Biometric Authentication of Computer Users Using Facial Images in Protected Execution Mode

Alexey Sulavko *, Irina Panfilova, Alexey Vulfin *, Pavel Lozhnikov, Alexander Samotuga *

Posted Date: 13 January 2026

doi: 10.20944/preprints202601.0869.v1

Keywords: biometric template protection; automatic machine learning; facial biometrics; artificial intelligence in protected execution mode; feature extraction; verification of biometric images

 doi:10.20944/preprints202601.0869.v1

Article

Biometric Authentication of Computer Users Using Facial Images in Protected Execution Mode

Alexey Sulavko [1,*], Irina Panfilova [1], Alexey Vulfin [2,*], Pavel Lozhnikov [1] and Alexander Samotuga [1,*]

[1] Department of Comprehensive information protection, Omsk State Technical University, Mira, h. 11, Omsk, 644050, Omsk region, Russian Federation
[2] Department of Computer Science and Information Security,Ufa University of Science and Technology
* Correspondence: vulfin.am@ugatu.su (A.V.), sulavich@mail.ru (A.S.); samotugasashok@mail.ru (A.S.); Tel.: +79-5339-490-54 (A.S.))

Abstract

The goal of the present work is to increase the reliability of facial biometric-based key generation, which is used for remote authentication with the protection of biometric templates. A neuro-extractor model was developed to associate a feature vector with a cryptographic key or user password. The model is a shallow neural network based on partially connected trigonometric neurons. The model is trained to generate a cryptographic key or password at the output when a feature vector of a legitimate user is received at the input and to generate a random code when a feature vector of an unknown subject is received at the input. The proposed neuron model is based on the use of trigonometric measures of proximity. This approach can significantly improve the accuracy of biometric image classification and the length of cryptographic keys associated with a biometric image, while ensuring the confidentiality of biometric templates. An algorithm for the synthesis and automatic learning of neuro-extractor was proposed. The experimental evaluation showed next results: $EER \approx 0.6\%$ and $EER \approx 1.2\%$ on the open datasets Faces94 and LFW. The trigonometric neuron network is capable of producing a 2048-bit key, which is a higher reliability indicator compared to previously achieved results. The model can be used not only in facial biometric applications, but also in authentication by other modalities, as well as in building a secure mode for executing artificial intelligence (AI) algorithms. The secure mode is an important component of the trusted AI concept. This mode complicates the analysis of operations performed by AI, unauthorized control of AI, and the extraction of its knowledge by unauthorized persons. The trigonometric neuron network has increased resistance to such destructive effects.

Keywords: biometric template protection; automatic machine learning; facial biometrics; artificial intelligence in protected execution mode; feature extraction; verification of biometric images

1. Introduction

In the world of digital communications, it is very important to confirm the authenticity of a remote partner. Today, artificial intelligence (AI) replaces humans in many ways. For example, such replaces are successful in, in conversation, correspondence, and sometimes in making important decisions and management tasks. So the issues of trust when using AI come to the forefront.

The reliability of artificial intelligence (AI) applications is largely determined by what data it was trained on and whether the size of the training set is sufficient. Any unauthorized interference in the work of AI may lead to consequences (material damage, threat to life, health, technological failure, disaster, etc.), depending on the purpose and capabilities of a specific AI implementation. Therefore, in particularly important tasks, a mechanism is needed to make sure that attackers could not somehow disrupt the logic of AI (retrain, substitute, etc.).

How can you be sure that a particular AI instance can actually be trusted? How can you be sure that an attacker was not able to retrain, replace the AI model, or otherwise influence the logic of its

 doi:10.20944/preprints202601.0869.v1

operation? An authentication mechanism that indirectly uses the knowledge of a trusted AI (neural network weighting parameters) as an authenticator is needed. This is implemented using a special AI architecture that allows you to execute neural network algorithms in secure mode.

Protected mode makes it difficult for any unauthorized person to implement the following scenarios: analyzing the operations performed by AI (to understand the essence of the transformations); AI control (by changing the operating algorithm, replacing AI data, adversarial attacks, etc.); AI knowledge extraction and interpretation. The concept of secure AI execution should be considered an integral element of the concept of trusted AI, along with explainability, robustness, etc.

Protected AI execution mode must be used in critical applications related to medicine, finance, and information security. The most striking example of the application of this concept is the biometric-based key generation for users authentication models, based on automatic training of artificial neural networks [1,2]. This example is interesting because such models make it possible to confirm the authenticity of not only the user, but also the AI instance underlying the biometric system (the user can also be sure that using the system is safe and that his biometric template is reliably protected). It should be noted separately that automatic machine learning is a process of training models that is conducted without the direct participation of a machine learning engineer. So, this specialist does not need to control the stages of training, determine the moment of its stop, or manually adjust the model parameters.

Protected AI execution mode must be used in critical applications related to medicine, finance, and information security. The most striking example of the application of this concept is the biometric-based key generation for users authentication models, based on automatic training of artificial neural networks. This example is interesting because such models make it possible to confirm the authenticity of not only the user, but also the AI instance underlying the biometric system (the user can also be sure that using the system is safe and that his biometric template is reliably protected).

This study is devoted to the creation of a new biometric-based key generation model for users facial authentication based on trigonometric neurons (t-neuro-extractor), first described in this work. The proposed model makes it possible to increase the accuracy of classification of biometric images, as well as the length of cryptographic keys or passwords associated with them, compared to previously existing biometric-based key generation models. Moreover, the proposed model is not vulnerable to known attacks on pre-existing models.

2. Analysis of Previously Achieved Results

According to recent research [3], most existing attacks on biometric systems, including those implemented based on machine learning methods, can be divided into two large-scale groups. First group consist of attacks on biometric presentation [4] (spoofing attacks). Second group include attacks on the structural components of the authentication algorithm in order to extract knowledge or data from them.

This paper discusses technologies for protecting biometric systems aimed at solving the problems of the second group of attacks. These technologies are divided into three categories: encrypted biometrics, cancelable biometrics and biometric cryptosystems.

The most promising approach to encrypting biometric templates today is considered to be homomorphic encryption (HE). Mentioned approach can be applied to features extracted from facial images using pre-trained deep neural network models [5]. Despite the obvious advantages, the computational complexity of the algorithm does not make it widely applicable in problems of protecting biometric templates. In addition, in many ways the algorithm is subject to the problem of error accumulation. For example, after performing many operations on encrypted data, the result may no longer match the result of those operations on unencrypted data. Current research in this area typically focuses on finding a balance between speeding up HE operations while minimizing losses in the resulting recognition accuracy [6].

 doi:10.20944/preprints202601.0869.v1

Cancellable biometrics consists of creating a unique template using reversible (e.g., image block randomization) or irreversible transformations (e.g., hashing) based on the user's biometric data. This approach assumes ability to revoke and change this template if compromised [7]. If a biometric template is compromised (for example, if the template is stolen), the system can generate a new template based on the same biometric data, but with a new set of transformations. Thus, it is possible to generate different templates while maintaining the same biometric data. Cancellable biometrics has certain limitations. In particular, the template may no longer be secure if the supporting data is compromised when salted [7].

An alternative approach to solving the problems of confidentiality of biometric templates, combining the advantages of biometrics and cryptography, is biometric cryptosystems [8]. Biometric cryptosystems can be considered a special case of cancelable biometrics, however, biometric cryptosystems do not always provide for the ability to easily replace templates if they are compromised. As part of ensuring the security of biometric data, biometric cryptosystems additionally solves specific problems of managing cryptographic keys (generating keys with the required length and entropy, as well as ensuring the security of key storage).

One of the central tasks when designing biometric cryptosystems is to produce a key of the maximum possible length, taking into account the requirements of cryptography. In this regard, an important concept is entropy, which is understood as the degree of randomness of the key generated at the output of biometric cryptosystems in the event of an attempt to identify/authenticate by an illegitimate user.

One of the basic concepts for building biometric cryptosystems can be considered the concept of a neural fuzzy extractor (or a neural network "biometrics-code" converter [9]). This concept constitutes a "black box" built on the basis of an artificial neural network, "knowing" its owner and reliably storing it password or cryptographic key. Neural fuzzy extractor is trained to generate and give the user his password (key) upon presentation of the corresponding biometric image "Genuine" (legitimate image). When presented with an image of any other subject or adversarial image (the "Impostor" image), the fuzzy neural extractor should generate a random binary code with high entropy. This property allows protection against adversarial attacks. In practice, there may be requirements for the length and information entropy of the key.

The specificity of constructing a fuzzy neural extractor is the use of a shallow neural network, which is built individually for each user, while a neural network is formed. The number of inputs of shallow neural network is equal to the number of features of the biometric image, and the number of outputs is the length of his personal key. Each neuron in the last layer generates one or more bits. The first implementation of a neural fuzzy extractor that was brought to real practice formed the basis of a series of Russian standards GOST R 52633 [10]. These standards are based on the shallow neural network automatic learning algorithm with one or two layers. Due to the small number of layers, such networks do not have very high generalization ability and are able to work only with pre-processed data - a feature vector with a normal distribution of values. The disadvantage of the mentioned scheme is its susceptibility to some specific types of attacks [11,12]. Other disadvantages include insufficient key length for a number of applications, or not very high classification accuracy compared to advanced machine learning methods.

Among the most significant modifications of the classical neural fuzzy extractor design, we note a neural fuzzy extractor based on correlation [9] neurons (c-neuro-extractor). This model works with pairs of features (instead of the original feature vector of the biometric image) and analyzes the degree of their correlation (positively correlated, independent or negatively correlated). This approach allows you to effectively work with biometric modalities characterized by strong internal correlation of features and create a neural fuzzy extractor whose parameters do not compromise the training set and cryptographic keys. However, correlation neurons are difficult to use if the user's biometric image (biometric template) does not have a sufficient number of pairs of highly correlated features. This

 doi:10.20944/preprints202601.0869.v1

feature is characteristic of facial images, which leads to the need to create an alternative approach to biometric facial authentication (as well as many other modalities).

Creating a neural fuzzy extractor based on a deep neural network trained using gradient descent optimization algorithms is still a difficult task. This is due to the fact that such algorithms are difficult to automate (the system must be trained on user data without the participation of an engineer). Thus, in [13], an attempt was made to create a neural fuzzy extractor for facial biometrics based on the backpropagation method. The authors note another important drawback of such implementations: the need to completely retrain the entire network when registering a new user. Similar difficulties are typical for the solution presented in [14] . A common disadvantage of such systems is low robustness.

An alternative to the neural fuzzy extractor is fuzzy extractor [15] (depending on the features of the biometric template protection scheme, other names can be found - fuzzy commitment, fuzzy vault, etc.). Such schemes (as the fuzzy extractor) play an important role in biometric cryptography because of their ability to handle biometric data that may be subject to noise and variability. Thus, in [16], the authors combine an improved fuzzy vault scheme with a deep neural network. This approach allows them to extract informative features from biometric images, while achieving a low level of type I error ($FMR < 0.01\%$) and an acceptable level of type II error ($FNMR < 1\%$). In addition, they conducted a comprehensive evaluation of different quantization and binarization methods, which confirmed their effectiveness under the FERET and FRGCv2 cross-database conditions.

Modern research shows that feature extraction using deep neural networks can minimize the risk of data compromise and improve data privacy. For example, the authors of [1] developed a system based on deep convolutional neural networks, while Pandey et al. [2] implemented a deep secure coding scheme that provides resistance to attacks on biometric templates. Mentioned approaches show how neural networks can improve the robustness of security systems.

