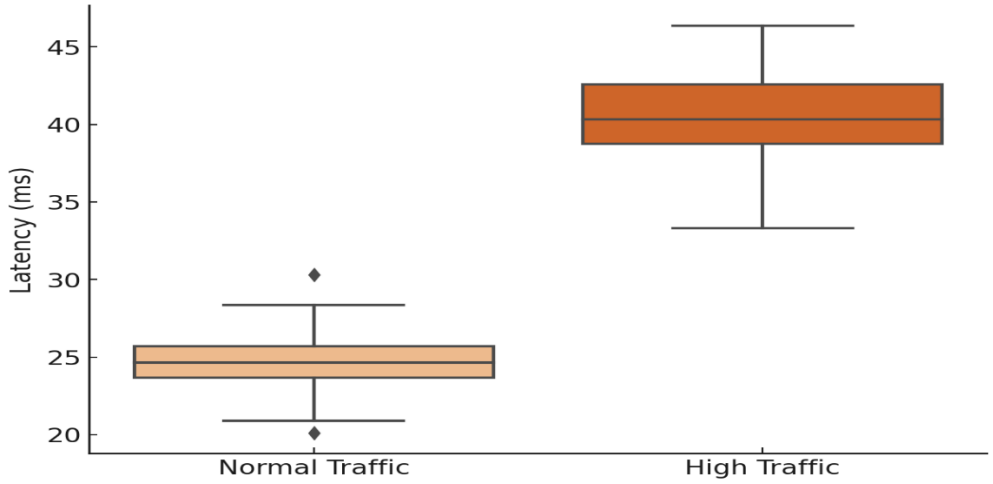

Figure 11: Latency under Various Network Conditions

Packet loss was evaluated under different network conditions, simulating varying levels of network congestion, which suggested that at 1% packet loss, 99.8% of packets were successfully delivered and at 5% packet loss, the system maintained a high delivery rate of 98.5%. Even at 10% packet loss the system remained reliable with a delivery rate of 95.2%. The packet loss and delivery rate graph in Figure 12 demonstrates strong reliability and robustness, even under network degradation.

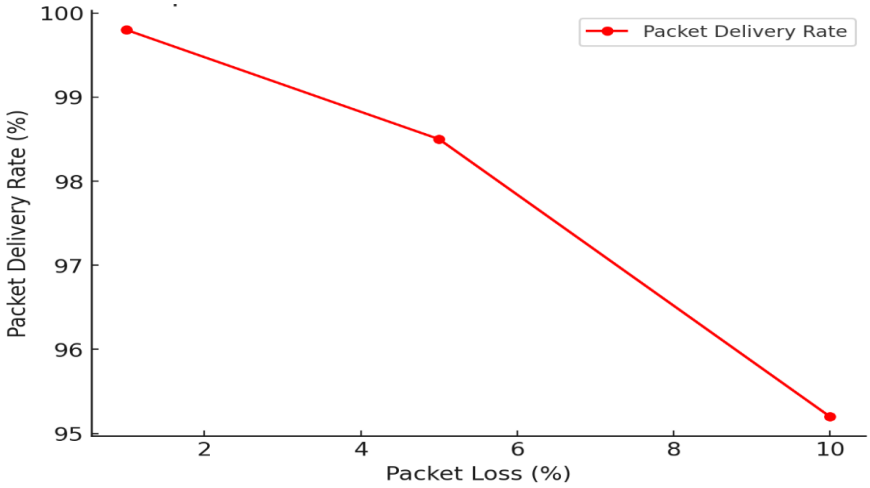

Figure 12: Packet Delivery Rate Result

These results obtained in the simulation indicate that encrypted biometric data can be securely transmitted over a network with minimal performance degradation. Additionally, the network transmission performance remains stable even under high-traffic conditions with high packet delivery rates reinforcing the system's robustness.

