INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2016, VOL. 62, NO. 2, PP. 197–202
Manuscript received May 29, 2016; revised June, 2016. DOI: 10.1515/eletel-2016-0027

Flexible Neural Network Architecture for Handwritten Signatures Recognition

Dawid Połap, Marcin Woźniak

***Abstract*—This article illustrates modeling of flexible neural networks for handwritten signatures preprocessing. An input signature is interpolated to adjust inclination angle, than descriptor vector is composed. This information is preprocessed in proposed flexible neural network architecture, in which some neurons are becoming crucial for recognition and adapt to classification purposes. Experimental research results are compared in benchmark tests with classic approach to discuss efficiency of proposed solution.**

***Keywords*—Neural networks, handwritten signatures preprocessing, signature recognition, chebyshev polynomials.**

I. Introduction

Pattern analysis and classification methods are useful techniques of Computational Intelligence (CI) with various applications. One of them is classification of handwritten texts so important for identity control systems present i.e. financial institutions, branch institutions and other structures with remote documents verification systems. Efficient methods of knowledge aggregation and retrieval must be applied in distributed systems, where input data is processed on remote unit to verify if the input signature match the pattern. Various methods of CI help i.e. in case of missing or incomplete data [1], [2] and authorship semantical identification [3]. Natural Language Processing (NLP) techniques can be applied in prescription processing [4] and robot instructions composition from natural behavior [5]. Neural Networks (NN) are structures that can be efficiently applied in these types of systems because of ability to generalize knowledge for creative systems [6] and variety of developed architectures with new abilities for multi agent systems [7], more efficient memory [8] and other applications [9].

A. Related Works

Rapid technological development makes that technology encountered at every step in our lives. Consequently, the quality of security in the protection of digital data must increase in order not only to ensure that our data is safe, but our identity as well. Identity verification can be done by specific information about the person or verifications of certain features. The best feature set is our physiological side, for example, fingerprints, which are unique to each individual. Fingerprint sensors operate on the principle of finding certain curves and the local characteristics, as shown in [10] and [11]. In a similar way, the verification based on the iris of the eye works - local characteristics and its shape are analyzed.

Fingerprints and iris of the eye are only two methods, which are quite rare due to numerous drawbacks - technological and financial. Much simpler and more popular method of verification is a signature, which is used by everyone. In banking, the signature is required in every document as proof of its validity. Courier companies and post offices use electronic or manual signature as a form of acknowledgment of receipt of the package or letter. In the past few years, with all the known methods of verification, signature verification is gaining the most popularity.

The verification of the signature is divided into two groups: off–line and on–line. The first group is a signature verification on the basis of the relevant pictures or scan of the document. In this reasoning, any noise is removed and the curve representing the signature is analyzed. Such methods allow to study graphology or confirm the validity of the signature after some time. Such methods rely on finding and comparing the specific characteristics of the curve, for example, rounding or distortion [12]. The second group are on–line signatures, signatures executed at the time of analysis. In addition to the analysis of the curve and its features, other features are taken into account, such as pen pressure, its angle of inclination, typing speed, or even the time between lifting the pen. Methods of extracting data from the signature performed alive are many and most of them are constantly improved in order to increase precision. One such method is the use of a vertical partitioning curve [13] or length normalization by using up–sampling and down–sampling [14]. Another approach is shown in [15], where the extraction of features based on gray level, the size and radian of signatures, and in [16] the authors introduced the use of a statistical approach to the subject. Extracted features from the signature must be processed by a classifier for the purpose of recognition.

The most commonly used classifiers among others are statistical analysis and neural networks. In [17] the idea of saving the features of the signature as a vector for training neural networks is shown, again in [18] the use of probabilistic neural networks in conjunction with the hybrid methods of discrete random transform, principal component analysis has been described and tested. A major problem in the application of neural networks is the number of samples - the network is more precise the more samples are used in the training process. The situation when the boss asks his employees to create hundreds of signatures is unimaginable. For the purposes of

Authors acknowledge contribution to this project of Operational Programme: "Knowledge, Education, Development" financed by the European Social Fund under grant application POWR.03.03.00-00-P001/15.

Dawid Połap and Marcin Woźniak are with Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland, (e-mail: Dawid.Polap@gmail.com, Marcin.Wozniak@polsl.pl).

