frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 29 April 2025
DOI 10.3389/frai.2025.1545946

OPEN ACCESS

EDITED BY
Rashid Ibrahim Mehmood,
Islamic University of Madinah, Saudi Arabia

REVIEWED BY
Aditya Kumar Sahu,
Amrita Vishwa Vidyapeetham University, India
Oleksandr Kuznetsov,
University of eCampus, Italy

*CORRESPONDENCE
Tsehayu Gizachew Yirga
✉ tsehayu.gizachew2016@gmail.com

RECEIVED 16 December 2024
ACCEPTED 31 March 2025
PUBLISHED 29 April 2025

CITATION
Gizachew Yirga T, Gizachew Yirga H and Addisu EG (2025) Cryptographic key generation using deep learning with biometric face and finger vein data. *Front. Artif. Intell.* 8:1545946. doi: 10.3389/frai.2025.1545946

Cryptographic key generation using deep learning with biometric face and finger vein data

Tsehayu Gizachew Yirga[1]*, Hailu Gizachew Yirga[2] and Eshetie Gizachew Addisu[3]

[1]Department of Computer Science, College of Computing and Informatics, Mekdela Amba University, Tulu Awlia, Ethiopia, [2]Department of Computer Science, College Informatics, University of Gondar, Gondar, Ethiopia, [3]Department of Information Systems, College of Informatics, University of Gondar, Gondar, Ethiopia

This research proposes a novel approach to cryptographic key generation using biometric data from face and finger vein modalities enhanced by deep learning techniques. Using pretrained models FaceNet and VGG19 for feature extraction and employing a Siamese Neural Network (SNN), the study demonstrates the integration of multimodal biometrics with fuzzy extractors to create secure and reproducible cryptographic keys. Feature fusion techniques, combined with preprocessing and thresholding, ensure robust feature extraction and conversion to binary formats for key generation. The model demonstrates impressive accuracy with a vector converter, achieving a sigma similarity of 93% and a sigma difference of 64.0%. Evaluation metrics, including False Acceptance Rate (FAR) and False Rejection Rate (FRR), indicate significant improvements, achieving FRR < 3.4% and FAR < 1%, outperforming previous works. Additionally, the adoption of Goppa code-based cryptographic systems ensures post-quantum security. This study not only enhances biometric cryptography's accuracy and resilience but also paves the way for future exploration of quantum-resistant and scalable systems.

KEYWORDS

deep learning, SNN, multimodal biometrics, post-quantum security, cryptography key

1 Introduction

In the age of digital transformation, ensuring the security of personal data has become more critical than ever. Traditional authentication systems, relying on passwords or PINs, are increasingly vulnerable to cyber-attacks and data breaches (Raja, 2024). Biometric authentication, which uses unique physical or behavioral traits such as fingerprints, face recognition, and vein patterns, offers a more secure and user-friendly alternative. Among these, the fusion of facial and finger vein biometrics has gained significant attention due to their robustness and accuracy. The process of extracting cryptographic keys directly from these biometric features, through techniques such as fuzzy extractors, promises a novel approach to secure data encryption and authentication (Akintoye and Akinwamide, 2024; Babu et al., 2024; Shamili Shanmugapriya et al., 2024).

Fuzzy extractors are cryptographic tools designed to generate secure keys from noisy biometric data. The concept leverages error-correcting codes to recover biometric traits even in the presence of minor variations, making them suitable for real-world applications where data are not always perfectly consistent. Traditional fuzzy extractors were primarily focused on single-modal biometric data, such as fingerprints or iris patterns. However, with the

advancement of deep learning and multimodal biometrics, there is growing potential to combine face and finger vein recognition to generate more reliable and secure cryptographic keys (Shevchenko and Anikin, 2023; Kuznetsov et al., 2018).

Deep learning, particularly convolutional neural networks (CNNs), has revolutionized how we process and understand biometric data. Pretrained models such as FaceNet for face recognition and specialized CNNs for finger vein recognition have demonstrated exceptional performance in extracting discriminative features from biometric images. These features can then be fused to create a unique cryptographic key (Kuznetsov et al., 2022; Yao et al., 2024). This approach not only enhances the accuracy of biometric authentication but also improves the robustness of the generated keys, making them less susceptible to errors caused by environmental factors or user behavior.

Traditional authentication techniques, such as PINs and passwords, are susceptible to a number of online attacks. Although biometric authentication provides a more secure option, current methods for generating biometric cryptographic keys have issues with scalability, security, and consistency. Because of biometric variances, previous studies that concentrated on unimodal biometric data, such as fingerprints, faces, or iris, had high False Acceptance Rates (FARs) and False Rejection Rates (FRRs) (Bele, 2024).

This research introduces a multimodal biometric cryptographic key generation framework using deep learning. By fusing face and finger vein features, the proposed method enhances security and reliability over single-modal approaches. Additionally, incorporating a McEliece cryptosystem with Goppa codes ensures resistance against quantum computing attacks. The experimental results confirm the superiority of this method, achieving FRR < 3.4% and FAR < 1%, outperforming prior works that report FRR of 8.3% and FAR of 7.4%.

2 Related work

Biometric authentication has gained widespread attention as a robust alternative to traditional password-based security systems. Among various biometric modalities, face and finger vein recognition stand out due to their high accuracy, non-intrusiveness, and resilience to spoofing attacks. Face recognition has become a prominent method due to its ease of use and widespread availability through cameras in smartphones and other devices. Recent advancements in deep learning, particularly with models such as FaceNet, have significantly improved face recognition accuracy by learning discriminative features directly from large datasets (Sydor et al., 2024). Studies have reported a face recognition accuracy of 97.35% on the Labeled Faces in the Wild (LFW) dataset, showcasing the reliability of deep learning-based approaches. However, despite these improvements, face recognition is still susceptible to adversarial attacks, occlusions, and variations due to aging or environmental conditions, necessitating additional layers of security.

Studies have reported varying performance metrics for different biometric authentication methods. For instance, the work "Securing the Digital World: A Comprehensive Guide to Multimedia Security" achieved a False Rejection Rate (FRR) of 2.5% and a False Acceptance Rate (FAR) of 1.8%, demonstrating a strong balance between security and usability. Implementation and Analysis of Digital Watermarking Techniques for Multimedia Authentication reported an authentication accuracy of 96.4%, emphasizing the robustness of watermarking techniques for secure biometric verification. Secure and Imperceptible Frequency-Based Watermarking for Medical Images achieved a Peak Signal-to-Noise Ratio (PSNR) of 52 dB and a Structural Similarity Index (SSIM) of 0.98, indicating high imperceptibility and robustness against attacks. Additionally, Robust Medical and Color Image Cryptosystem Using Array Index and Chaotic S-Box recorded an encryption efficiency of 99.2%, ensuring secure transmission of biometric data while maintaining high image quality (Kavitha et al., 2024). These results highlight the effectiveness of different biometric security approaches, but challenges such as reproducibility, robustness against adversarial attacks, and computational efficiency remain significant concerns. Our work addresses these issues by integrating multimodal biometric fusion with optimized cryptographic key generation techniques, ensuring a more secure and scalable authentication system.