The study [17] proposes a BTP scheme based on fuzzy commitment, which uses Euclidean distance as a metric to calculate the match score of biometric templates. Kuznetsov and et al. [18] presented a solution that combines deep neural networks with cryptographic algorithms. Using such methods not only improves security but also maintains high performance of biometric systems.

One of the interesting projects is the cryptosystem for face identification proposed in [19]. This cryptosystem includes two subsystems: One-to-many search using IoM hashing and a chaff-less fuzzy vault for uniform mixing genuine and "extra" sets. Despite the high level of identification accuracy, the system faces limitations related to its dependence on facial feature recognition and parameter settings.

Classical fuzzy extractor models cannot correct sufficient number of errors when releasing (generating) a key without significant loss of information because of the redundancy of error syndromes of the correcting codes. They are based on the use of error-correcting codes. The higher the correcting ability of the code, the more erroneous bits of quantized biometrics they can correct, and the shorter the length of the key associated with the biometrics. To increase the length of the key, it is necessary to use more biometric information (more features), but the more features are used, the less informative these additional features are (the most stable and informative features containing less noise are selected first), respectively, the more errors occur. Increasing the number of features leads to new errors, etc. In addition, fuzzy extractors do not take into account the distribution parameters of features, unlike the neural fuzzy extractor.

Research [20] shows that fuzzy extractor schemes are usually inferior to neural fuzzy extractors both in terms of cryptographic key length and recognition accuracy. It is also important to note that feature extraction using deep neural network architectures is not the same as the process of cryptographic key generation. Thus, in the study [21], the authors transform feature vectors into a binary code (key). However, as practice shows, in this case, the outputs of the feature extractor will always be correlated with the input biometric images, which can violate the requirements for the entropy of cryptographic keys. Feature extraction can act solely as a preliminary stage in biometric image processing.

 doi:10.20944/preprints202601.0869.v1

However, a lot of modern works are aimed at improving the reliability of fuzzy extractor schemes. Thus, in [22], an approach based on generated intervals and a two-level error correction technique was proposed, which significantly increased the noise resistance and reliability of the generated keys. Talreja et al. [23] developed a zero-shot method combining hashing and error correction, and researchers in [24] implemented LDPC codes trained on deep neural networks, providing a low error rate and high data reliability.

It should be noted that an important contribution to the development of biometric cryptography was the work of Lai et al. [25], who proposed a symmetric encryption scheme Keyring for biometric cryptosystems. This approach provided additional data protection because of symmetric encryption and became a significant step towards increasing the resistance of biometric systems to attacks. Also significant is the SecureFace system developed by Mai et al. [26]. This approach is aimed at ensuring the protection of face templates by using advanced data processing and protection algorithms. SecureFace demonstrates high efficiency when working with various biometric data and ensures the minimization of vulnerabilities of modern biometric systems.

3. Proposed Approach

In this study, it is proposed to separate a neural network model for face detection (face detector) and a neural network model for feature extraction (feature extractor) into separate independent blocks, separating them from the fuzzy neural extractor (Figure 1). The feature extractor makes it possible to obtain normally distributed features that unambiguously describe a biometric image from the point of view of their belonging to a subject (class). The feature extractor does not store the personal biometric data of registered users in the form of knowledge and must be trained on a set of anonymized image examples to ensure differential privacy. If a new user is added to the system or removed, additional training of this block is not required.

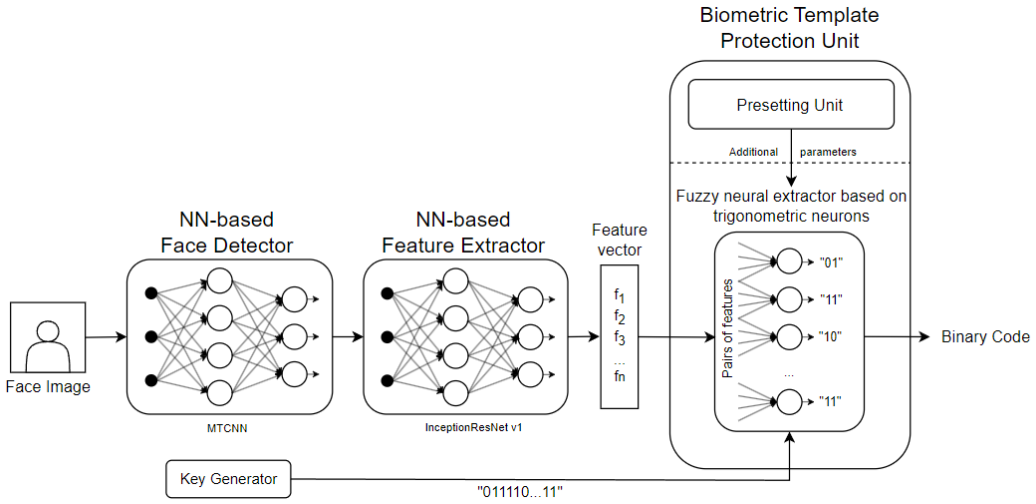


Figure 1. Block diagram of building a secure user authentication system by face

The first two blocks are built on the basis of convolutional neural networks (CNN) and first allow you to detect a face in the input image by cropping it along the contour, and then extract an n-dimensional feature vector (embedding). The feature vector $\overline{a} = (a_1, a_2, ..., a_n)$ is an input to a fuzzy neural extractor, based on the model of trigonometric neurons (t-neuro-extractor) proposed in this work, which classifies biometric images of users in a secure mode. It allows you to associate an externally specified cryptographic key (password) with the responses of its outputs, with each individual trigonometric neuron producing two key bits (unlike the classic fuzzy neural extractor model).

 doi:10.20944/preprints202601.0869.v1

3.1. Face Detection and Feature Extraction

The face detector and feature extractor are separate CNNs trained for face detection and feature extraction, respectively. For face detection, a multi-task cascade convolutional neural network MTCNN [27] was used, consisting of three subnetworks - P-Net, R-Net and O-Net. The network is trained to detect faces, and returns a tensor from the coordinates of the face's bounding box and key facial landmarks. To create a feature extractor, the transfer learning method [28] was used. This approach allows you to use the knowledge and experience gained in solving one problem to improve the efficiency of solving other related problems. The feature extractor is based on a pre-trained neural network Inception-ResNet v1 [29] (based on the ResNet architecture), originally proposed for the classification task, from which the last fully connected layers were excluded. After modification, the network allowed the extraction of vectors from 512 features. MTCNN and Inception-ResNet v1 are pre-trained on large-scale datasets (WIDER FACE and VGGFace2 [30]) and are part of the facenet-pytorch which is a Python open source library [31].

3.2. T-Neuro-Extractor

3.2.1. Mathematical Model of a Trigonometric Neuron

In tasks of biometric identification and authentication, the $\bar{a} = (a_1, a_2, ..., a_n)$ feature vector , arriving at the system input, which describes the biometric image of the subject, is compared with some reference vector ', which uniquely characterizes this user. However, by analyzing the relationships between features, including correlations, it is possible to extract additional information about the differences or similarities of the compared images [9]. In this case, any pair of features (a_i, a_j) can be represented in a separate feature subspace and studied for functional dependence (Figure 2).

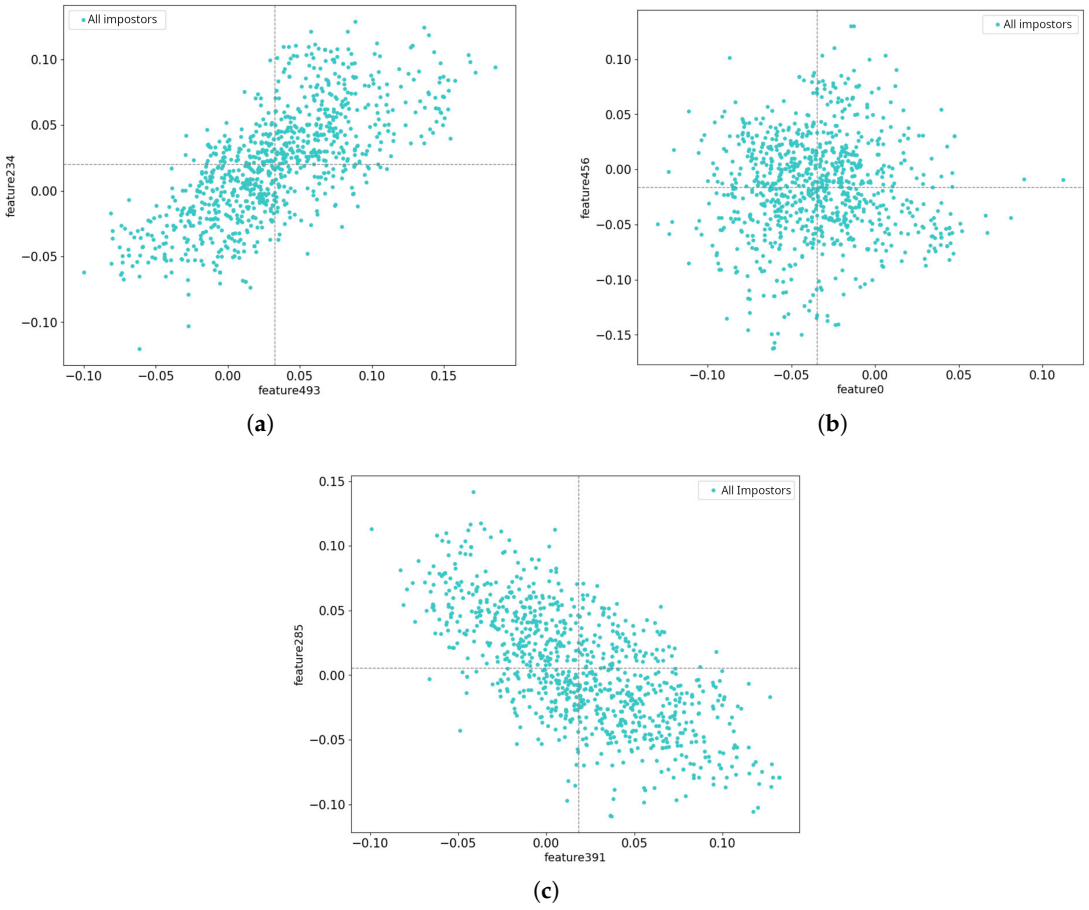


Figure 2. Subspaces of random pairs of features of the training set with different correlations of features: a) positive, b) weakly expressed, c) negative.

 doi:10.20944/preprints202601.0869.v1

Depending on the presence or absence of correlations between features, the "cloud" of points describing images in the subspace of a pair of features will be distributed from the lower left edge to the upper right edge (in the case of a positive correlation) or from the lower right to the upper left edge (in the case of a negative correlation). The coordinates of the center of the own region of the "Impostors" class ("center of mass") are equal to the average values of the corresponding features $O = (m_i, m_j)$ in Figure 2, dotted lines pass through the "center of mass".

Any pair of features can potentially be converted into a meta-feature [9], and the space of the original features can be converted into a rectifying space of meta-features using the mapping:

$$a_l^{'} = f(a_i, a_j),$$

where i and j are the feature numbers of the original feature vector $(i \neq j)$, $a_l^{'}$ is a meta-feature, i.e. a feature obtained by synthesizing two or more original features using a functional transformation f, l - meta-feature number. The number of possible subspaces of feature pairs and, accordingly, the number of meta-features is determined by formula (1) [9]:

$$n^{'} = 0.5(n(n-1)), \tag{1}$$

where n is the initial number of features.