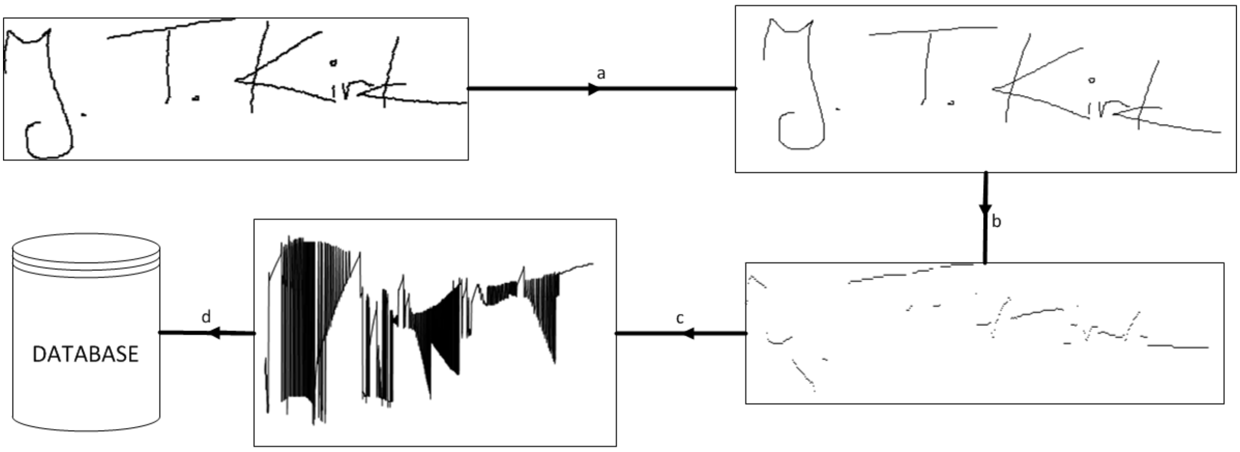

Fig. 1. A simplified model of the preprocessing of the signature sample: a) potential rotation of the signature; b) simplification of the curve – for each vertical line, the average point is found; c) application of Chebyshev interpolation for a curve; d) adding the reduced sample to the database.

the implementation of the application with neural network, it is necessary to generate artificial samples. This process should create samples different from each other, but preserving as much as possible features of a signature. The possibility of identity verification based on synthetically generated static data is presented in [19]. Another method is the possibility of creating samples using a heuristic algorithm [20]. One more important aspect for NN is training process, where motivation can boost NN for faster adaptation [21] and [22]. For the purpose of reducing the time to learn a new neural network architectures are created like self adapting which enable stable man–machine interactions [23], [24]. An alternative to neural networks is the use of dynamic analysis of the signature [25] or local stability analysis [26].

In this work, we present an alternative method for more flexible verification of existing samples. For this purpose we propose a flexible architecture of the neural network as a classifier of samples created by using interpolation method to simplify the curve representing the signature.

II. Handwritten Curve Transformation

User signature verification is a complex problem. We all have different handwriting style. Moreover the signature can be deformed if we sign in a hurry or for some reasons twitch. In electronic systems users are signing within provided space where location or rotation of the input curve can cause some problems to DSS. However user should be able to write freely and developed solution shall be responsible to recognize input curve properly and if necessary cut, rotate and resize it for recognition purposes. In proposed solution we are transforming signature curve from 530×270 pixels input into 40×15 pixels processed objects. We can define transformation as an interpolation between input and processed object, where handwritten curve becomes interpolated.

In a first step input curve (the signature) is simplified by determining average points – in each vertical row. Points are found and for each coordinate arithmetic mean is calculated. In the second step, all points are interpolated by Chebyshev method.

A. Chebyshev Method

Transformation of input signature into simplified object uses recursively created Chebyshev polynomials of the first kind

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x). \tag{1}$$

For the benchmark tests we used first five polynomials: $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$ and $T_4(x) = 8x^4 - 8x^2 + 1$ to compose an interpolating function $\varphi(x)$

$$\varphi(x) = \frac{1}{2}c_0 + \sum_{j=1}^{m=4} c_j T_j. \tag{2}$$

where discrete input coefficients c_j are calculated according to

$$c_j = \frac{2}{m+1}\sum_{j=0}^{m=4} \varphi(x_k)T_j(x_k). \tag{3}$$

This transforms input signature into 40×15 pixels object forwarded to adjustable neural network.