A previous study (Sulavko et al., 2025) presents a secure method for handling user-specific information using a Neural Fuzzy Extractor (NFE). The NFE integrates pre-existing classifiers with fuzzy extractors through an artificial neural network-based expander, maintaining performance while enhancing security. The reported FAR and FRR of 4.5% suggest a trade-off between security and usability. The authors demonstrate the NFE's effectiveness by retrofitting it to classic neural networks for basic biometric authentication scenarios. However, the reliance on neural network-based expansion may introduce computational overhead, and further work is needed to explore its application to multimodal biometrics. Future research could focus on optimizing NFEs for different biometric traits and improving efficiency in large-scale deployments.

Another significant contribution comes from the article "A Secure Biometric Key Generation Mechanism via Deep Learning and Its Application" (Wang et al., 2022). This method utilizes random binary codes to represent biometric data and establishes a relationship between biometric data and the codes for each user. To protect privacy and ensure revocability, a random permutation operation shuffles the binary code to update a new biometric key. A fuzzy commitment module generates helper data without revealing biometric information during enrollment. The method is evaluated using benchmark datasets and outperforms existing methods in terms of the genuine accept rate at a 1% False Acceptance Rate, while also meeting revocability and randomness criteria. However, the need for network retraining limits its applicability in zero-shot enrollment scenarios, where users enroll without retraining the model. Future work should explore ways to improve stability and security under zero-shot conditions to enhance the practicality of biometric key generation systems.

Furthermore, the article "Deep Learning-based Biometric Cryptographic Key Generation with Post-Quantum Security" (Kuznetsov et al., 2023) explores convolutional neural networks for extracting biometric features from human facial images for key generation. Code-based cryptographic extractors process these features, resulting in a low error rate of less than 10%. Post-quantum cryptography enhances the security of generated keys, making them resilient to future computational threats. However, the study primarily focuses on facial biometrics, which alone may not provide the highest level of security due to potential vulnerabilities such as face spoofing. Future research should investigate the integration of multimodal biometrics and further optimization of code-based extractors to improve security and performance. Another method by De Oliveira Nunes et al. (2024) introduces oblivious extractors, which allow

authentication without transmitting helper data (HD) to the client. While this approach enhances privacy by reducing the risk of HD interception, achieving suitable FRR and FAR rates below 10% remains a challenge. The effectiveness of this method could be improved by optimizing feature extraction techniques and integrating stronger cryptographic principles to mitigate statistical and reusability attacks.

Deep learning-based fuzzy extractors show promise for generating cryptographic keys from biometrics. They overcome the limitations of traditional methods and offer improved security and privacy. However, further research is needed to address challenges with accuracy and high error rates because of using unimodal to generate biometric keys. Therefore, we proposed code-based cryptography key generation using deep learning from multimodal biometrics.

Our research extends these works by integrating biometric multimodal fusion, deep learning, and quantum-resistant cryptography, ensuring a more secure and scalable authentication system. By employing FaceNet and VGG19 for feature extraction, a Siamese Neural Network (SNN) for pattern learning, and a McEliece cryptosystem for quantum-resistant security, our approach surpasses existing methods in terms of robustness, accuracy, and resilience against modern cyber threats. Unlike traditional methods that rely on either unimodal biometrics or non-optimized feature fusion, our approach ensures better generalization and resistance to environmental variations, making it suitable for real-world biometric authentication applications.

3 Methodology

3.1 Dataset

Biometric technologies, particularly face and finger vein recognition, have become key in security and authentication, supporting applications from personal devices to large-scale government systems. Their reliability and efficiency make them vital for enhanced security and seamless authentication. This research utilized a dataset from Kaggle, containing 425 images across 85 classes each for face and finger vein data.

3.2 Dataset preprocessing

To prepare face and fingerprint data for machine learning models, preprocessing is essential. Preprocessing for both face and fingerprint data can be carried out as follows:

3.2.1 Preprocessing face dataset

Dataset preprocessing: Preprocessing is essential to prepare face and fingerprint data for machine learning models. Pre-trained Haar cascade classifiers were used to locate faces within images accurately. Detected face images were resized to a uniform dimension of 160 × 160 pixels in Red, Green, Blue (RGB) format. Gaussian blurring was applied to reduce background noise in face images, enhancing clarity and feature distinction. Each class was ensured to have exactly five images using the ImageDataGenerator from Keras. Various transformations were performed, including rotation, shear, zoom, and horizontal flipping, to generate additional samples for underrepresented classes and maintain uniformity across classes.

3.2.2 Preprocessing finger vein dataset

In preprocessing of the finger vein dataset, the focus was on enhancing finger vein images using ROI extraction and contrast improvement. Each image was converted to grayscale, Gaussian blur was applied to reduce noise, and the Canny edge detector was used to extract the region of interest (ROI). The contrast of the extracted ROI was then enhanced using CLAHE (Contrast Limited Adaptive Histogram Equalization). This method ensures efficient and systematic preprocessing of finger vein data for subsequent analysis.

3.3 Feature extractor models

FaceNet was chosen for face feature extraction due to its performance in biometric applications. It creates complex feature vectors, or embeddings, that capture unique facial traits such as landmarks and expressions, which are crucial for differentiating individuals. To produce embeddings that preserve facial similarity and enable accurate face identification and verification, FaceNet uses a deep metric learning technique. It is a great option for challenging facial recognition tasks due to its adaptability in managing various stances, lighting scenarios, and expressions.

VGG19 was selected for fingerprint feature extraction because of its deep architecture, which consists of 19 layers, effectively learns intricate patterns in images, and captures subtle features such as ridges and valleys in fingerprints. It is the ideal choice for accurate fingerprint identification and classification because of its straightforward architecture, performance on large datasets, and dependable generalization to fresh fingerprint photographs. To further improve feature representation, Principal Component Analysis (PCA) is applied to the high-dimensional features that VGG19 has recovered. PCA reduces dimensionality by transforming the data into a more manageable 128-dimensional space and identifying and retaining the most significant primary components.

3.4 Feature fusion

Feature fusion is a technique that combines features from different modalities, such as face and fingerprint data, to create a more comprehensive representation for recognition tasks. This integration can occur at various stages in machine learning, including early fusion, where features are combined at the input level, and late fusion, where they are extracted separately and combined later. Early fusion enhances model performance by capturing connections between modalities simultaneously, requiring only one training step, making it more efficient than late fusion, which involves training multiple models.

3.4.1 SNN model development

An architecture known as a Siamese Neural Network (SNN) is made for tasks that require learning or verifying similarity between three inputs: an anchor, a positive that is similar to the anchor, and a negative that is different from the anchor. Every input is routed via identical subnetworks with the same architecture and weights, producing

high-dimensional embeddings for every input. The network maximizes the distance between different input embeddings and minimizes the distance between similar input embeddings using the triplet loss function. The architecture consists of two dense layers activated by sigmoid and Rectified Linear Unit (ReLU). The shared weights are tuned during training to discriminate between different and similar data points according to their embeddings. The end product is an SNN that can map comparable inputs more closely together while separating them in the embedding space, as shown in Figure 1.