Such a functional transformation f must meet at least two criteria:

1. Do not contain characteristics that compromise the image of a legitimate user ("Genuine"). This criterion is fundamental for building a secure biometric authentication system.
2. Allow to describe with sufficiently high accuracy the location of the image in the rectifying hyperspace of meta-features, taking into account the high variability of biometric images.

Let us take the Euclidean metric as an estimate of the distance from the point $O = (m_i, m_j)$ to some biometric image:

$$d(a, m) = \sqrt{(a_i - m_i)^2 + (a_j - m_j)^2},$$

As can be seen from Figure 3, the distances from the "center of mass" to the biometric images of different d_1 and d_2 subjects will be equal. Each of these images belongs to different classes. Some classes of images are located quite close to each other (images of one class are orange triangles, images of another are blue triangles in Figure 3). In this regard, using the Euclidean metric (or another similar one) it will be problematic to correlate images with their classes (draw a separating hyperplane).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 January 2026 doi:10.20944/preprints202601.0869.v1

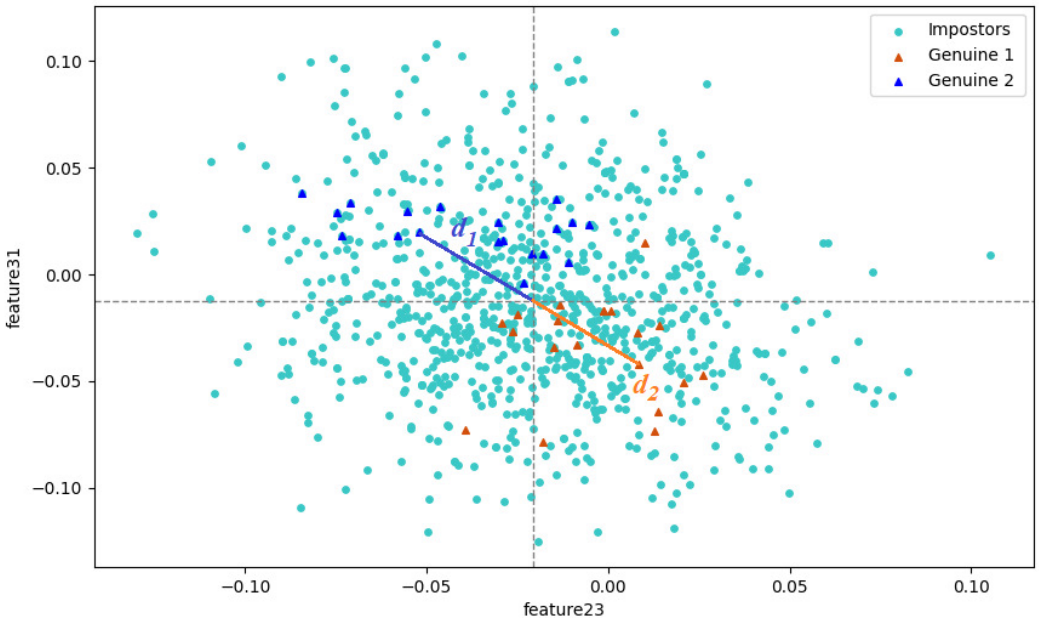


Figure 3. Euclidean distances from the center of the own region "Impostors" to an image of a certain class (d_2) and a random image (d_1)

Two proximity measures (2) and (3) of images in the subspace of pairs of features are proposed, taking into account the angle between the $\overline{d}$ vector (the segment between the "center of mass" $O = (m_i, m_j)$ and a specific biometric image), and the $\overline{v}$ vector, equal to length $\overline{d}$ vector but coinciding in direction with the OX axis (Figure 4).

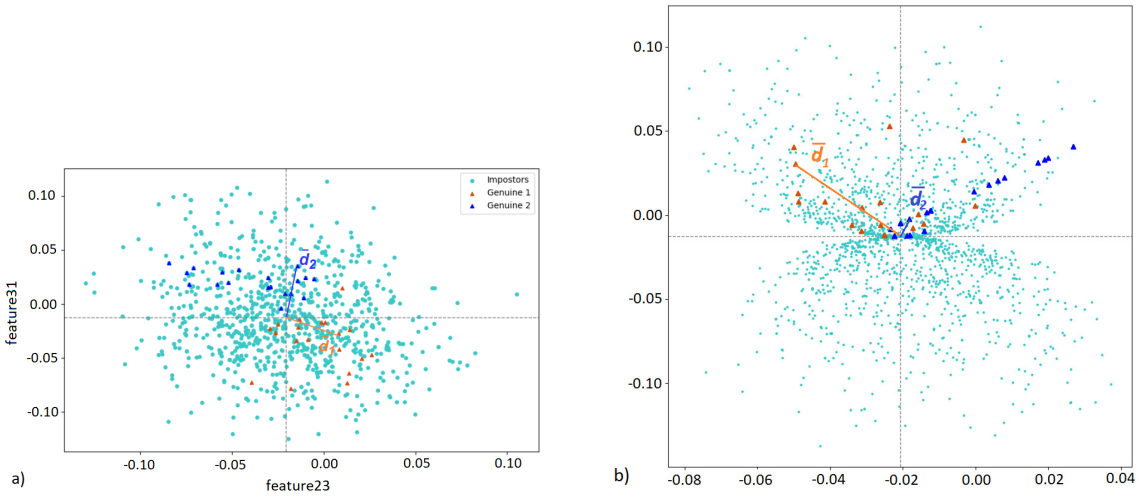


Figure 4. Calculation and visualization of the distance from the "center of mass" to the images in accordance with (2): a) original, b) meta-subspace.

$$d(a,m) = \sqrt{(a_i - m_i)^2 + (a_j - m_j)^2} * sin(\widehat{\overline{d}, \overline{v}}) \quad (2)$$

where $sin(\widehat{\overline{d}, \overline{v}})$ - sine of the angle between $\overline{d_l}$ and $\overline{v_l}$ vectors.

Measure (2) is based on the so-called trigonometric distance leveling, which is used when changing geodetic works, or more precisely produces tacheometric surveys [32]. Despite the fact that metric (2) is not an analogue of full trigonometric leveling, it has a corrective effect on the original distance and gives a more informative idea of the positions of the images relative to each other. Let us illustrate

 doi:10.20944/preprints202601.0869.v1

what visual changes in the subspace of the original pair of features look like (Figure 4a) if we change the location of all images relative to the "center of mass" in accordance not with the Euclidean distance, but with the distance after leveling (2). To do this, we will construct a derivative subspace taking into account trigonometric leveling (or meta-subspace, Figure 4b).

An alternative metric (3) avoids additional computational costs for calculating distances:

$$a_l^{'} = sin(\widehat{\overline{d}, \overline{v}}) \tag{3}$$

You can compare efficiency metrics only based on experiment (see below 4). The quantities calculated using formulas (2) and (3) can also be considered as meta-features that describe the location of the biometric image in the subspace of a pair of initial features, taking into account trigonometric leveling. Then formulas (2) and (3) are a mapping of the original feature space into the meta-feature space.

The simplest trigonometric neuron is based on metric (4) and takes as input meta-features constructed using mappings that take into account any version of trigonometric leveling, which was the reason for the name of the proposed neuron model. The represented neuron summarizes the values of meta-features. More complex schemes for integrating input values are also possible (for example, weighted summation, multiplication, etc.), as well as other options for trigonometric leveling of distances (formulas (2) and (3) are a special case of such transformations). More complex designs of trigonometric neurons are beyond the scope of this article.

$$y = \sum_{l=1}^{k} a_l^{'} \tag{4}$$

where k is the number of synapses (inputs) of the neuron. The neuron makes a decision on the sum of input values of meta-features according to a $\varphi(y)$ two-level threshold activation function (5):

$$\varphi = \begin{cases} 1, & y \geq T_2 \\ 0, & T_1 < y < T_2 \\ -1, & y \leq T_1 \end{cases} \tag{5}$$

where T_1 and T_2 are decision thresholds in favor of one of the three function values.

The threshold function is necessary to quantize the results of transformations, since neurons must generate a binary code at their outputs.

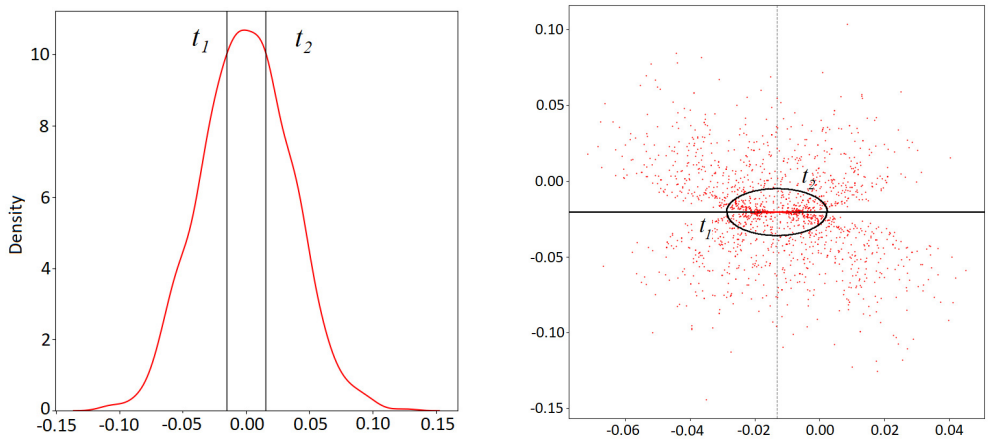
In order for the t-neuro-extractor to associate the user's key with the biometric image, the output values of the activation function must be converted into binary states of the type "10", "00", "01". Then each neuron will produce 2 bits of information. To implement this procedure, you need to select the hash transformation number for the neuron (Table 1) .

 doi:10.20944/preprints202601.0869.v1

Table 1. 24 options for hashing transformations of a neuron response into binary code.

№	-1	0	1	№	-1	0	1
1	11	00	01	**13**	01	00	11
2	11	00	10	**14**	01	00	10
3	11	01	00	**15**	01	10	00
4	11	01	10	**16**	01	10	11
5	11	10	00	**17**	01	11	10
6	11	10	01	**18**	01	11	00
7	00	01	11	**19**	10	00	01
8	00	01	10	**20**	10	00	11
9	00	10	01	**21**	10	01	11
10	00	10	11	**22**	10	01	00
11	00	11	10	**23**	10	11	00
12	00	11	01	**24**	10	11	01

A trigonometric neuron is partially connected. To configure it, you need to select several pairs of features and determine t_1 and t_2 thresholds for each pair. The thresholds should divide the corresponding subspaces into three sectors $([-\infty, t_1], (t_1, t_2), [t_2, \infty])$ so that each sector contains approximately an equal number of "Impostor" training samples. Pairs of features should be selected so that all "Genuine" training samples for each pair fall into one specific sector (Figure 5).



(**a**) (**b**)

Figure 5. Visualization of thresholds obtained based on metrics 3 a) on the probability density plot of the meta-feature, b) in the meta-subspace of a pair of features.

All thresholds of the trigonometric neuron depend only on the training samples of "Impostor" images, which are not secret. The "Impostors" samples must be representative and consist of data from random people; it is collected by the developer of the biometric system once and must be anonymized. For this reason, it does not contain the data of subjects registered in the system. Accordingly, neuron thresholds are open information. Keeping the thresholds secret, like the Impostors training set, is pointless. This is due to the fact that an attacker can collect a representative set of "Impostors" samples and calculate thresholds that are not strictly equal to the system ones, but quite close to them.