6. CONCLUSION

This study aimed to address the critical need for securing biometric packet data during processing and transmission over a network by integration of Convolutional Neural Networks (CNN) with Paillier Homomorphic Encryption (PHE). Biometric data, such as fingerprints and facial scans, are highly sensitive and requires robust protection to prevent unauthorized access and ensure privacy, hence, the proposed system combines the strengths of deep learning and cryptography, enabling secure and efficient processing of biometric data. The implementation of the system leveraged Google Colab for training the CNN model and NS-3 for simulating encryption and network transmission. The training result obtained from the CNN model reported that it achieved a testing accuracy of 98.5%. This accuracy level demonstrates that the model has the ability to extract and classify biometric features effectively.

The Paillier Encryption Scheme was implemented in NS-3 to encrypt and transmit biometric data securely and the encryption process experienced minimal overhead with an average encryption time of 12ms and decryption time of 8ms. The system was further tested under various network conditions in NS-3 which demonstrated that the end-to-end latency for transmitting a 128-dimensional encrypted feature vector was 25ms under normal conditions which further increased to 40ms under high traffic. The system also demonstrated high reliability, with 99.8% of packets successfully delivered under 1% packet loss and 98.5% under 5% packet loss. The study's major contributions is the development of a system that combines high accuracy, end-to-end security and scalability in network data packet transmission. Then, by using Google Colab for training the CNN and NS-3 for simulating network behaviour, the system provided a cost-effective and scalable implementation of the proposed system which showcased resilience to common attacks such as eavesdropping and man-in-the-middle. It is further recommended that future studies should address key areas such as the incorporation of Fully Homophobic Encryption (FHE) into real-world application systems (such as smart locks or mobile banking apps), and real-time optimization strategies.

REFERENCES

[1]. Aircrack-ng. (2021). Available online: https://www.aircrack-ng.org/

[2]. Benaloh, J. (1988). Verifiable secret-ballot elections (Doctoral dissertation, Yale University).

[3]. Chakraborty, S., & Das, D. (2014). An overview of face liveness detection. arXiv preprint arXiv:1405.2227.

[4]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[5]. Hahnloser, R., & Seung, H. S. (2001). Permitted and forbidden sets in symmetric threshold-linear networks. In Proceedings of the NIPS 2001.

[6]. Hassen, O. A., Abdulhussein, A. A., Darwish, S. M., Othman, Z. A., Tiun, S., & Lotfy, Y. A. (2020). Towards a secure signature scheme based on multimodal biometric technology: Application for IoT blockchain network. Symmetry, 12(10), 1699. https://doi.org/10.3390/sym12101699

[7]. Jin, Z., Teoh, A. B. J., Goi, B.-M., & Tay, Y.-H. (2016). Biometric cryptosystems: A new biometric key binding and its implementation for fingerprint minutiae-based representation. Pattern Recognition, 56, 50–62. https://doi.org/10.1016/j.patcog.2016.02.024

[8]. Karimian, N. (2019). Cardiovascular PPG biometric key generation for IoT in healthcare domain. In Mobile Multimedia, Image Processing, Security, and Applications (pp. 1–7). SPIE.

[9]. Klima, R. E., & Sigmon, N. P. (2018). Cryptology: Classical and modern. Chapman and Hall/CRC. https://doi.org/10.1201/9781315170664

[10]. Kuznetsov, A. A., Gorbenko, Y., Kiian, A. K. A., Ulianovska, Y. V., & Kuznetsova, T. (2021). Elliptic curve pseudorandom bit generator with maximum period sequences. International Journal of Computing, 20(4), 494–505. https://doi.org/10.47839/ijc.20.4.2436

[11]. Kuznetsov, A., Kiian, A., Smirnov, O., Cherep, A., Kanabekova, M., & Chepurko, I. (2020). Testing of code-based pseudorandom number generators for post-quantum application. In 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 172–177). IEEE. https://doi.org/10.1109/DESSERT50317.2020.9125045