3.5 Features of binary string converter

The function used for the feature vector converter is defined mathematically in Equation 1, which applies a thresholding rule to convert continuous real-valued features into binary format. One part of the extractor recommended for creating reliable keys from biometric photos is the feature vector converter. A feature vector with real-valued elements can be entered into the feature vector converter. Deep learning techniques extract the feature vector from the biometric images. The converter transforms the real-valued feature vector into a binary string using a binarization rule. A threshold value is used to define the binarization rule. The binarization rule compares each element of the feature vector with the threshold value. If an element is greater than the threshold, it is converted to 1; otherwise, it is converted to 0. The converter processes each element of the binarized feature vector and concatenates the binary values to form a binary string representation. The feature vector converter plays a crucial role in transforming the continuous real-valued features extracted from biometric images into a binary format. The binary distance between vectors is calculated using Equation 2, which computes the mean of the absolute differences between the elements of the two vectors. This binary representation is then used in the fuzzy extractor to generate cryptographically strong keys for authentication and security purposes. In general, we have used thresholds to convert the continuous real-valued features extracted from biometric images into a binary format. These are using zero thresholding and mean thresholding values. These values are compared with continuous real-valued features extracted from biometrics, and values below these are converted to zeros and values above these are converted to ones.

The similarity between binary vectors is determined using Equation 3, which is based on the binary distance.

The function used for the feature vector converter is:

The mathematical equation for the to_binary_string function can be written as

$$B_i = \begin{Bmatrix} 1\,if\ \ fi \geq ti \\ 0\,if\ \ fi < ti \end{Bmatrix} \tag{1}$$

Where B_i is the i-th element of the binary output vector, fi is the i-th element of the input vector, and ti is the i-th element of the threshold vector t . This equation applies the rule $I(f > t)$, meaning it returns 1 if the condition $f_i > = t_i$ is satisfied and 0 otherwise for each element i in the vectors.

The equation for the binary distance as implemented in the code can be expressed mathematically as:

$$D(x,y) = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{2}$$

Where $D(x, y)$ is the binary distance between vectors x and y, n is the number of elements in each vector, x_i and y_i are the corresponding elements of vectors x and y, and $|x_i - y_i|$ is the absolute difference

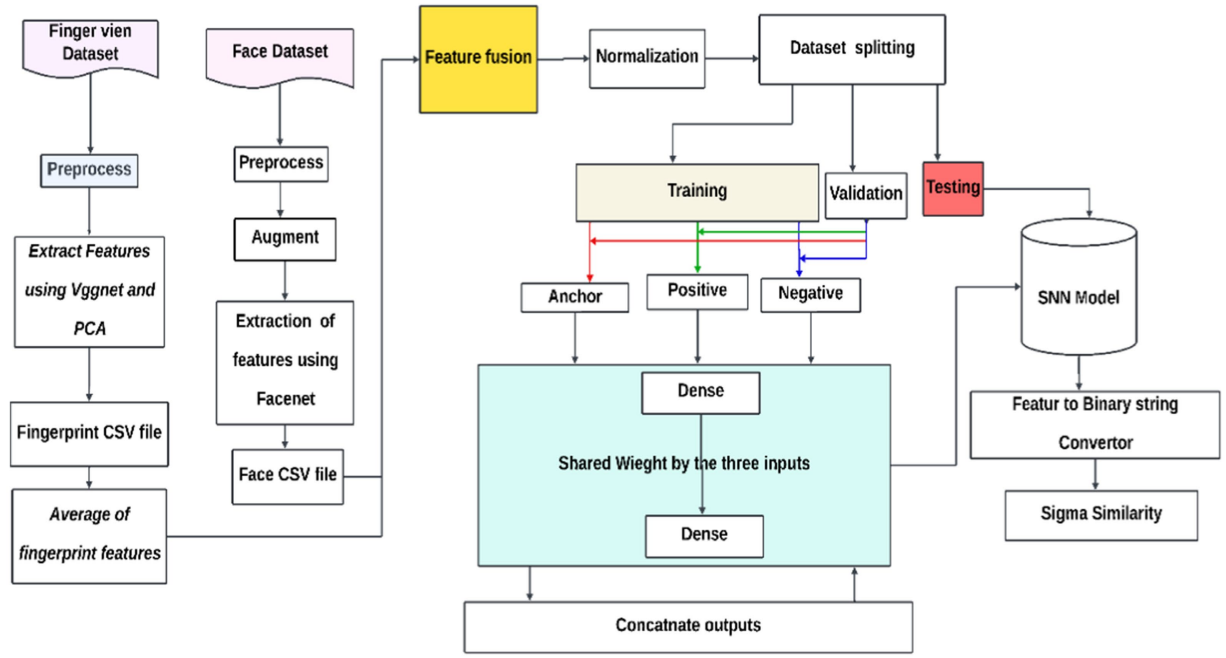

FIGURE 1
Development of the SNN model.

between the elements x_i and y_i. This equation computes the mean of the absolute differences between the elements of the two vectors.

Here is the equation for the similarity based on the binary distance:

$$S(x,y) = 1 - D(x,y) \tag{3}$$

Where $S(x, y)$ is the similarity between vectors x and y and $D(x, y)$ is the binary distance between the vectors, which is computed above.

3.6 Evaluation parameters

The average similarity for items from the same class is calculated using Equation 4, which aggregates similarity scores across feature vectors. When evaluating the precision and efficacy of a system, model, or procedure, evaluation parameters are essential. They offer measurable metrics that aid in assessing a system's performance in many scenarios. These criteria are crucial for determining one's advantages, disadvantages, and potential growth areas. Researchers and developers can optimize the performance of their systems by examining these metrics. In this study, we measured the effectiveness of our biometric authentication system using a number of evaluation factors.

Similarity evaluation ($\hat{\sigma}$ same): This parameter represents the average similarity between binary vectors formed for pairs of the same individual. It measures how closely the binary strings extracted from biometric data of the same person match each other. Higher values of $\hat{\sigma}$ same indicate a higher degree of similarity between the binary vectors, which is desirable for accurate authentication.

For a batch of feature vectors from the same class, the sigma_same (similarity between pairs of vectors in the same class) is calculated as:

The average similarity for items from different classes is computed using Equation 5, which measures cross-class similarity.

$$\hat{\sigma}\,\text{same} = \frac{1}{\text{Nsame}} \sum_{i=1}^{C} \sum_{j=1}^{M} \text{sim}\left(f_j f_{j+1}\right) \tag{4}$$

Hère: $\hat{\sigma}$ same is the averaged similarity (sigma) for items from the same class, C is the total number of unique classes, M is the number of items in the batch corresponding to class C (i.e., the number of vectors in that class), $\text{sim}\left(f_j f_{j+1}\right)$ is the similarity function applied to consecutive pairs of feature vectors, and Nsame is the total number of same-class pairs summed over all classes, which is equal to the sum of $M-1$ for each class.

Similarity evaluation ($\hat{o}$ diff): This parameter represents the average similarity between binary vectors formed for pairs of different individuals. It measures the level of similarity between the binary strings extracted from the biometric data of different individuals. Lower values of $\hat{o}$ diff indicates a higher degree of dissimilarity between the binary vectors, which is important for distinguishing individuals.

$$\hat{\sigma}\,\text{diff} = \frac{1}{\text{Ndiff}} \sum_{i=1}^{C-1} \sum_{j=1}^{\min(Mi,Mi+1)-1} \text{sim}\left(f_{i,j} f_{i+1,j}\right) \tag{5}$$

Hère: $\hat{\sigma}$ diff is the averaged similarity (sigma) for items from different classes. C is the total number of unique classes. Mi and $Mi + 1$ are the number of vectors in class i and class $i + 1$, respectively. $\text{sim}\left(f_{i,j} f_{i+1,j}\right)$ is the similarity between vectors from class i and class i + 1. Ndiff is the total number of cross-class pairs considered, which is equal to the sum of $\min(Mi, Mi+1) - 1$ over all adjacent class pairs.