Thus, the thresholds are the same for all users and are calculated in advance by the biometric system developer using the algorithm described in the next paragraph.

Therefore, to train a neuron, it is enough to select the appropriate hash transformation number and pairs of features, i.e. define an input table that is also open. The hash transformation is chosen randomly, but only among those options that correspond to the selected sector and the key bit pair to which the neuron is tuned. For example, if the neuron should generate "11" and the second sector was selected, then the available options are 11, 12, 17, 18, 23, 24.

 doi:10.20944/preprints202601.0869.v1

The only secret element of the neuron is the sector number that was selected during training and which is associated with two bits of the user key. The attacker does not know which bits the neuron is set to and does not know which sector is the correct one, which determines the security of the biometric template.

When selecting a key, you should carry out a brute force of biometric images represented by feature vectors. The possibility of directed brute force is not available to an attacker, since he does not have an indication of the proximity of the binary code generated at the output of the t-neuro-extractor and the user key (there is only a check - true/false, as is the case with a password hash). Therefore, to hack t-neuro-extractor, you need to select the correct sectors of all neurons at once (not separately), since only in this case will the correct user key be obtained in its entirety. Thus, if an attacker does not know the user by sight (does not have his biometric image), then he will not be able to extract the key (and vice versa).

A table of neuron connections can be considered as the knowledge of a specific user's t-neuro-extractor.

Thus, having arrived at the input of the neuron, pairs of features are processed using the functional (2) or (3) and fall into the threshold function, which assigns the resulting sum of values to one of three states "-1", "0", "1", which , in turn, is converted to binary code.

3.2.2. Calibration of t-Neuro-Extractors

For t-neuro-extractors to work correctly, it is necessary to correctly configure the thresholds for each subspace of feature pairs. The easiest way to demonstrate the division into sectors is to plot the probability density of meta-features (Figure 5a). The thresholds, represented on the graph as two vertical lines, are calculated as follows:

1. The values of meta-features are calculated using metrics (2) or (3) :

$$A = \{a'_{l_1}, a'_{l_2}, a'_{l_3}, ..., a'_{l_q}\},$$

where q is the number of biometric images in the subspace of the (a_l, a_j) feature pair.

2. The obtained values of meta-features are ranked in ascending order:

$$B = sort(A),$$

where $sort(A)$ is a function that sorts the elements of set A in ascending order.

3. The resulting increasing array is divided into 3 equal sectors, resulting in t_1 and t_2 thresholds:

$$t_1 = b_{\frac{q}{3}}, t_2 = b_{\frac{2q}{3}},$$

where b is the value of the element of array B.

This procedure allows you to avoid constructing an empirical probability density function if the distribution law of meta-features is unknown and differs from normal. Algorithm in Figure 6 seeks to find thresholds at which $\Phi(t_1) \approx 0.333, \Phi(t_2) \approx 0.667$, where $\Phi(.)$ is the distribution function of the meta-feature. The larger the sizes of the "Impostors" training set, the closer the threshold values are to the desired ones.

 doi:10.20944/preprints202601.0869.v1

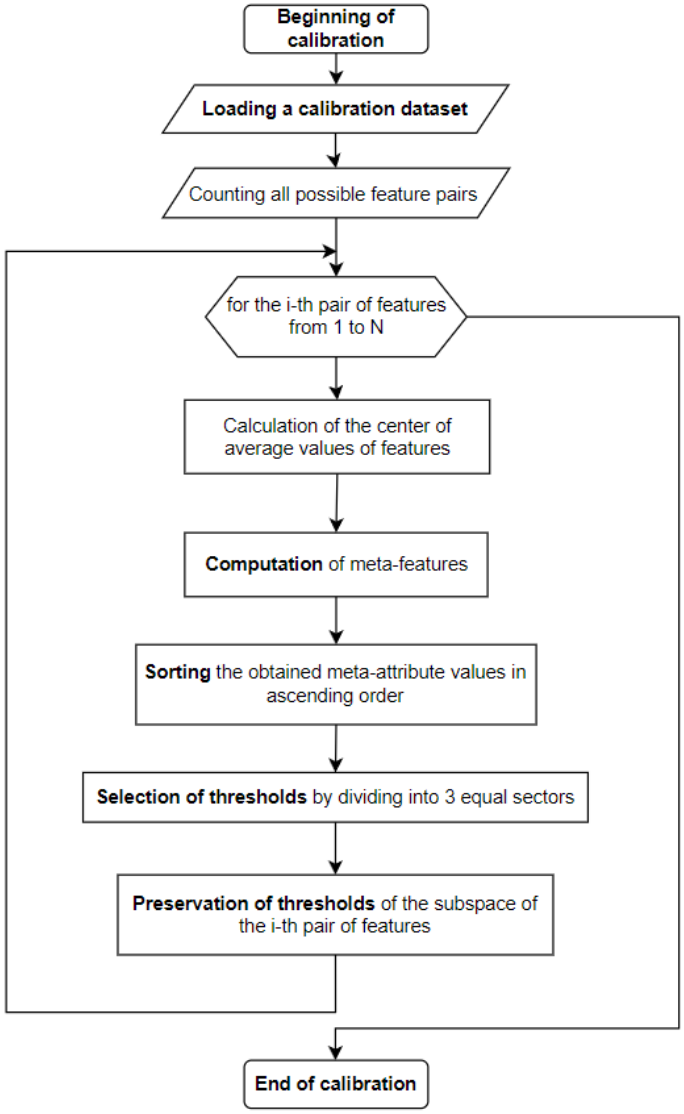


Figure 6. Calibration algorithm of t-neuro-extractors

For each pair of features, the threshold values are saved and used in the synthesis and training of the t-neuro-extractor. After the algorithm has completed its work, there is no need to store the "Impostor" data used during its work, if the thresholds do not need clarification (adjustment).

The algorithm is potentially applicable under various feature and meta-feature distribution laws, although the nature of the distribution may affect the efficiency of the t-neuro-extractor. The most significant factor is the discrepancy between the distribution laws of features for the "Genuine" and "Impostors" training samples. For example, if for calibration were used images obtained under poor lighting or high occlusion, and for training the t-neuro-extractor were used high-quality images with good lighting without obstructing objects , then the "Genuine" samples may be biased relative to the "center of mass". This situation will reduce the quality of t-neuro-extractor training. The presence of bias-sensitive features is determined by the architecture of the feature extractor. In present work, the distributions of the initial characteristics were close to normal (sometimes a slight asymmetry of the distribution was observed), which was checked by the chi-square test.

3.2.3. Synthesis and Training of t-Neuro-Extractor

The T-neuro-extractor is built separately for each subject, that is, for each class of the training set of biometric data. The t-neuro-extractor synthesis and training procedure is carried out automatically

 doi:10.20944/preprints202601.0869.v1

without the use of iterative learning based on the backpropagation method. The t-neuro-extractor training algorithm is shown in Figure 7.

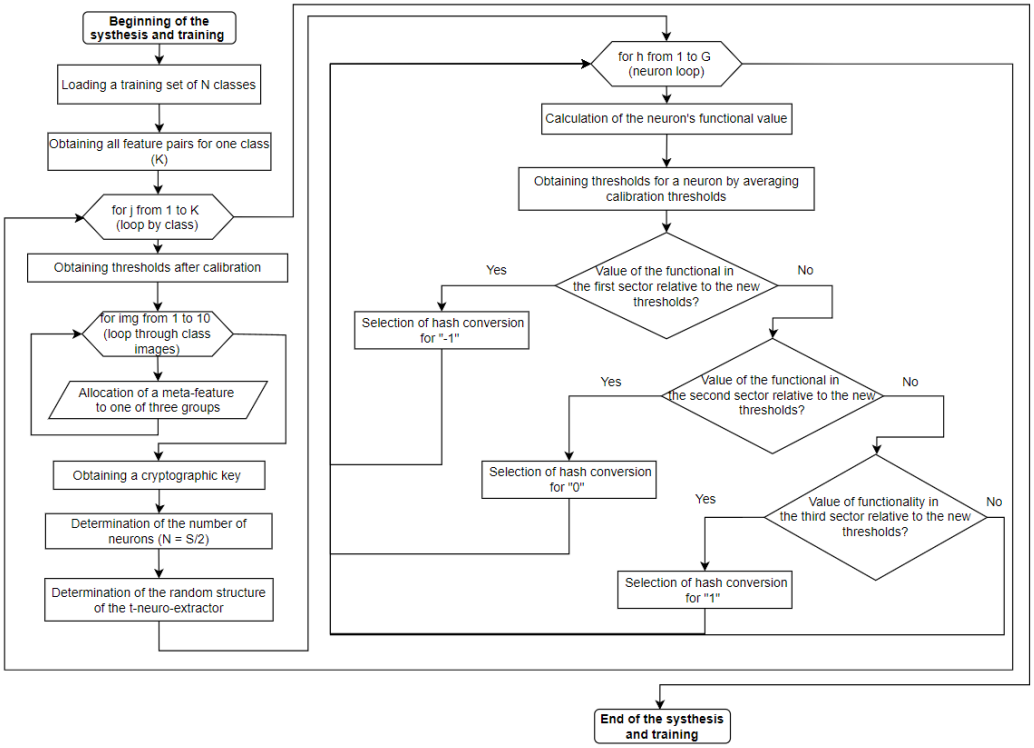


Figure 7. Algorithm for training a neural fuzzy extractor based on a trigonometric neuron

The first stage of constructing a neural fuzzy extractor based on a trigonometric neuron will be the selection of such pairs of features of the training data set, in the subspace of which the biometric images of the subject will be located strictly in one of three sectors. Note that the "strictness" of the arrangement of images in sectors can be varied based on the size of the "Genuine" training samples (the more training images, the more deviations from a specified sector are allowed). That fact that the more "Genuine" training images lie outside the selected sector, the higher the probability of "false refusal" errors will be should be taken into account. Too strict rules with large volumes of the "Genuine" training set can lead to the impossibility of synthesizing t-neuro-extractor. This is because in such cases there may not be enough suitable feature pairs, which limits the ability of the algorithm to work effectively and may lead to its failure.

Three non-overlapping groups of feature pairs are formed (a separate group for each sector). For each neuron, pairs of features from a certain (one) group are randomly selected. The number of pairs of features is equal to the number of synapses of the neuron. The inputs of each neuron must be unique and no pair can be reused in another neuron. Based on each group of feature pairs, approximately an equal number of neurons are formed. This is necessary to avoid statistical biases and reduce the information entropy of the codes at the output of the t-neuro-extractor when "Impostor" images arrive at its inputs (a slight discrepancy of 2-3 neurons is allowed). When neurons are randomly mixed in the neural fuzzy extractor structure, it becomes impossible to perform a directed search of input images to determine the user key (or its parts).

Threshold values for the activation function (7) of a neuron are calculated using the formulas:

$$T_1 = \frac{1}{k}\sum_{z=1}^{k} t_{1z} \qquad and \qquad T_2 = \frac{1}{k}\sum_{z=1}^{k} t_{2z},$$

 doi:10.20944/preprints202601.0869.v1

where k is the number of synapses (inputs) of a neuron; z – number of synapse (input) of the neuron; t_{1z} and t_{2z} are thresholds for a pair of features entering the neuron input, obtained as a result of the calibration algorithm.