III. Flexible Neural Network Architecture

Flexible neural network architecture is composed with assumption that some of the neurons are more important for classification purposes. Therefore these units shall be given a priority in recognition. For this reason we have introduced an impact factor that is assigned to each unit. On the network we process transformed objects that are composed of $n = 40 \times 15$ pixels assigned to each of network input. This is represented in a matrix $IM_{n-2\times n}$

$$\begin{bmatrix} \text{input}_1 \\ \text{input}_2 \\ \dots \\ \text{input}_n \end{bmatrix} \rightarrow \begin{bmatrix} \text{Im}_1^1 & \text{Im}_1^2 & \dots & \text{Im}_1^{n-2} \\ \text{Im}_2^1 & \text{Im}_2^2 & \dots & \text{Im}_2^{n-2} \\ \dots & & & \\ \text{Im}_n^1 & \text{Im}_n^2 & \dots & \text{Im}_n^{n-2} \end{bmatrix} \rightarrow \begin{bmatrix} \text{out}_1 \\ \text{out}_2 \\ \dots \\ \text{out}_m \end{bmatrix} \tag{4}$$

Fig. 2. Sample signatures processed for proposed neural classifier. We can see an original handwritten curve with its interpolation by the Chebyshev method that is forwarded to the flexible neural network architecture as an input object.

where impact coefficients are

$$\mathrm{Im}_i^k = \frac{\sum_{i=1}^{n} \mu_i^k (f_i^k (\sum_{j=1}^{n} (w_{ij}^k \cdot x_j^k)))}{\sum_{l=1}^{n} (\sigma_{max}^{(l)} + \sigma_{min}^{(l)})}. \tag{5}$$

Impact of classification for each unit is calculated according to

$$\mu_i^k(y) = exp\left[\frac{-(y-r)^2}{2c^2}\right], \tag{6}$$

where

$$c = \max(\sigma_x) + \min(\sigma_x), \tag{7}$$

and

$$r = \max(\sigma_x) - \min(\sigma_x). \tag{8}$$

Each of the classification impact factors $\mathrm{Im}_n^n \in (\sigma_{min}^{(n)}, \sigma_{max}^{(n)})$ is measured as statistical distribution spread of points in input object, where we measure it for upper and lower values of each signature. This is stored as a matrix

$$\begin{bmatrix} \sigma_{min}^{(1)} & \sigma_{max}^{(1)} \\ \sigma_{min}^{(2)} & \sigma_{max}^{(2)} \\ \dots & \\ \sigma_{min}^{(n)} & \sigma_{max}^{(n)} \end{bmatrix} \tag{9}$$

presented for NN in training process as additional knowledge about each user. The input signal to the neuron n is multiplied by $\sigma_{min}^{(n)}$ and the output by $\sigma_{max}^{(n)}$ what can be presented as

$$input_n = \sigma_{min}^{(n)} \sum_{i=1}^{n} w_i x_i \tag{10}$$

and

$$output_n = \sigma_{max}^{(n)} f(input), \tag{11}$$

where w_i is the weight on the connection between neurons i and n, x_i is the output value from neuron i, and f is the activation function.

A. Training Process

Flexible neural network structure is trained to recognize the interpolated signatures. Algorithm 1 presents training operations. Adaptive neural network learns decreasing recognition error. In the training process we measure it for the output according to

$$\Delta_K \leftarrow \frac{1 - output_k}{expected_k - output_k}, \tag{12}$$

and for neurons in hidden layers

$$\Delta_k \leftarrow \frac{1 - output_k}{\sum_{l \in outputs} w_{lk} \Delta_k}. \tag{13}$$

These values are applied to update weights for each i-th input

$$w_i \leftarrow w_i + \Delta_k w_i. \tag{14}$$

Algorithm 1 Flexible Neural Network Training Process

```
Start
Define the activation function, learning coefficient, error_value and threshold value
Load inputs vectors
while global_error < error_value do
  Propagating inputs into forward
  for all layers do
    for all neurons do
      Calculate the sum of the weights entering to the neuron
      Add threshold value to calculated sum
      Calculate the activation function for the neuron
      Calculate impact value
    end for
    Backward error propagation
    for all neuron in output layer do
      Calculate global_error
      for all neuron in output layer do
        Calculate error
        Update weights
      end for
    end for
  end for
end while
Stop
```

IV. Benchmark Tests

In experiments we have verified efficiency of newly proposed flexible neural network architecture. 400 samples of original signatures were created (200 signatures for two people). Then, for each of them an additional 200 forged signatures created. For the input signature (some of them are presented in Fig. 2) we implemented transformation using Chebyshev polynomials. Results were divided into training set (75% of signatures) and verification set (25% of signatures).