By using these two parameters, ($\hat{\sigma}$ same) and ($\hat{\sigma}$ diff) can comprehensively evaluate the performance of the biometric authentication system. High ($\hat{o}$ same) values ensure the system reliably recognizes the same individual, enhancing the true positive rate. Conversely, low ($\hat{o}$ diff) values ensure that the system effectively distinguishes between different individuals, reducing the false positive rate. Together, these parameters help balance the trade-off between security and convenience in biometric authentication systems.

3.7 Code-based fuzzy extractor for biometric cryptography

The Code-Based Fuzzy Extractor for Biometric Cryptography is a cryptographic method that uses the McEliece code-based cryptosystem. It can extract error-tolerant, nearly uniform randomness (K) from biometric data (w) and recover it from an analogous input (w'). K can be used as a cryptographic key without the requirement for conventional key storage thanks to this technique, which integrates information-theoretic security with cryptographic systems, even though computational security is frequently relied upon in such applications (Kuznetsov et al., 2018).

By leveraging the strength and resilience of the McEliece cryptosystem against quantum cryptanalysis, the suggested fuzzy extractor enhances the security of biometric cryptography. The extractor ensures accuracy even in errors by correcting biometric image distortions. By doing away with the requirement for a non-secret helper string, it also streamlines the key creation procedure. Even with cutting-edge quantum computing technology, the extractor should be immune to quantum cryptanalysis, making it appropriate for safe cryptographic key production. To ensure a dependable and secure technique for key extraction from biometric data, the study investigates the balance between False Rejection Rate (FRR) and False Acceptance Rate (FAR) in biometric key creation. The McEliece cryptosystem, named after its creator, Robert McEliece, uses error-correcting codes, specifically Goppa codes, for public-key encryption. The security of the McEliece cryptosystem relies on decoding random linear codes, which are impervious to attacks such as factoring or discrete logarithm-based attacks. The proposed fuzzy extractor offers a secure and efficient approach to encrypting and decrypting messages, as shown in Figure 2.

The McEliece cryptosystem uses the private key, consisting of the inverses of matrices S and P (denoted as S_inv and P_inv), along with the Goppa code used to generate the public key using Equation 6. To encrypt a message, the sender multiplies the message by the public key Gx, creating a codeword using Equation 7. Random errors are then introduced at t locations to enhance security. The encrypted codeword is sent to the recipient, who uses S_inv and P_inv to decode it. The recipient first unshuffles the cipher using P_inv using Equation 8, decodes it with the Goppa code, and finally unscrambles it with S_inv to using Equation 9 retrieve the original message. The security of McEliece comes from the difficulty of decoding random linear codes, which resists attacks such as factoring or discrete logarithm-based methods. The system's challenge lies in determining the generator matrix of a code from the code itself. As an

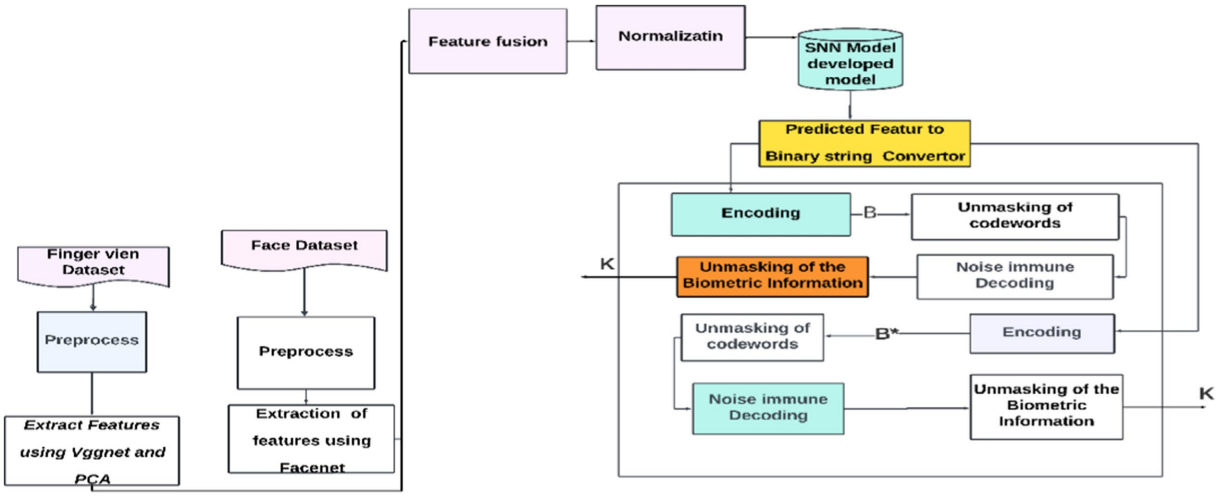

FIGURE 2
Developed SNN model with code-based cryptography.

efficient and secure approach to public-key encryption, the McEliece cryptosystem exemplifies robust key generation, encryption, and decryption processes.

Select a Goppa code using the given parameters (n, k, and t): n represents the length of the codeword, k represents the dimension of the code, and t represents the error-correcting capabilities of the code.

Public key k x n matrix Gx is generated by:

$$G_x = G_p.S.P \tag{6}$$

Encoding the biometric data using the public key and introducing t errors, I is k bit information of the biometric.

$$B_x^* = I.G_x + \mathrm{e} \tag{7}$$

Where $B_x = I.G_x$ is the codeword of masked code with a generator matrix G_p, I is the k-bit public text or biometric information, vector e is the secret error vector with hamming weight (number of non-zero positions) that equals wH(e) = t.

Decoding:

$$\vec{B}_x^* = B_x^*.P_{-inv} = I'.G_p + e' \tag{8}$$

In addition, decoded it to obtain I'.

$$I = I'.S_{-\mathrm{inv}} \tag{9}$$

then I is generated. Here, I is the password, which is expressed above as K.

3.7.1 Initial registration

This is the process where information (biometric information) is converted into a particular form or codeword using a public key known as encoding. Unmasking of codeword: This step involves revealing or interpreting the encoded data using the secret keys. Noise immune decoding: This implies that the codeword can be decoded in the presence of some noise or interference using the Goppa code.

3.7.2 Usage stage

Encoding: Once again, this represents the encoding of information using the public key of the data provided by users. Unmasking of information: Similar to the "Unmasking of codeword" in the first diagram, it might involve revealing the encoded information. Noise immune decoding: As before, it suggests that the information can be decoded notwithstanding noise interference.

Here, B and B^* belong to the same person as defined in Figure 1 and in our interpretation, $B = I.G_x + \mathrm{e}$ and $B^* = I.G_x + e^*$ (3). If e and e^* are distinct vectors with a Hamming weight lower than t, then decoding the vectors will result in the recovery of the identical vector I'. Once the vector is unmasked, the secret key K is generated.