The threshold values for the feature pairs entering the neuron input are determined based on the distribution of biometric features obtained by using the neural network. For the trigonometric leveling measure (2), the thresholds are completely correspond to the distribution of areas distribution area of these biometric features. Thus, the range of threshold values can vary significantly depending on the data set used and the specific feature pairs. The thresholds are determined based on the sorted values of the meta-features and divide the data into three sectors with an equal number of training samples of the "Impostor" class. For the measure (3), based on the averaged sine of the angle between the vectors, the threshold values are stably in the range from 0 to 1. This is because the sine function takes values in this interval, and averaging further narrows the distribution.

The t-neuro-extractor structure should be built on the basis of $N = S/2$, where N is the number of neurons, S is the desired length of the cryptographic key.

After neurons are trained, the distribution of hash transformations is uniform. Due to this, an equally probable appearance of states "0" and "1" is achieved at the output of the t-neuro-extractor when images of "Impostors" are received at the inputs, and as a result, high entropy.

4. Experimental Results

This section is devoted to a comprehensive assessment of the performance of the proposed biometric facial authentication system. The main metric was Equal Error Rate (EER), which is the state of the system in which False Acceptance Rate (FAR) and False Rejection Rate (FRR) have approximately the same value. FAR, FRR and EER can be measured as percentage or probability. EER is calculated from ROC curves that demonstrate the system's ability to balance security (minimizing FAR) and usability (minimizing FRR). The lower the EER value, the higher the efficiency of the biometric system. You can balance the FRR and FAR indicators by changing the acceptance threshold, measured by the number of acceptable errors in the binary code generated using t-neuro-extractor. In practice, this is achieved by using error-correcting codes [33].

In some experiments, the unbalanced accuracy metric was also used, reflecting the overall percentage or probability of correct decisions. The mathematical representations of the described metrics used for further research are given in Table 2.

Table 2. Metrics for evaluating the performance of the proposed solution.

Metrics	Formula	Peculiarity
Accuracy	$\frac{TP+TN}{TP+FP}$	Evaluates the overall model performance without taking into account class balance.
FRR (false rejection rate)	$\frac{FN}{FN+TP}$	The percentage of legitimate users incorrectly rejected by the authentication system, i.e., when the system fails to recognize an authorized individual.
FAR (false acceptance rate)	$\frac{FP}{FP+TN}$	The percentage of unauthorized users incorrectly accepted by the authentication system, i.e., when the system mistakenly grants access to an unauthorized individual.
EER (equal error rate)	$EER = FAR = FRR$	Determined by setting a threshold value at which FAR and FRR are equal.

4.1. Data Sets for Experiments

To select the optimal parameters of the neural fuzzy extractor, its initial testing and further training, as part of the presented study, a special SFDv1 data set was generated. SFDv1 included video recordings of the faces of 75 subjects. For each subject, three video recordings were generated, obtained under three different types of lighting. First type of lighting is daylight (light source - window), second

 doi:10.20944/preprints202601.0869.v1

type is artificial light in the room (lamp from above) and third type is absence of light in the room (dark corner of the room, where faces are visible). The average duration of the video recording was about 45 seconds per subject. To obtain various positions of the face relative to the frame, the subjects made circular movements with their heads, and also turned to the right, left, up and down. Each subject signed an informed consent to participate in the experiment and consent to the processing of personal data. The data of the experiment participants was stored and processed in anonymized form. Each of the three video recordings for one subject was processed and randomly selected 20 frames (facial images, images). Thus, for each subject, a package of 60 facial images was formed, represented by three types of lighting.

As additional data sets for conducting comparative analysis and testing the effectiveness of the proposed system, the following were used in the work:

1. One of the most popular and widely used datasets in the field of face recognition is LFW (Labeled Faces in the Wild) [34]. The LFW dataset contains a set of images of celebrity faces (JPEG format), collected in various lighting conditions, angles and emotional expressions. In total, LFW contains more than 13,000 facial images of 5,749 different people. However, only a few classes are represented by more than 20 images, or rather only 57. Due to this, only 57 classes of the LFW dataset are suitable for training and testing the proposed secure biometric authentication system.

2. A small but representative Faces94 dataset [35], which is a set of face images (JPEG format) of 153 people. Each person is represented by an average of 20 images. The total number of images is 3060. Note also that the images in Faces94 are presented in low resolution, which also adds complexity to recognition algorithms.

4.2. Testing Trigonometric Neurons

Before assessing the performance of the proposed secure biometric system, it is necessary to determine the boundary values of the hyperparameters. These hyperparameters include such as the number of neuron synapses, within which it makes sense to test the system (for reasons of learning speed and potential accuracy of the t-neuro-extractor). The sought hyperparameters are:

1. Number of inputs of one neuron (number of synapses) η;

2. ω – the share (percentage) of images in the training set that fall into one of three sectors, at which a decision is made to assign a pair of features to a specific sector (“strictness” of the location of images in sectors).

On the one hand, it is intuitively clear that the more synapses each neuron has, the “more accurately” the t-neuro-extractor will function. This is due to the fact that each neuron will receive an exhaustive amount of information as input, which potentially reduces the likelihood of erroneous decisions. However, the number of feature pairs is limited. Therefore, it is necessary to find a balance between the number of synapses and the key length. The testing described in this section makes it easier to find the optimal t-neuro-extractor configuration. It is based on the accuracy metric applied to a single neuron as a classifier.

Figure 8 shows graphs obtained as a result of calculating accuracy depending on the number of synapses. The number of synapses varied from $\eta = 2$ to $\eta = 20$. In addition, each of the described dependencies was realized at different values of ω ($\omega = 70\%$, $\omega = 80\%$, $\omega = 90\%$ and $\omega = 100\%$). The neuron was tested on the SFDv1 dataset, with 10 images from each user used for training. The experiment was carried out separately for each of the two types of trigonometric neuron (2) and (3) presented in the work.

 doi:10.20944/preprints202601.0869.v1

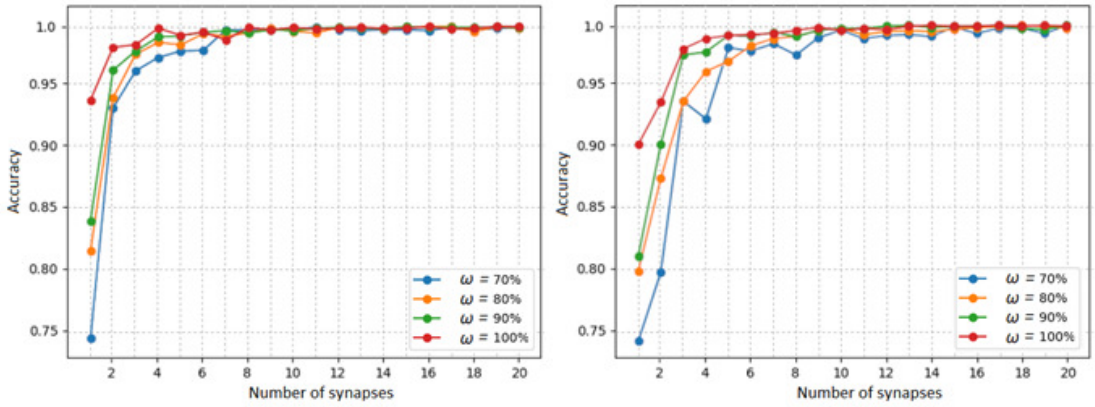


Figure 8. Dependence of the accuracy of the operation of a trigonometric neuron on the number of synapses for different values of ω: a) for a neuron based on measure (2); b) for a neuron based on measure (3).

It is necessary to select such neuron configurations in which it is possible to obtain maximum accuracy with the lowest values of η and ω. This will produce the longest key length.

4.3. Performance Assessment

An experiment was conducted to select the initial test configuration of t-neuro-extractor based on the SFDv1 dataset (Table 3). 10 random users were selected from the data set to train and test it, the rest were used for calibration. When choosing suitable configurations, key attention is paid to the ability of t-neuro-extractor to produce a key of the maximum length for each user with a minimum error in recognizing its individual neurons.

 doi:10.20944/preprints202601.0869.v1

Table 3. Comparative table of optimal configurations for assembling a t-neuro-extractor.

№	"Severity" of entering the sector, ω	Number of synapses, η	Number of neurons	Key length, bits	Successful synthesis
Neuron based on measure (2)					
1	70%	7	256	512	100%
			512	1024	100%
			1024	2048	100%
2	80%	8	256	512	96%
			512	1024	92%
			1024	2048	86%
3	90%	7	256	512	80%
			512	1024	68%
			1024	2048	60%
4	100%	4	256	512	72%
			512	1024	64%
			1024	2048	58%
Neuron based on measure (3)					
1	70%	10	256	512	100%
			512	1024	100%
			1024	2048	100%
2	80%	9	256	512	100%
			512	1024	100%
			1024	2048	100%
3	90%	12	256	512	100%
			512	1024	84%
			1024	2048	74%
4	100%	9	256	512	72%
			512	1024	68%
			1024	2048	52%

Only one t-neuro-extractor configuration based on measure (2) meets the specified requirements. Apparently, using the Euclidean distance as a basis introduces noise. However, such a t-neuro-extractor is capable of producing a 2048-bit key using only 70% of the images that fall into one of the sectors. In the case of neurons based on measure (3), two configurations are suitable, allowing 100% of users to successfully produce long cryptographic keys (2048 bits). However, according to the experiment presented in Figure 8, configuration No. 1 has the highest accuracy of recognition of an individual neuron (99.93% versus 99.81% for No. 2). In this regard, the specified configuration was chosen as the main one for the t-neuro-extractor based on measure (3).

The considered configurations were further used to conduct a comparative experiment for three experimental datasets.

An experiment was carried out to evaluate the effectiveness of the proposed system, which was repeated 3 times, each time a different division of data sets into disjoint sets was carried out:

- A calibration set (consists of one third of the images from the data set);
- A set that used for training and testing of t-neuro-extractor (consists of two thirds of the images from the dataset).

Thus, each image was used as a training, test and calibration sample.