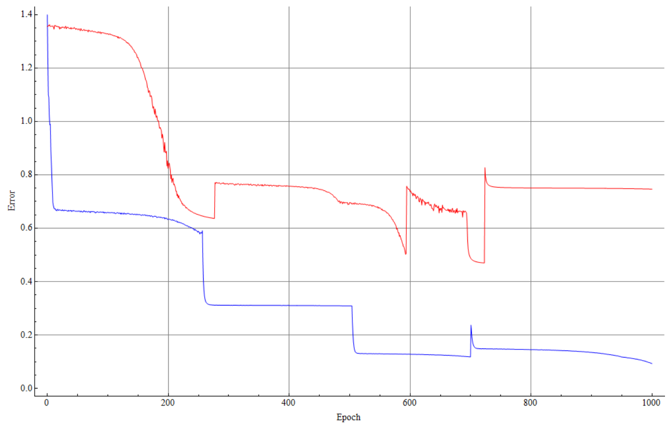

Fig. 3. Minimization of error in training of the proposed structure of the network (blue line) in contrast with the classical neural network training process without impact coefficients (red line).

Classical approach without impact coefficients and adaptive architecture (according to Algorithm 1) were trained to recognize signatures. Sigmoid function was set as the activation function, a threshold value was initiated as 0.3 and the learning coefficient was 0.4.

The training process of the neural network is shown in Fig. 3. Training of the proposed network architecture is almost always smoothly minimized in contrast to the classical network where drastic jumps occur by increasing the value of the error. Moreover, both the networks were trained to obtain an error equal to 0.1. For flexible network that error was obtained for 1000 iteration, and for classical network, obtaining such an error was possible in the billionth iteration. For such a trained neural networks, their effectiveness has been tested by obtaining a result of the network for each of verification sample. In Fig. 4 and Fig. 5 we present results of sample tests for two implemented neural architecture. Fig. 4 presents verification over 30 randomly selected signatures from verification set by the use of classic neural network approach without flexibility coefficient. Fig. 5 presents sample classification result over 30 randomly selected signatures by the use of proposed flexible architecture. In the presented benchmark tests classic approach has given a correct classification at the level of about 65%, while proposed flexible approach has given correct classification at the level of about 85%. This means the application of flexibility coefficient improves training process and therefore enables neural network to classify input signatures about 20% better.

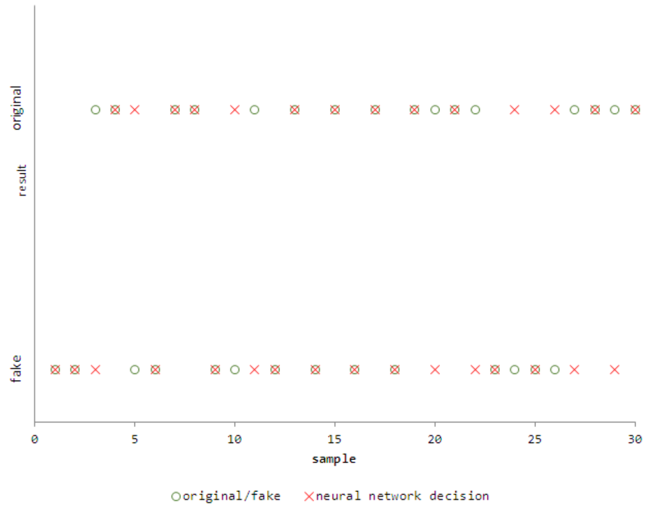

Fig. 4. Test 1: Sample verification process by classic neural network without flexibility coefficients performed over probe containing 30 randomly selected signatures from verification set.

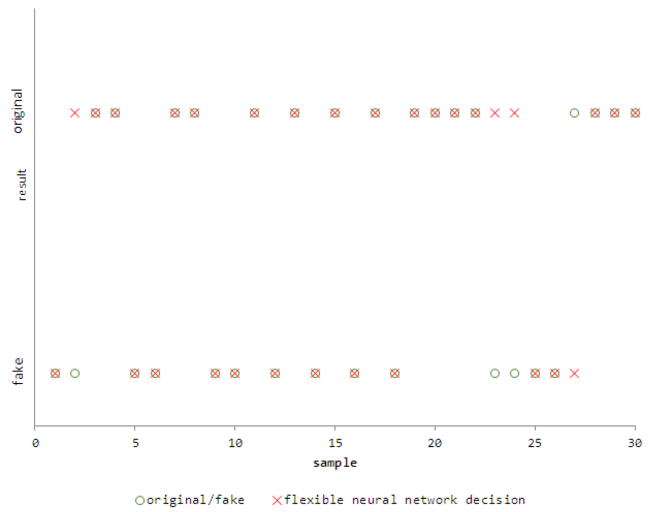

Fig. 5. Test 2: Sample verification process by flexible neural network performed over probe containing 30 randomly selected signatures from verification set.