3.8 Biometric extractor performance indicators

The False Rejection Rate (FRR) is calculated using Equation 10, which considers the probability of distortions in biometric features. False Rejection Rate (FRR) and False Acceptance Rate (FAR) are important metrics in biometric authentication. The FAR quantifies the possibility of an unauthorized person being erroneously granted access by the biometric system, whereas the False Rejection Rate (FRR) assesses the extent of inaccurate access denials for authorized users. To assess this probability, we shall examine two scenarios.

Let us assume that the biometric data scanning and processing produced a binary string $B^* = I.G_x + e^*$, where the hamming weight of an error vector e^* represents the potential differences between B^* and a reference biometric set B. The number of non-zero positions in a vector e^* is defined by the probability of a non-zero character occurring in e^*. This probability represents the likelihood of a character in the codeword $B_x = I.G_x$ being distorted for both authorized and

unauthorized users. However, these probabilities fluctuate between the two. Let us examine the initial scenario: Assume that the vector $B^* = I.G_x + e^*$ is owned by the authorized user. The probability of a single character distortion in B_x is denoted as P0. The formula can be used to estimate the value of FRR (Kuznetsov et al., 2022):

$$\mathrm{FRR} = 1 - \sum_{I=0}^{t} C_k^i p_0^i \left(1 - \mathrm{p_0}\right)^{k-\mathrm{i}} \tag{10}$$

The term $\sum$ represents the summation from $i = 0$ to t, where t is the maximum number of allowable errors or distortions in the biometric feature. C_k^i denotes the combination or binomial coefficient, which calculates the number of ways to choose i distortions out of k total distortions. p_0^i is the probability of having i distortions in the biometric feature. $\left(1 - p_0\right)^{k-\mathrm{i}}$ represents the probability of having k-i correct characters in the biometric feature. For each possible number of distortions (i) from 0 to t, the formula calculates the probability of that specific scenario occurring. The probabilities are then subtracted from 1 to get the overall False Rejection Rate.

Assume that the vector $B^* = I.G_x + e^*$ is possessed by an unauthorized user. The probability of a single character distortion is denoted as p1. The False Acceptance Rate (FAR) is determined using Equation 11, which evaluates the likelihood of unauthorized access. Subsequently, the value of FAR can be assessed in accordance with the prescribed formula (Kuznetsov et al., 2022):

$$\mathrm{FAR} = \sum_{I=0}^{t} C_k^i p_1^i \left(1 - \mathrm{p_1}\right)^{k-\mathrm{i}} \tag{11}$$

The term $\sum$ represents the summation from $i = 0$ to t, where t is the maximum number of allowable errors or distortions in the biometric feature. C_k^i denotes the combination or binomial coefficient, which calculates the number of ways to choose i distortions out of k total distortions. p_1^i denotes the likelihood of occurrence i distortions in the biometric feature. $\left(1 - p_1\right)^{k-\mathrm{i}}$ denotes the likelihood of occurrence k-i correct characters in the biometric feature. For each possible number of distortions (i) from 0 to t, the formula calculates the probability of that specific scenario occurring. The probabilities are then summed up to get the overall False Acceptance Rate.

4 Experimentation

This section includes the hyperparameters we employed, the model configurations created, the outcomes of experiments conducted during the training, the evaluation of the construction of the Siamese Neural Network (SNN), the Binary string converter, the Cryptography Code, and, lastly, an explanation of the summary discussion.

4.1 Extracted features

From the above Figures 3, 4, we extracted 128 features from each image in column format and saved them as CSV for later use to fuse the face features with the finger vein. From here, we saved the images with their labels, image paths, and features.

4.2 The averaged and normalized finger vein features

Before fusing face features, the averaged and normalized finger vein features for each class were calculated and saved as a separate CSV file, as shown in Figure 5. This process produces a representative feature vector that summarizes the attributes of samples within each class, reducing the dimensionality of the data and aiding in tasks such as visualization, grouping, and classification. Averaging features across multiple samples in a class minimizes noise and outliers, leading to more reliable and consistent class representations.

Specifically, for classes of faces and fingerprints defined as $n \times n$, the first row of the finger vein class, which represents the first-class average of a finger vein, is combined with the first five rows of face features associated with that individual. This process continues incrementally, with each subsequent row of finger vein being fused

id	image_pat	label	feature_0	feature_1	feature_2	feature_3	feature_4
1	C:\Users\u	0	-0.20163	0.065301	0.072926	-0.04735	-0.18514
2	C:\Users\u	0	-0.18024	0.097979	0.081345	-0.00799	-0.1573
3	C:\Users\u	0	-0.19446	0.042553	0.105301	-0.06998	-0.18223
4	C:\Users\u	0	-0.20945	0.054452	0.092095	-0.06021	-0.16264
5	C:\Users\u	0	-0.20631	0.064757	0.097405	-0.06571	-0.17031
6	C:\Users\u	1	-0.09437	0.020673	0.002396	-0.05208	-0.03435
7	C:\Users\u	1	-0.10158	-0.02546	0.054473	-0.01736	-0.0569
8	C:\Users\u	1	-0.04969	0.035006	0.017549	-0.05273	-0.04194
9	C:\Users\u	1	-0.06241	-0.02054	0.045278	-0.01368	-0.06353
10	C:\Users\u	1	-0.08219	0.06808	0.031285	-0.05004	-0.06452
11	C:\Users\u	2	-0.10069	0.098915	-0.01488	-0.09441	-0.15992

FIGURE 3
Sample of the extracted feature of face.

id	image_pat	label	feature_0	feature_1	feature_2	feature_3	feature_4
1	D:\Face_a	0	-187.494	3.878946	181.5851	-41.6585	37.91176
2	D:\Face_a	0	-184.66	-2.25568	134.7486	-20.0301	62.05299
3	D:\Face_a	0	-153.714	109.0982	103.0551	-161.113	47.11851
4	D:\Face_a	0	-156.543	106.2003	134.7735	-145.36	30.03239
5	D:\Face_a	0	-179.046	-87.6998	135.327	53.27776	11.5431
6	D:\Face_a	0	-181.44	-90.4385	129.4548	47.05991	18.01299
7	D:\Face_a	0	-219.039	71.55534	-2.43678	-112.134	-90.4779
8	D:\Face_a	0	-203.932	43.8863	43.09702	-113.289	-69.5364

FIGURE 4
Sample of finger vein features.

id	image_pat	label	feature_0	feature_1	feature_2	feature_3	feature_4
1	D:\Face_a	0	0.093783	0.297417	0.473098	0.355112	0.453035
19	D:\Face_a	1	0.119999	0.285973	0.483671	0.180488	0.575696
37	D:\Face_a	2	0.090892	0.330565	0.302924	0.167321	0.464865
55	D:\Face_a	3	0.076344	0.46669	0.434379	0.358541	0.414232
73	D:\Face_a	4	0.194615	0.383629	0.780723	0.577002	0.274465
91	D:\Face_a	5	0.063021	0.186627	0.45704	0.315507	0.361808
109	D:\Face_a	6	0.710163	0.509235	0.25908	0.320084	0.007714
127	D:\Face_a	7	0.744163	0.680915	0.769917	0.701139	0.076094
145	D:\Face_a	8	0.619662	0.627295	0.658871	1	0.301013
163	D:\Face_a	9	0.657794	0.94416	0.208731	0.51652	0.492209