The results of the calculations are presented in Figure 9, respectively. The graphs show the dependence of FRR and FAR on the permissible number of incorrect bits in the binary code generated by t-neuro-extractors. The number of incorrect bits is measured by the Hamming distance between the binary code generated at the output of t-neuro-extractors and the correct user key. The maximum value of the Hamming distance axis is equal to the key length.

 doi:10.20944/preprints202601.0869.v1

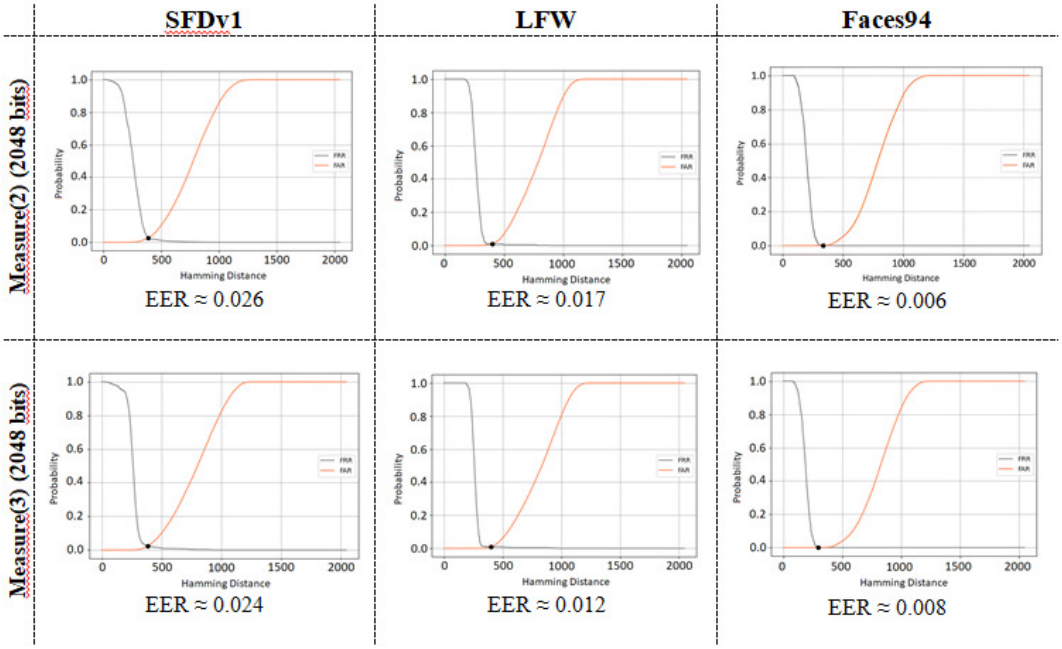


Figure 9. Graphs of average EER values for t-neuro-extractor based on various neurons with a maximum key length (2048 bits)

As can be seen from the figures, the best performance was achieved when conducting experiments based on the Faces94 dataset ($EER \approx 0.006$). In this case, the key length reaches 2048 bits, which is a decent result.

Table 4 presents a comparison of the system proposed in the work with the most relevant works in recent years that offer alternative solutions for biometric templates protecting (BTP).

Table 4. Comparative analysis of the proposed solution with existing implementations of secure biometric systems.

BTP Approach	Data set	Metric, probability	Maximum key length, bits
Peng J. et al. [14]	Faces94	EER = 0.0022	322
Rathgeb C. et al. [16]	FERET + FRGCv2	FRR <0,01 with FAR <0.0001	-
Dong X. et al. [19]	LFW	Accuracy = 0.9853 (98.53%) with EER = 0.001	-
	VGGFace2	Accuracy = 0.9853 (98.53%) with EER = 0.001	-
	IJB-C	Accuracy = 0.4373 (43.73%) with EER = 0.001	-
Neural fuzzy extractor [10]	SFDv1	EER = 0.004	256
MEB Encoding (Kumar Pandey R. et al.) [2]	PIE	EER = 0.0114	1024
CNN (Kumar Jindal A. et al.) [1]	PIE	EER = 0.036	1024
t-neuro-extractors (measure(2))	SFDv1	EER = 0.026	2048
	LFW	EER = 0.017	2048
	Faces94	**EER = 0.006**	**2048**
t-neuro-extractors (measure(3))	SFDv1	EER = 0.019	2048
	LFW	EER = 0.012	2048
	Faces94	**EER = 0.008**	**2048**

 doi:10.20944/preprints202601.0869.v1

The approach proposed in this work is able to achieve relatively low EER values with a maximum key length.

5. Justification of the Cryptographic Strength of the Proposed Solution

From a cryptographic point of view, in the proposed t-neuro-extractor model, the biometric image plays the role of plaintext, and the generated cryptographic key corresponds to the ciphertext. This allows analyzing potential attack vectors aimed at compromising keys using both theoretical and empirical methods. The cryptographic strength of the system is assessed by analogy with classical cryptographic schemes, in terms of resistance to plaintext and/or ciphertext attacks, resistance to pattern prediction and reproduction, and the ability to generate keys with high entropy.

A brute force attack is difficult because the t-neuro-extractor structure assumes the simultaneous selection of correct sectors for all neurons. With a key length of 2048 bits, this means the need to select a single acceptable combination of configurations for 1024 neurons, which, in the absence of feedback on the proximity of the result, makes the attack practically unrealizable. Even with pre-known cryptographic keys, an intruder cannot restore a biometric image, since the key generation process itself relies on meta-features constructed in a functionally nonlinear feature space.

Similarly, a known-plaintext attack (i.e., a known biometric image) is also ineffective because the internal structure of t-neuro-extractor depends on the random distribution of neurons, selected hash transformations, and sectors. Even if the intruder has a biometric image, he cannot reproduce the exact table of connections and correspondences of neurons with binary values, making it impossible to recover the key.

In addition, experiments aimed at empirically analyzing the system's resistance to these attacks were conducted:

1. Analysis of the distribution of hash transformations: for 75 users (SFDv1 dataset) with different keys associated with t-neuro-extractor. All t-neuro-extractor neurons were divided into 4 groups by output states (01, 00, 11, 10), and histograms of the frequency of occurrence of a particular hash transformation were constructed for each group (Figure 10). Similarly, neurons were divided into three groups by sectors and analyzed using histograms (Figure 11).

The analysis of the distribution of output bit combinations and their grouping by sectors showed that the generation of cryptographic keys in the proposed neural fuzzy extractor model does not lead to statistically significant deviations that allow to restore a biometric image. The uniformity of the distribution of bit values confirms the absence of systematic vulnerabilities associated with the predictability of output data, and the differentiation of neurons by sectors ensures effective filtering of uninformative features, excluding the possibility of restoring the original data by reverse engineering. These results confirm the stability of the proposed scheme to attacks with known ciphertext and known plaintext, demonstrating its cryptographic strength.

2. To empirically evaluate the cryptographic strength of the proposed scheme, a series of experiments aimed at simulating a known-plaintext attack were conducted. The goal of the experiment was to establish whether there are hidden patterns in the distribution of hash transformation variants that could be used by an intruder to restore the t-neuro-extractor structure or the original biometric template.

 doi:10.20944/preprints202601.0869.v1

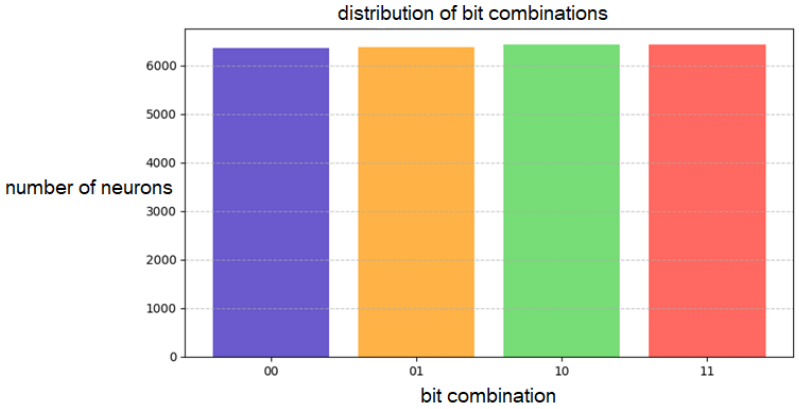


Figure 10. Histogram of the frequency of occurrence of hash transformations.

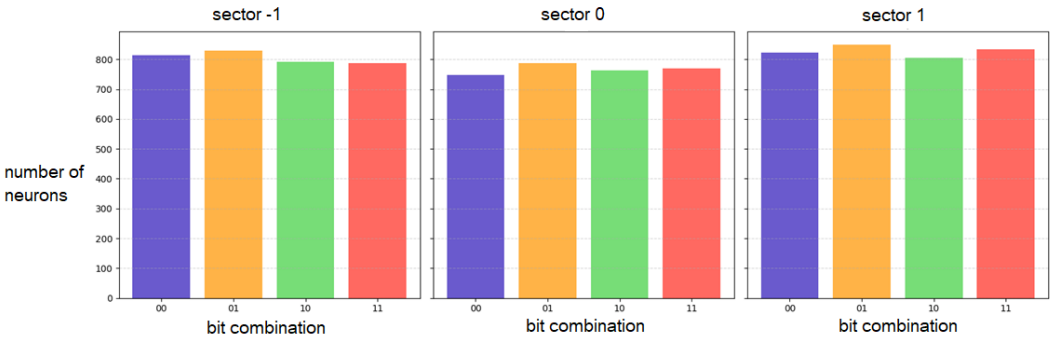


Figure 11. Histogram of hash transformations frequency of occurrence by sectors.

At the first stage of the analysis, the distribution of transformation variants within each of the three sectors (sector 0, sector 1, sector -1) was investigated. The results, presented in Figure 12, showed that the frequency of occurrence of different hash transformation variants in each sector remains uniform. The observed frequency variation is small and falls within the statistical noise, this show the absence of preferred transformations associated with specific sectors.

At the second stage, the analysis was conducted on the output states of neurons corresponding to the codes “00”, “11”, “01” and “10” (Figure 13). The study of the distributions showed that, despite small local deviations, the overall structure of the distributions remains almost uniform, and no pronounced correlations were found between the codes and hashing options. This confirms the absence of a structural dependence that then can be used to restore the original biometric features. We suggest that small deviations be caused by an insufficient sample size of users, and cannot be features or indicators specifying a specific transformation or bit bigram.

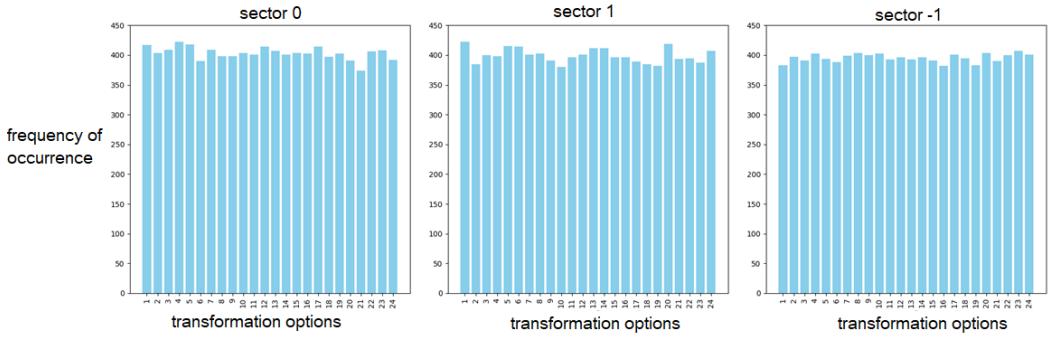


Figure 12. Histogram of the distribution of transformation options within each of the three sectors

 doi:10.20944/preprints202601.0869.v1

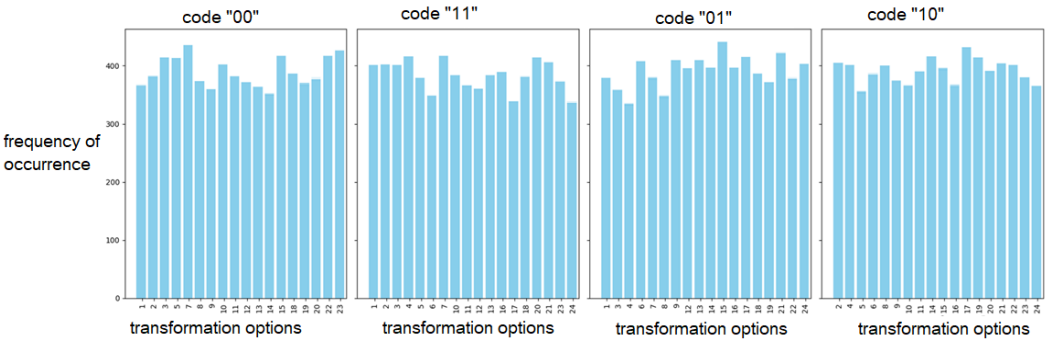


Figure 13. Histogram of output states of neurons corresponding to codes “00”, “11”, “01” and “10”

Finally, in the third stage, a summary histogram of the distribution of all transformations by the generated and trained t-neuro-extractor models for 75 different users was constructed (Figure 14). The obtained data show that the frequency of occurrence of each of the 24 variants of hash transformations in the aggregate remains approximately the same, with fluctuations within the acceptable statistical spread. No clearly dominant transformations were recorded.