TABLE I
VALIDATION RESULTS OF 200 SAMPLES OF THE ORIGINAL SIGNATURES AND 200 SAMPLES OF FORGED SIGNATURES

Architecture	Sample	Classified	Validation Rate
classical	Orginal	132	76 %
	Fake	101	60,5 %
flexible	Orginal	186	93 %
	Fake	179	86,5 %

The averaged results of the validation are presented in Table I. Based on the obtained results, the average efficiency of classical network amounted to 58.25% and flexible network to 89.75%. In Fig. 6, the effectiveness of verification for specific samples for flexible neural network is shown.

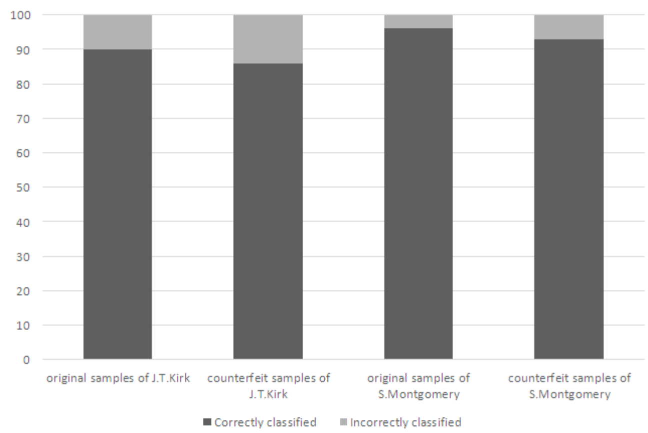

Fig. 6. The correctness of classified samples by the proposed structure of the neural network.

A. Conclusions

In user verification systems various devices have different input appliance and resolution. Moreover we can expect different languages and various handwriting styles. Therefore applied solution must be efficient enough to distinguish real signatures among fake ones. Since human signature is mathematically a continuous or broken curve we can apply mathematic method to transform it. For this reason we have proposed transformation technique that interpolates input signature into tailored object for flexible neural network DSS. Proposed neural network structure adapts in training process to recognition purposes by giving some of the neurons higher importance factors. These units are therefore deciding on the authentication of the input signature.

Proposed architecture has shown higher precision in performed benchmark tests, where in comparison to classic approach we achieved 17% increase in precision for original signature and 26% for fake signatures. Introduction of impact coefficients influenced training process by helping on faster convergence to set error value. Therefore presented experimental research results show that proposed solution can be efficiently introduced to user verification systems based on signature processing. For further work we plan to introduce fuzzy measures of importance instead of factors. These will lead to more flexible recognition that will adapt to different signatures with better accuracy.

V. Final remarks

Efficient methods of user verification are necessary in growing digitalization of various aspects of everyday life as well as new issues in offices and agencies. Parallel to new technology that is giving new possibilities a need for new and improved methods and algorithms is visible. Therefore in this article we proposed new approach to develop flexible constructions of neural networks that can assist in validation purposes. Proposed structure is assuming flexibility to classified inputs in assigning coefficients to neural units. Therefore some of them are becoming more important and gain priority in decision making. This is very useful in situations where some parts of the data are more important than the others. In the presented examples where some parts of the signatures are not possible to fake, therefore proposed classifier takes advantage of the flexible construction to make classification of these parts a crucial for final decision.

References

[1] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, “Decision trees for mining data streams based on the gaussian approximation,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 1, pp. 108–119, 2014.

[2] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, “A new method for data stream mining based on the misclassification error,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 5, pp. 1048–1059, 2015.

[3] A. Venčkauskas, R. Damaševičius, R. Marcinkevičius, and A. Karpavičius, “Problems of authorship identification of the national language electronic discourse,” in *Information and Software Technologies*. Springer, 2015, pp. 415–432.

[4] D. S. Carrell, D. Cronkite, R. E. Palmer, K. Saunders, D. E. Gross, E. T. Masters, T. R. Hylan, and M. Von Korff, “Using natural language processing to identify problem usage of prescription opioids,” *International journal of medical informatics*, vol. 84, no. 12, pp. 1057–1064, 2015.