FIGURE 5
Sample of normalized and averaged finger vein features.

id	image_pat	label	feature_0	feature_1	feature_2	feature_3	feature_4	feature_5
1	C:\Users\u	0	-0.10784	0.362718	0.546024	0.307767	0.26789	0.325142
2	C:\Users\u	0	-0.08645	0.395396	0.554443	0.347123	0.295738	0.392512
3	C:\Users\u	0	-0.10067	0.33997	0.578399	0.285132	0.270804	0.283723
4	C:\Users\u	0	-0.11567	0.351869	0.565192	0.294901	0.290396	0.315774
5	C:\Users\u	0	-0.11253	0.362174	0.570503	0.289406	0.282724	0.318014
6	C:\Users\u	1	0.025632	0.306646	0.486067	0.128411	0.541348	0.259285
7	C:\Users\u	1	0.018416	0.260517	0.538144	0.163133	0.518794	0.219072
8	C:\Users\u	1	0.070314	0.32098	0.50122	0.127762	0.533755	0.306998
9	C:\Users\u	1	0.057593	0.265433	0.528949	0.166809	0.512161	0.266088
10	C:\Users\u	1	0.037813	0.354053	0.514955	0.130453	0.511175	0.31386
11	C:\Users\u	2	-0.0098	0.429479	0.288042	0.072907	0.304944	0.519625

FIGURE 6
Sample feature of finger vein and face features of finger vein feature fused.

with the next five rows of facial features until all data are processed. This method ensures that each averaged finger vein feature contributes to the corresponding face features, facilitating a comprehensive representation in the fused dataset, as illustrated in Figure 6. The resulting combined value reflects an incremental addition of features, enhancing the overall input before model training.

4.3 Dataset splitting summary

The distinct train-validation-test splitting method used is an 80-10-10 split. Table 1 summarizes the overall dataset splitting for the experiment and task.

The dataset split for each trial, with distinct categories and their associated values, is detailed in Table 1. It provides an overview of the data from the training, validation, and test sets used to build the model. So, in total, we used 425 datasets. We used 320 data to train the model, 40 for validation, and 45 for testing.

4.4 Hyperparameters used

We implemented a Siamese Neural Network with hyperparameter tuning using Keras and Kerastuner. The model includes adjustable parameters, batch size, regularization, dropout rate, and hidden layer size, using a custom distance loss function to train on triplet data. The SiameseHyperModel class manages hyperparameter optimization with RandomSearch and generates training batches. After training, the best model and its hyperparameters are evaluated and summarized. The model's training process is configured based on 25 epochs, a batch size of 35, and a learning rate of 0.001. The 'hard number = 10' parameter helps the model handle challenging examples, while the Adam optimizer ensures efficient weight updates with adaptive learning rates. Validation steps monitor performance to prevent overfitting, with steps per epoch and steps per validation set to 15 and 2, respectively. These hyperparameters together influence the model's learning behavior, convergence, and generalization performance.

5 Experimental result

This article contains the results and conclusions from a number of tests we ran to assess how well our deep learning-based SNN model performed for the fusion features. We also presented the findings from the SNN model with the code-based cryptography key generation. Initially, we provided the SNN outcomes, which played a crucial role in developing our cryptographic key generation model. The performance of the SNN model was evaluated by computing loss training and validation.

TABLE 1 Dataset splitting.

Development	Training	Validation	Testing	Total
Exp	80%	10%	10%	1,500

5.1 Result of SNN model

The component of the SNN model evaluation's accuracy and loss is shown in this section. *Model Loss:* "Loss" measures the performance of a machine learning model by quantifying the difference between the predicted and actual outputs. It tracks the model's learning progress during training, with values showing improvement across epochs.

5.1.1 Accuracy

The accuracy of the model is measured using the similarity and dissimilarity of the features based on the label given, as shown in Table 2. If two images are similar, their difference is below the threshold value, and the images belong to the same person, but if the images are dissimilar, their difference is above the threshold, and they belong to different persons.

5.1.2 Results of FAR and FRR for the code-based cryptography

Table 2 exhibits the empirical calculation of the $\hat{\sigma}$ same, which is the average similarity of retrieved binary vectors for the same individual using multiple deep learning models. The provided data are utilized in the calculation of p_0, which is $p_0 = 1-\hat{o}$. Consequently, we determine that p_0 is equal to 0.07. These numbers are used to compute $p_0 = 1-\hat{\sigma}$ same (Kuznetsov et al., 2022). Similarly, we assess the value of p_1's based on the empirical results for $\hat{o}$ diff. We have deduced that the value $p_1 = 0.37$. These results provide significant novel insights into the performance of several deep learning models in generating binary vectors for biometric verification. They also indicate potential for additional progress and innovation in this field. The extraction approach relies on the use of code-based cryptosystems that employ a linear block code $(n, k, d) = (2^m, 2^m - mt, 2t+1)$ block $(n, k, d) = (2^m, 2^m - mt, 2t+1)$ with a fast-decoding process of polynomial complexity. The binary Goppa code with parameters $(n, k, d) = (2^m, 2^m - mt, 2t+1)$ for some m in Z^+ is thought to be the best alternative (Kuznetsov et al., 2022). A comprehensive analysis and comparison of the False Acceptance Rate (FAR) and False Rejection Rate (FRR) were conducted to evaluate the system's performance. The method being discussed relies on code-based cryptosystems that utilize a linear block (n, k, d) code that can be decoded rapidly with polynomial complexity. In our investigations, we generated binary strings of length $n = 128$ or $m = 7$. The parameters k and d of the Goppa codes for different t values are shown in Table 3. Table 3 also included the calculated values of FRR and FAR for estimations for different Goppa codes.

5.2 Comparative analysis of results

The outcomes of various other pertinent studies were compared with the study we conducted, as shown in Table 4. We provided a thorough comparison analysis of these findings below.

TABLE 2 Sigma similarity.

Threshold type	Sigma same	# of same pairs	Sigma diff	# of diff pairs	Sigma score
0.5 vector	tf.Tensor(0.92664933, shape = (), dtype = float32)	36	tf.Tensor(0.6359863, shape = (), dtype = float32)	32	0.613826
Expected value	tf.Tensor(0.9863281, shape = (), dtype = float32)	36	tf.Tensor(0.9250488, shape = (), dtype = float32)	32	0.388859

In comparison to previous research, our work demonstrates significant improvements in biometric key generation by achieving a lower False Rejection Rate (FRR) of less than 3.4% and a False Acceptance Rate (FAR) of less than 1%, outperforming (Kuznetsov et al., 2022) FRR of 8.3% and FAR of 7.4%. Additionally, while it generated 37-bit keys, our research produced up to 51, providing enhanced security, especially with post-quantum cryptosystems. Moreover, by utilizing bimodal biometrics, our approach improves accuracy and robustness over single-modal systems, as expressed in Table 4, addressing the limitations of relying on a single biometric modality. This shows not only technical improvement but also compassion for users by ensuring higher security and usability.