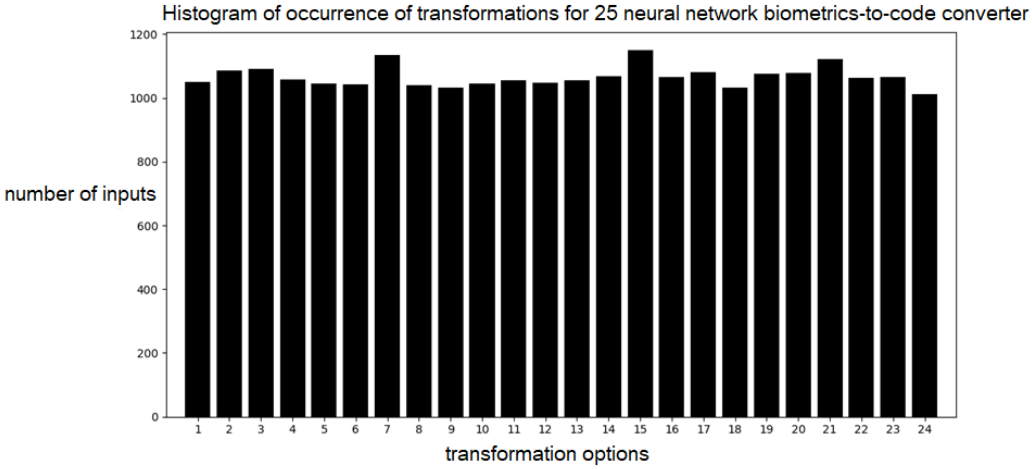


Figure 14. Histogram of output states of neurons corresponding to codes “00”, “11”, “01” and “10”

Thus, the conducted analysis of distributions by sectors, output states of neurons and the entire set of users confirm the absence of predictable patterns in using hash transformations. The gradation of the intensity of visualizations shows a sufficient randomness in the t-neuro-extractor structure. So, the intruder does not have access to any effective methods for restoring the structure of the model or the original biometric image based on the analysis of output codes. This confirms the high cryptographic resistance of the proposed scheme against attacks with known plaintext.

To further substantiate the possibility of generating cryptographically strong keys up to 2048 bits long, a separate experiment was conducted aimed at quantitatively assessing the entropy of the output code and its stability when presented with biometric images of the “Impostor” class. During the experiment, the lengths of the output code varied (8, 10, 12 and 14 bits), as well as the number of inputs (synapses) in neurons from 1 to 50. Entropy was measured as the Shannon entropy of the resulting code, reflecting the level of cryptographic randomness:

$$H(X) = \sum_{i=1}^{N} p(x_i) \log_b p(x_i),$$

 doi:10.20944/preprints202601.0869.v1

where N — is the length of the output code of the t-neuro-extractor (number of bits); $p(x_i)$ — is a relative frequency of occurrence of a certain state (for example, “1”) in the -th bit of the output code during multiple presentations of biometric images.

The stability of the output code was determined based on the calculation of the normalized deviation of the probability of the appearance of “1” in the bits of the key at the outputs of neurons from a random value of 0.5. For each neuron, the relative frequency of the appearance of a single output was determined when multiple images of the same subject (“Impostor”) were presented, then the absolute deviation of this frequency from 0.5 was calculated with normalization, and the results were averaged over all neurons. Thus, stability reflects the level of predictability of the output relative to random behavior, in which the probability of “1” is 0.5:

$$\gamma_k = \sum_{l}^{L} 2(P_l(1) - 0.5),$$

where k — is a number of user ("Impostor"); L — is a total number of neurons in the output layer of the t-neuro-extractor; l — is a neuron index; (1) — is a probability (relative frequency) of the value “1” appearing at the output of the $l-th$ neuron when presented with different biometric images of subject k; γ_k — is an estimate of the average stability of the output codes for k subject.

The probability estimation was performed empirically based on multiple presentations: 10 images for each subject were used in the experiment.

The experiment showed that with an increase in the number of synapses, the stability of the output code increases, which is associated with an increase in the stability of neuron activation when biometric images are presented (Figure 15). A moderate decrease in the entropy of the output code is observed. This is associated with a decrease in the variability in the results of neuron operation. At small code lengths (8–14 bits), the maximum entropy values (up to 0.97 per bit) were achieved with a few synapses (1–3), and the maximum stability (up to 0.71) was observed with 30–50 synapses. The optimal balance between entropy and stability was recorded with a synapses number in 10–20, where the entropy remained above 0.90, and the stability did not exceed 0.65. These observations were made at small code lengths.

To evaluate the system behavior for large key lengths (256, 512, 1024, and 2048 bits), a series of output code stability calculations was additionally performed, since direct calculation of information entropy for such codes is practically impossible (it requires a huge sample exceeding 22048 examples and a colossal amount of computing resources). The results showed that with an increase in the key length, the average stability of remains at the level of 0.64, comparable with the results obtained for short lengths. Since the entropy calculation for large code lengths was not performed directly because of data volume and computing resource limitations, it is assumed that the entropy level will be close to the values recorded for short key lengths, considering similar configurations of neurons and input data (deviations are possible, but a decrease in entropy lower than 0.8 is extremely unlikely).

Thus, the experiment confirms that the proposed t-neuro-extractor architecture can generate (releasing) long cryptographic keys (up to 2048 bits) with sufficiently high entropy of output sequences when receiving “Impostor” images at the inputs of the t-neuro-extractor with virtually no increase in the probability of authentication error.

 doi:10.20944/preprints202601.0869.v1

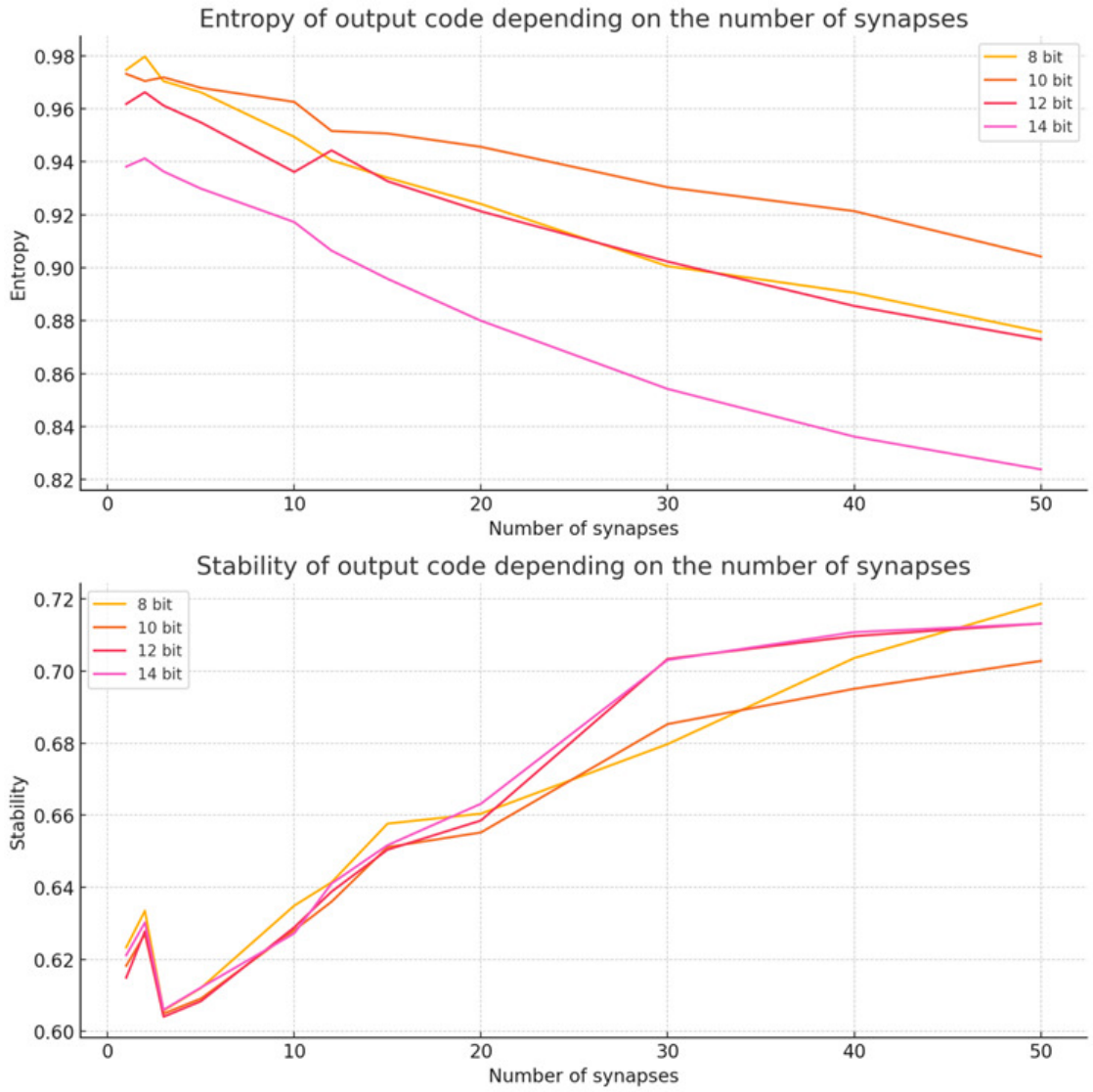


Figure 15. Stability of the output code depending on the number of synapses

The proposed architecture also provides resistance to false acceptance attacks, pre-image attacks, and decodability attacks. The low Equal Error Rate values reached in the main experiment confirm the minimal risk of false acceptance. Since the biometric template and the generated key are not directly related, pre-image attacks are infeasible: knowledge of the key does not provide information about the template. The lack of explicit correction codes and the non-standardized internal structure of t-neuro-extractor make it impossible to apply decodability attacks similar to those described in the works on fuzzy commitment [36].

The system also satisfies the criteria of revocability and unlinkability. Revocability (the first criterion) is achieved through the t-neuro-extractor's ability to be quickly retrained with a new key, avoiding any need to alter user biometric data. Unlinkability is guaranteed because biometric data is not stored directly; each model instance is unique, preventing session linking through data or identifiers. The model works with meta-features, which are calculated based on the original feature vector. This operation is unidirectional and does not susceptible to reverse engineering (it is not possible to obtain a feature vector a from a meta-feature vector a').

Thus, the presented theoretical arguments and experimental results confirm the robustness of the proposed architecture. The model provides resistance to all key attack types, generates keys up to 2048

 doi:10.20944/preprints202601.0869.v1

bits long with high entropy, demonstrates stability during repeated presentations, and supports the fundamental properties of modern biometric template protection systems.

6. Conclusions

The protected execution mode of neural network image classification algorithms is an important component of the concept of trusted AI. This mode serves to minimize information leaks about knowledge and the decision-making process. In particular, it makes it difficult to analyze the operations performed by AI, prevent unauthorized control of AI, and extract and interpret its knowledge by unauthorized persons. Therefore, protected mode can be called the opposite of explainable AI, which in turn, on the contrary, aims to provide the user with the maximum amount of information about the decisions made by AI.