[5] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz, “Inferring maps and behaviors from natural language instructions,” in *Experimental Robotics*. Springer, 2016, pp. 373–388.

[6] A. Horzyk, “How does generalization and creativity come into being in neural associative systems and how does it form human-like knowledge?” *Neurocomputing*, vol. 144, pp. 238–257, 2014.

[7] J. A. Starzyk *et al.*, “Memristor crossbar architecture for synchronous neural networks,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 8, pp. 2390–2401, 2014.

[8] V. A. Nguyen, J. A. Starzyk, W.-B. Goh, and D. Jachyra, “Neural network structure for spatio-temporal long-term memory,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 971–983, 2012.

[9] A. Horzyk, “Innovative types and abilities of neural networks based on associative mechanisms and a new associative model of neurons,” in *Artificial Intelligence and Soft Computing*. Springer, 2015, pp. 26–38.

[10] A. Muñoz-Briseño, A. Gago-Alonso, and J. Hernández-Palancar, “Fingerprint indexing with bad quality areas,” *Expert Systems with Applications*, vol. 40, no. 5, pp. 1839–1846, 2013.

[11] H. Kasban, “Fingerprints verification based on their spectrum,” *Neurocomputing*, vol. 171, pp. 910–920, 2016.

[12] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Writer-independent feature learning for offline signature verification using deep convolutional neural networks,” *arXiv preprint arXiv:1604.00974*, 2016.

[13] K. Cpałka and M. Zalasiński, “On-line signature verification using vertical signature partitioning,” *Expert Systems with Applications*, vol. 41, no. 9, pp. 4170–4180, 2014.

[14] F. L. Malallah, S. M. S. Ahmad, W. A. W. Adnan, O. A. Arigbabu, V. Iranmanesh, and S. Yussof, “Online handwritten signature recognition by length normalization using up-sampling and down-sampling,” *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 4, no. 1, pp. 302–313, 2015.

[15] X. Chen, “Extraction and analysis of the width, gray scale and radian in chinese signature handwriting,” *Forensic science international*, vol. 255, pp. 123–132, 2015.

[16] S.-J. Park, S.-J. Hwang, J.-P. Na, and J.-H. Baek, “On-line signature recognition using statistical feature based artificial neural network,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 19, no. 1, pp. 106–112, 2015.

[17] S. Sthapak, M. Khopade, and C. Kashid, “Artificial neural network based signature recognition & verification,” *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, vol. 2, no. 8, pp. 191–197, 2013.

[18] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang, and B. Y. Hiew, “Image-based handwritten signature verification using hybrid methods of discrete radon transform, principal component analysis and probabilistic neural network,” *Applied Soft Computing*, vol. 40, pp. 274–282, 2016.

[19] J. Galbally, M. Diaz-Cabrera, M. A. Ferrer, M. Gomez-Barrero, A. Morales, and J. Fierrez, “On-line signature recognition through the combination of real dynamic data and synthetically generated static data,” *Pattern Recognition*, vol. 48, no. 9, pp. 2921–2934, 2015.

[20] M. Woźniak, D. Połap, G. Borowik, and C. Napoli, “A first attempt to cloud-based user verification in distributed system,” in *Computer Aided System Engineering (APCASE), 2015 Asia-Pacific Conference on*. IEEE, 2015, pp. 226–231.

[21] J. A. Starzyk, J. T. Graham, P. Raif, and A.-H. Tan, “Motivated learning for the development of autonomous systems,” *Cognitive Systems Research*, vol. 14, no. 1, pp. 10–25, 2012.

[22] J. Graham, J. A. Starzyk, and D. Jachyra, “Opportunistic behavior in motivated learning agents,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 8, pp. 1735–1746, 2015.

[23] K. Waledzik and J. Mandziuk, “An automatically generated evaluation function in general game playing,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 6, no. 3, pp. 258–270, 2014.

[24] M. Swiechowski and J. Mandziuk, “Self-adaptation of playing strategies in general game playing,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 6, no. 4, pp. 367–381, 2014.

[25] K. Cpałka, M. Zalasiński, and L. Rutkowski, “A new algorithm for identity verification based on the analysis of a handwritten dynamic signature,” *Applied Soft Computing*, vol. 43, pp. 47–56, 2016.

[26] G. Pirlo, V. Cuccovillo, M. Diaz-Cabrera, D. Impedovo, and P. Mignone, “Multidomain verification of dynamic signatures using local stability analysis,” *Human-Machine Systems, IEEE Transactions on*, vol. 45, no. 6, pp. 805–810, 2015.