6 Discussion of results

This section covers the model configuration, experimental findings, and hyperparameters utilized in the SNN model and cryptography code.

The proposed system utilizes deep learning models, including FaceNet for face feature extraction and VGG19 for fingerprint feature extraction, combined with a Siamese Neural Network (SNN) for feature fusion. While the computational complexity of these models is significant, our study primarily focused on security and accuracy metrics rather than execution time. Future work will include detailed performance analysis, measuring training and inference speed across different hardware platforms to ensure the system's feasibility for real-time biometric authentication applications.

A dataset-splitting summary is shown in Table 1. Figure 6 lists the hyperparameters that were used: epoch, batch size, dropout, optimizer, activation function, length of the sequence, embedding dimension, loss, and train-test-split. One way to summarize the explanation of the experimental findings is as follows: Dataset Splitting: For the train and test SNN model, the dataset was divided into training, validation, and testing sets. Table 1 provides an overview of the data assigned to each set.

TABLE 3 FRR and FAR estimations for different Goppa codes.

t	*K*	*d*	SNN model	
			FAR	FRR
1	121	3	1.5754351486329734e-24	0.9990170515143107
5	79	15	4.100625196054598e-19	0.8905780902806273
10	65	19	2.6653196034529995e-14	0.28469187709689847
14	37	27	2.6218530171918587e-11	0.03455211265419574
17	9	35	2.0027487838478667e-09	0.003629338457096099

6.1 Model loss

Figure 7 shows that as the epoch increases, the model performs better, as can be seen by the decreasing value of the loss measure. The image depicts a graph illustrating the "Model loss" progression throughout the training of a machine-learning model across many epochs. An epoch in machine learning refers to a single iteration over the entire training dataset. The graph illustrates two lines, with one line depicting the loss on the training set (in blue) and the other line indicating the loss on the validation set (in red). Each line in the graph demonstrates a decrease in loss as the number of epochs increases, suggesting that the model is effectively acquiring knowledge from the data. The x-axis is labeled "Epoch" and shows the progression of epochs from 1 to 25. The y-axis is labeled "Loss" and measures the magnitude of loss, with values from 0 to approximately 0.35.

6.2 Accuracy

This is calculated by computing a sigma score based on two values. The method is computing the average similarity in a Siamese network between photos of the same class (i.e., similar pairs) and images of different classes (i.e., dissimilar pairings). A formula cubes the difference between the average similarity of similar and different pairings (sigma_same and sigma_diff), multiplies it by sigma_same, and caps the result at zero to assess how well the network separated these pairs. The score measures the network's ability to discriminate between similar and dissimilar pairings; higher values correspond to better performance. Class grouping of feature vectors (embeddings) and associated labels facilitates comparisons both within and across classes. The first step of the technique is to classify the data points into batches and then compute the average similarity (sigma_same) for comparable pairings within each batch. After converting the embeddings, a similarity function computes the similarity scores,

TABLE 4 A comparative analysis of the results.

Source	Methods and technologies	FRR	FAR	Additional Comments
Wang et al. (2021)	A secure biometric key generation mechanism via deep learning and its application	GAR = 98.47 and EER = 1.09 at fixed 1% FAR		During new enrollment, retrain the network to learn the mapping b\n new biometric image and binary code
Jana et al. (2022)	Neural fuzzy extractors for biometric authentication	2.5%–4.4%	2.5%–4.4%	Advances in iris-based biometric authentication have been made; however, the study of facial biometrics has not been expanded
Kuznetsov et al. (2022)	Code-based cryptographic extractor using Keras FaceNet face recognition	8.3%	7.4%	For a produced key length of 37 bits with a helper string. Furthermore, the code-based cryptosystems on which our extractor is built offer post-quantum level security
Our research	Cryptographic key generation via deep learning using Bi-modal biometric face and fingerprint	<3.4% for t between 14 and 17	<1%	For producing a key length of 128 bits using bimodal. Furthermore, the code-based cryptosystems on which our extractor is built offer post-quantum level security

FIGURE 7
Model loss.

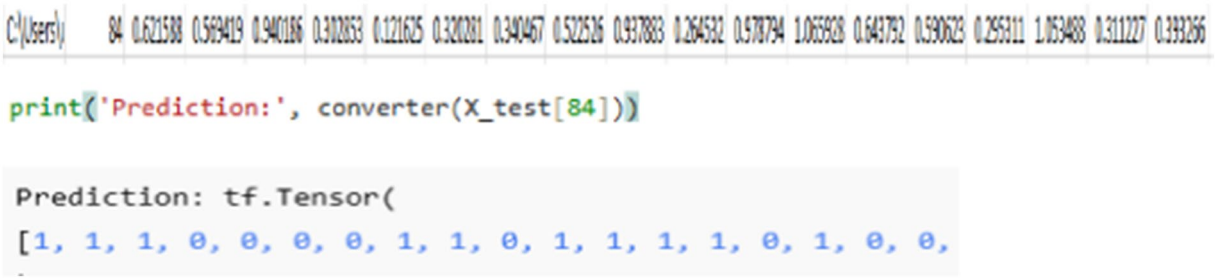

FIGURE 8
Binary string output.

which are then averaged and added together. To ensure an equal amount of comparisons from both classes, data points for dissimilar pairings are compared across batches. These different pairs' similarity scores (sigma_diff) are computed and averaged. The average values of sigma_same and sigma_diff are then used to compute the sigma score, which gives an overall indication of how well the network can distinguish between similar and different embeddings. Overall, this accuracy estimator provides a way to evaluate the performance of a Siamese network by calculating average similarities for similar and dissimilar pairs based on a chosen converter function. It also computes a sigma score to quantify how well the network separates the two categories. From Table 2, based on the 0.5 threshold, the output is given. The average sigma same similarity and sigma difference similarity based on the above definition using a 0.5 threshold is 0.93 and 0.64, respectively.

Presently, FRR ≈ 25% and FAR ≈ 10% are considered acceptable markers of biometric identification based on a person's facial picture. Meanwhile, facial recognition technology has advanced significantly in recent years. As per the National Institute of Standards and Technology (NIST) research (Libby and Libby, 2021), the optimal face recognition method exhibits an error rate of approximately 0.08% under ideal conditions. Furthermore, it is imperative to minimize the FAR values for biometric password generation systems as much as possible, ideally aligning them with the probability of password guessing. The situation in which the False Rejection Rate (FRR) is equal to or less than 10% is depicted in Table 3. The values shown are within the required range (<10%), with t varying from 10 to 17. The comparative analysis in Table 4 highlights the evolution of biometric key generation mechanisms, showing significant advancements in accuracy, security, and adaptability over time. Early methods had higher error rates, while more recent approaches, like our research, demonstrate much lower FRR and FAR rates, achieving less than 3.4 and 1%, respectively, for t between 10 and 17, with a strong up to 51-bit key length. Our biometric cryptographic key generation system ensures strong error tolerance and stability using Goppa codes, which correct variations in biometric data. Multimodal fusion significantly reduces spoofing risks by requiring multiple biometric traits for authentication. Unlike Rivest Shamir Adleman (RSA) and Elliptic Curve Cryptography (ECC), the McEliece cryptosystem provides post-quantum security, making it resistant to future quantum attacks. Compared to previous works, our method achieves a lower FRR (<3.4%) and FAR (<1%), significantly improving from previous FRR (8.3%) and FAR (7.4%). Additionally, our system generates 128-bit+ cryptographic keys, surpassing the 37-bit keys of earlier approaches, ensuring higher security and robustness.