Protected mode is based on special neural network models and architectures. An example is the proposed model of biometric-based key generation for users facial authentication based on trigonometric neurons - t-neuro-extractor. The model is integrated into the structure of a secure facial biometric authentication system and represents a separate block that classifies biometric images of users in a secure mode. The mathematical apparatus of t-neuro-extractor is based on the use of two alternative measures of proximity of biometric images in the subspace of feature pairs. This approach can significantly improve the classification accuracy of biometric images and the length of cryptographic keys, while providing protection against known attacks against secure neural network algorithms.

The developed solution can be effectively applied in various fields, including highly secure authentication systems for government and corporate structures, financial technologies, online banking, cryptocurrency platforms, medical systems and IoT devices. Its use in these areas will ensure a high level of security, preventing unauthorized access to confidential data, protecting against attacks and information leaks.

An experimental evaluation of a secure authentication system based on the proposed model demonstrated its higher efficiency compared to alternative implementations of biometric-based key generation for users facial authentication. The best value on the test datasets was $EER \approx 0.006$ on the open Faces94 dataset and 0.012 on LFW. The resulting implementation is capable of producing a key with a length of 2048 bits, which is a higher indicator of the reliability of the proposed solution compared to those achieved previously.

To extract features, the neural network architecture Inception-ResNet v1 was used. According to the results of the experiments, the features extracted from facial images had predominantly weak mutual correlations. To form neurons, as a rule, features with no correlation or insignificant correlation were selected. Thus, the developed model works well with weakly correlated features. In this sense, it complements the correlation neuron model, which, on the contrary, shows high results when the input features are highly correlated [9] and is unable to work with independent features.

Because of its special neuron architecture, the T-neuro-extractor generates longer keys with higher entropy. These neurons can form networks capable of learning non-iterative algorithms, without using gradient descent. This allows us to create a universal tool (algorithm) for synthesizing and automatically training t-neuro-extractor for any modality, unlike neural fuzzy extractor, based on classical neural networks, the training of which is difficult to automate.

Further research will be aimed at testing and adapting the proposed model to other biometric modalities, as well as combining the proposed neuron model with correlation neurons [9] to create a hybrid neural network model biometric-based key generation, consisting of a layer of neurons of different types (trigonometric and correlation). It is assumed that such a model will be the most resistant to possible destructive influences (adversarial attacks, knowledge extraction, etc.).

Author Contributions: Conceptualization, A.V. and A.S. (Alexey Sulavko); methodology, A.S. (Alexey Sulavko); software, I.P., A.S. (Alexey Sulavko) ; validation, A.V., A.S.(Alexey Sulavko), A.S. (Alexander Samotuga), P.S. and I.P.; formal analysis, A.S. (Alexey Sulavko); investigation, A.V. and A.S. (Alexey Sulavko) ; resources, A.V.; data curation, A.S. (Alexey Sulavko) and I.P.; writing—original draft preparation, A.S. (Alexey Sulavko), and

 doi:10.20944/preprints202601.0869.v1

I.P; writing—review and editing, A.S. (Alexander Samotuga), P.S.; visualization, I.P., A.S. (Alexey Sulavko) ; supervision, A.S. (Alexey Sulavko); project administration, A.S. (Alexey Sulavko); funding acquisition, A.S. (Alexey Sulavko). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the state assignment of Ministry of Science and Higher Education of the Russian Federation grant number theme No. FSGF-2023-0004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This study was conducted in accordance with the international principles of ethical research involving human subjects (Declaration of the Helsinki Association), as well as the requirements of the Federal Law of the Russian Federation No. 152-FZ of July 27, 2006 "On Personal Data" and Federal Law No. 572-FZ of December 29, 2022 "On the implementation of identification and (or) authentication of individuals using biometric personal data, on amendments to certain legislative acts of the Russian Federation and recognition of certain provisions of legislative acts of the Russian Federation as invalid". All experimental protocols were approved by the expert committee [Federal State Autonomous Educational Institution of Higher Education "Omsk State Technical University"] (expert opinion dated 09.09.2024)

Data Availability Statement: Two of the three datasets used in the experiments are available for distribution: 1. Faces94 dataset: Facial Images: Faces94 dataset page 2. LFW dataset: Labelled Faces in the Wild (LFW) dataset page. The third dataset SFDv1 was generated by the authors of the study. In accordance with the legislation of the Russian Federation (Federal Law No. 152 of July 27, 2006 "On Personal Data"), the distribution of personal data, including biometric data in the form of photo and video images of a person, without the written consent of the subject of personal data is prohibited. When conducting experiments, the subjects gave consent only to the use of personal data in an anonymized form for research purposes. For this reason, we cannot post this data in its original form in open sources. For more detailed clarification, you can ask the author of the article, Irina Evgenievna Panfilova, at the email address panfilova_2015@bk.ru.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jindal, A.K.; Chalamala, S.; Jami, S.K. Face template protection using deep convolutional neural network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2018.
2. Pandey, R.K.; Zhou, Y.; Kota, B.U.; Govindaraju, V. Deep secure encoding for face template protection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2016.
3. Abdullahi, S.M.; Sun, S.; Wang, B.; Wei, N.; Wang, H. Biometric template attacks and recent protection mechanisms: A survey. *Inf. Fusion* **2024**, *103*, 102144.
4. Yu, Z.; Qin, Y.; Li, X.; Zhao, C.; Lei, Z.; Zhao, G. Deep learning for face anti-spoofing: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5609–5631.
5. Terhörst, P.; Fährmann, D.; Damer, N.; Kirchbuchner, F.; Kuijper, A. Beyond identity: What information is stored in biometric face templates? **2020**.
6. Bassit, A.; Hahn, F.; Veldhuis, R.; Peter, A. Hybrid biometric template protection: Resolving the agony of choice between bloom filters and homomorphic encryption. *IET Biom.* **2022**, *11*, 430–444.
7. Manisha.; Kumar, N. Cancelable Biometrics: a comprehensive survey. *Artif. Intell. Rev.* **2020**, *53*, 3403–3446.
8. Lutsenko, M.; Kuznetsov, A.; Kiian, A.; Smirnov, O.; Kuznetsova, T. Biometric cryptosystems: Overview, state-of-the-art and perspective directions. In *Advances in Information and Communication Technology and Systems*; Lecture notes in networks and systems, Springer International Publishing: Cham, 2021; pp. 66–84.
9. Sulavko, A. Biometric-based key generation and user authentication using acoustic characteristics of the outer ear and a network of correlation neurons. *Sensors (Basel)* **2022**, *22*, 9551.
10. Akhmetov, B.; Ivanov, A.; Alimseitova, Z. Training of neural network biometry-code converters, 2018. Paper presented at the 3rd international symposium on the genetics of industrial microorganisms, University of Wisconsin, Madison, 4–9 June 1978.
11. Bogdanov, D.S.; Mironkin, V.O. Data recovery for a neural network-based biometric authentication scheme. *Matematiceskie voprosy kriptografii* **2019**, *10*, 61–74.

 doi:10.20944/preprints202601.0869.v1

12. Marshalko, G.B. On the security of a neural network-based biometric authentication scheme. *Matematiceskie voprosy kriptografii* **2014**, *5*, 87–98.
13. Vulfin, A.; Vasilyev, V.; Nikonov, A.; A.D., K. Neural network biometric cryptography system. *Proceedings of the Information Technologies and Intelligent Decision Making Systems (ITIDMS2021)* **2021**, *2843*.
14. Peng, J.; Yang, B.; Gupta, B.B.; Abd El-Latif, A.A. A biometric cryptosystem scheme based on random projection and neural network. *Soft Comput.* **2021**, *25*, 7657–7670.
15. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139.
16. Rathgeb, C.; Merkle, J.; Scholz, J.; Tams, B.; Nesterowicz, V. Deep face fuzzy vault: Implementation and performance. *Comput. Secur.* **2022**, *113*, 102539.
17. Gilkalaye, B.P.; Rattani, A.; Derakhshani, R. Euclidean-distance based fuzzy commitment scheme for biometric template security. In Proceedings of the 2019 7th International Workshop on Biometrics and Forensics (IWBF). IEEE, 2019.
18. Kuznetsov, O.; Zakharov, D.; Frontoni, E. Deep learning-based biometric cryptographic key generation with post-quantum security. *Multimed. Tools Appl.* **2023**, *83*, 56909–56938.
19. Dong, X.; Kim, S.; Jin, Z.; Hwang, J.Y.; Cho, S.; Teoh, A.B.J. Secure chaff-less fuzzy vault for face identification systems. *ACM Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 1–22.
20. Malygin, A.; Seilova, N.; Boskebeev, K.; Alimseitova, Z. Application of artificial neural networks for handwritten biometric images recognition. *Comput. Model. New Technol.* **2017**, *21*, 31–38.
21. Roy, N.D.; Biswas, A. Fast and robust retinal biometric key generation using deep neural nets. *Multimed. Tools Appl.* **2020**, *79*, 6823–6843.
22. Wang, P.; You, L.; Hu, G.; Hu, L.; Jian, Z.; Xing, C. Biometric key generation based on generated intervals and two-layer error correcting technique. *Pattern Recognit.* **2021**, *111*, 107733.
23. Talreja, V.; Valenti, M.C.; Nasrabadi, N.M. Zero-shot deep hashing and neural network based error correction for face template protection. In Proceedings of the 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, 2019.
24. Chen, L.; Zhao, G.; Zhou, J.; Ho, A.T.S.; Cheng, L.M. Face template protection using deep LDPC codes learning. *IET Biom.* **2019**, *8*, 190–197.
25. Lai, Y.L.; Hwang, J.Y.; Jin, Z.; Kim, S.; Cho, S.; Teoh, A.B.J. Symmetric keyring encryption scheme for biometric cryptosystem. *Inf. Sci. (Ny)* **2019**, *502*, 492–509.
26. Mai, G.; Cao, K.; Lan, X.; Yuen, P.C. SecureFace: Face Template Protection. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 262–277.
27. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503.
28. Hosna, A.; Merry, E.; Gyalmo, J.; Alom, Z.; Aung, Z.; Azim, M.A. Transfer learning: a friendly introduction. *J. Big Data* **2022**, *9*, 102.
29. Qi, X.; Zhang, L. Face recognition via centralized coordinate learning **2018**. [arXiv:cs.CV/1801.05678].
30. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. VGGFace2: A dataset for recognising faces across pose and age **2017**. [arXiv:cs.CV/1710.08092].
31. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.
32. Zhou, X.P.; Sun, M. Study on accuracy measure of trigonometric leveling. *Appl. Mech. Mater.* **2013**, *329*, 373–377.
33. Steane, A.M. Simple quantum error-correcting codes. *Phys. Rev. A* **1996**, *54*, 4741–4751.
34. Learned-Miller, E.; Huang, G.B.; RoyChowdhury, A.; Li, H.; Hua, G. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*; Springer International Publishing: Cham, 2016; pp. 189–248.
35. Sikder, J.; Chakma, R.; Chakma, R.J.; Das, U.K. Intelligent Face Detection and Recognition System. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT). IEEE, 2021.
36. Tams, B. Decodability attack against the fuzzy commitment scheme with public feature transforms **2014**. [arXiv:cs.CR/1406.1154].

 doi:10.20944/preprints202601.0869.v1

disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.