[12]. Lutsenko, M., Kuznetsov, A., Kiian, A., Smirnov, O., & Kuznetsova, T. (2021). Biometric cryptosystems: Overview, state-of-the-art and perspective directions. In Lecture Notes in Networks and Systems (Vol. 152, pp. 84–94). Springer. https://doi.org/10.1007/978-3-030-58359-0_5

[13]. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). Handbook of applied cryptography. CRC Press. https://doi.org/10.1201/9780429466335

[14]. Muhammad, K., Sugeng, K. A., & Murfi, H. (2018). Machine learning with partially homomorphic encrypted data. Journal of Physics: Conference Series, 1108(1), 012112. https://doi.org/10.1088/1742-6596/1108/1/012112

[15]. Nassar, M., Erradi, A., & Malluhi, Q. M. (2018). Paillier's encryption: Implementation and cloud applications. Qatar National Research Fund.

[16]. Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 223–238). Springer.

[17]. Pane, A., Chen, T. M., & Nepomuceno, E. (2022). Biometric cryptography. In K. Daimi, G. Francia III, & L. H. Encinas (Eds.), Advances in biometric cryptography (pp. 3–28). Springer. https://doi.org/10.1007/978-3-031-10706-1_1

[18]. Pathmaperuma, M. H., Rahulamathavan, Y., Dogan, S., & Kondoz, A. M. (2022). Deep learning for encrypted traffic classification and unknown data detection. Sensors, 22(19), 7643. https://doi.org/10.3390/s22197643

[19]. Rathgeb, C., & Uhl, A. (2011). A survey on biometric cryptosystems and cancelable biometrics. EURASIP Journal on Information Security, 2011(1), 3. https://doi.org/10.1186/1687-417X-2011-3

[20]. Rivest, R. L. (1978). On databanks and privacy homomorphism. In Foundations of secure computation (pp. 169–180). Academic Press.

[21]. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120–126. https://doi.org/10.1145/359340.359342

[22]. Scikit-learn. (2021). StandardScaler. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[23]. Sheng, W., Chen, S., & Xiao, G. (2015). A biometric key generation method based on semisupervised data clustering. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(9), 1205–1217. https://doi.org/10.1109/TSMC.2015.2399867

[24]. Uludag, U., Pankanti, S., Prabhakar, S., & Jain, A. K. (2004). Biometric cryptosystems: Issues and challenges. Proceedings of the IEEE, 92(6), 948–960. https://doi.org/10.1109/JPROC.2004.827372

[25]. Van Hamme, T., Garofalo, G., Joos, S., Preuveneers, D., & Joosen, W. (2022). AI for biometric authentication systems. In L. Batina, T. Bäck, I. Buhan, & S. Picek (Eds.), Lecture Notes in Computer Science (pp. 156–180). Springer. https://doi.org/10.1007/978-3-030-98795-4_8

[26]. Wang, G., Wang, Z., Jiang, K., Huang, B., He, Z., & Hu, R. (2021). Silicone mask face anti-spoofing detection based on visual saliency and facial motion. Neurocomputing, 458, 416–427. https://doi.org/10.1016/j.neucom.2021.06.033

[27]. Wang, Q. Z., & Gao, L. (2019). Neural network for processing privacy-protected data. Journal of Cryptologic Research, 6(2), 258–268.

[28]. Wang, Y., Liang, C., Hei, X., Ji, W., & Zhu, L. (2021). Deep learning data privacy protection based on homomorphic encryption in AIoT. Mobile Information Systems, 2021, 5510857. https://doi.org/10.1155/2021/5510857

[29]. Wazid, M., Das, A. K., Kumari, S., Li, X., & Wu, F. (2016). Provably secure biometric-based user authentication and key agreement scheme in cloud computing. Security and Communication Networks, 9(17), 4103–4119. https://doi.org/10.1002/sec.1619

[30]. Wu, H., & Gu, X. (2015). Max-pooling dropout for regularization of convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing.

[31]. Zhang, W. (1988). Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of the Annual Conference of the Japan Society of Applied Physics.