Binary string output of the model: By feeding the model with a testing dataset from class 84, which includes the feature values shown in Figure 8, the model generates a binary string output. Using a binary string converter, the predicted values are transformed into a binary representation. The model's prediction closely matches the expected output, demonstrating its accuracy in classification and conversion.

7 Conclusion and future work

This research demonstrates an advanced approach to cryptographic key generation by integrating deep learning with multimodal biometric data from face and finger vein modalities. Leveraging pretrained models such as FaceNet and VGG19, along with a Siamese Neural Network

(SNN) and fuzzy extractors, the system achieves high accuracy and robustness, with a False Acceptance Rate (FAR) below 1% and a False Rejection Rate (FRR) below 3.4%. Incorporating Goppa code-based cryptographic systems ensures post-quantum security, making the approach highly resilient. Future work will focus on optimizing neural network architectures, integrating additional biometric modalities, and exploring quantum-resistant algorithms to address emerging challenges. Real-time implementation, enhanced scalability, and adaptability to diverse user environments will also be prioritized to improve usability and robustness. Future work will focus on testing with larger and more diverse biometric datasets to further validate the robustness of the approach. Additionally, efforts will be directed toward optimizing neural network architectures, integrating additional biometric modalities, and exploring quantum-resistant algorithms. Real-time implementation, enhanced scalability, and adaptability to diverse user environments will also be prioritized to improve usability and robustness. While the computational complexity of these models is significant, our study primarily focused on security and accuracy metrics rather than execution time. Future work will include detailed performance analysis, measuring training and inference speed across different hardware platforms to ensure the system's feasibility for real-time biometric authentication applications.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Ethics statement

Ethical approval was not required for the studies involving humans because ethical approval was not required for this study as it utilized publicly available datasets that do not contain sensitive or personally identifiable information.

Author contributions

TGY: Writing – original draft, Writing – review & editing, Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization. HGY: Conceptualization, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – review & editing. EA: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

Supplementary material

The Supplementary material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2025.1545946/full#supplementary-material

References

Akintoye, K. A., and Akinwamide, S. (2024). Enhancing cell phone security through finger vein biometric authentication systems. *World J. Adv. Eng. Technol. Sci.* 13, 034–039. doi: 10.30574/wjaets.2024.13.1.0368

Babu, G. P., Aswal, U. S., Sindhu, M., Upadhye, V. J., Natrayan, L., and Patil, H. (2024). "Enhancing security with machine learning-based finger-vein biometric authentication system" in 2024 5th international conference on Mobile computing and sustainable informatics (ICMCSI) (Piscataway, NJ, USA: IEEE), 797–802. doi: 10.1109/ICMCSI61536.2024.00123

Bele, P. S. B. (2024). The role of biometric and authentication in security. *Int. J. Res. Appl. Sci. Eng. Technol.* 12, 1997–1999. doi: 10.22214/ijraset.2024.65520

De Oliveira Nunes, I., Rindal, P., and Shirvanian, M. (2024). Oblivious extractors and improved security in biometric-based authentication systems. *Lecture Notes Comput. Sci.* 14344, 290–312. doi: 10.1007/978-3-031-50594-2_15

Jana, A., Paudel, B., Sarker, M. K., Ebrahimi, M., Hitzler, P., and Amariucai, G. T. (2022). Neural fuzzy extractors: a secure way to use artificial neural networks for biometric user authentication. *Proc. Priv. Enhancing Technol.* 2022, 86–104. doi: 10.56553/popets-2022-0100

Kavitha, G., Prasannakumar, V., and Pranav, R. P. (2024). "Enhancing digital security: a comprehensive multi-model authentication framework leveraging cryptography and biometrics" in 2024 8th international conference on inventive systems and control (ICISC) (Singapore: Springer), 476–486. doi: 10.1007/978-981-99-6586-1_32

Kuznetsov, A., Kiyan, A., Uvarova, A., Serhiienko, R., and Smirnov, V. (2018). "New code based fuzzy extractor for biometric cryptography" in International scientific-practical conference problems of Infocommunications. Science and technology (PIC S&T), vol. *2018* (Piscataway, NJ, USA: IEEE), 119–124. doi: 10.1109/INFOCOMMST.2018.8632040

Kuznetsov, O., Zakharov, D., and Frontoni, E. (2023). Deep learning-based biometric cryptographic key generation with post-quantum security. *Multimed. Tools Appl.* 83, 56909–56938. doi: 10.1007/s11042-023-17714-7

Kuznetsov, A., Zakharov, D., Frontoni, E., Romeo, L., and Rosati, R. (2022). "Deep learning based fuzzy extractor for generating strong keys from biometric face images"

in 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 421–426.

Libby, C., and Libby, C. (2021). Facial recognition technology in 2021: masks, bias, and the future of healthcare. *J. Med. Syst.* 45:39. doi: 10.1007/s10916-021-01723-w

Raja, C. L. (2024). Digital transformation and data protection. *Int. J. Sci. Res. Eng. Manag.* 8, 1–5. doi: 10.55041/IJSREM37987

Shamili Shanmugapriya, S., Anila, S. V., Hemalatha, D., Karthika, K., Sankaradass, V., and Singh, S. (2024). "Optimizing security posture via synergized fingerprint and facial recognition in integrated AuthenticationFramework" in 2024 international conference on advances in data engineering and intelligent computing systems (ADICS) (Heidelberg, Germany: IEEE), 1–5. doi: 10.58532/v3becs8p1ch4

Shevchenko, V. V., and Anikin, I. V. (2023). "Generation of cryptographic keys and Person's verification based on face biometric" in 2023 international conference on industrial engineering, applications and manufacturing (ICIEAM) (Ternopil, Ukraine: IEEE), 815–819. doi: 10.35774/econa

Sulavko, A., Panfilova, I., Inivatov, D., Lozhnikov, P., Vulfin, A., and Samotuga, A. (2025). Biometric-based key generation and user authentication using voice password images and neural fuzzy extractor. *Appl. Syst. Innov.* 8:13. doi: 10.3390/asi8010013

Sydor, A., Balazh, D., Vitrovyi, Y., Kapshii, O., Karpin, O., and Maksymyuk, T. (2024). Research on the state-of-the-art deep learning based models for face detection and recognition. *Inf. Commun. Technol. Electron. Eng.* 4, 49–59. doi: 10.23939/ictee2024.02.049

Wang, Y., Li, B., Zhang, Y., Wu, J., and Ma, Q. (2021). A secure biometric key generation mechanism via deep learning and its application. *Appl. Sci.* 11:8497. doi: 10.3390/app11188497

Wang, Y., Shi, D., and Zhou, W. (2022). Convolutional neural network approach based on multimodal biometric system with fusion of face and finger vein features. *Sensors* 22, 1–15. doi: 10.3390/s22166039

Yao, Q., Song, D., Xu, X., and Zou, K. (2024). Visual feature-guided diamond convolutional network for finger vein recognition. *Sensors* 24:6097. doi: 10.3390/s24186097