Imitation model for keystroke dynamics base on state context representation

Dmitry V. Pashchenko¹, Alexey I. Martyshkin², Dmitry A. Trokoz³, Elena A. Balzannikova⁴

¹Penza State Technological University, Penza, Russia, dmitry.pashchenko@gmail.com
²Penza State Technological University, Department of computing machines and systems, Penza, Russia, alexey314@yandex.ru
³Penza State Technological University, Department of computing machines and systems, Penza, Russia dmitriy.trokoz@gmail.com
⁴Penza State Technological University, Research Department, Penza, Russia, elenabalzannikova@gmail.com

Abstract. This paper describes different methods of keystroke dynamics. As part of a study on the development of methods for the dynamic analysis of keystroke dynamics, a method is proposed for representing the process of typing on the keyboard in the form of a successive change of states. Based on states it is proposed to define certain sequences called the state context as feature of a biometric image. Based on this representation, an imitation model of keystroke dynamics was developed, which allows replacing the physical source of biometric data for the purpose of testing and debugging the identification system of keystroke dynamics.

Keywords: keystroke dynamics, biometry, imitation model

1. Introduction

The development of modern information and Internet technologies places increasing demands on security systems that perform a number of critical functions:

- preventing unauthorized access to information and systems;
- Prevention of illegal actions on behalf of another user;
- Prevention of the illegal distribution of confidential or personal information

Besides, often to provide security, it is necessary to establish the authorship of a particular electronic document.

Today, to implement these functions, a variety of methods are used based on the possession of secret information (password or PIN), ownership of an attribute (for example, a smart card), using of biometric data for authentication, or using an electronic digital signature to establish authorship.

Of greatest interest in this area are biometric methods, because they provide a number of advantages: the data source is inseparable from the user, biometric data is much more difficult to steal or forge. In addition, a distinctive feature of biometric methods is the ability to dynamically
identify the user, thereby revealing the substitution of the operator. Other authentication methods (knowledge or attribute) require a single identification only at the time of user login. However, not all biometric methods can provide continuous identification. Methods such as fingerprint scanning, iris scanning or voice recognition are difficult to apply for these purposes. To implement this function, it would be possible to use user recognition by face geometry using a web camera located opposite the monitor, but standard input devices of a personal computer are more suitable for these purposes: a keyboard and a mouse. These identification methods provide a number of additional advantages: the lack of additional equipment and the possibility of a hidden identification procedure.

2. The concept of keystroke dynamics

2.1 Survey
Among the methods of analyzing Keystroke dynamics, there are two main areas: static and dynamic analysis.

Static analysis involves an identification procedure by entering a predetermined sequence of characters: a passphrase. A typical area of application for this type of analysis is the user authorization procedure based on entering the system by using a password, which, in combination with the keystroke dynamics analysis, improves the system security in case of compromise. Building the keystroke dynamics image for each user requires several samples of entering a passphrase, which, basically, does not take much time.

Dynamic analysis allows you to identify the user regardless of the text they type. This approach allows not only to identify the user during login, but also to perform continuous analysis of the Keystroke dynamics during the entire session, in order to recognize the substitution of the active user. In addition, this method allows not to use predefined password while the identification procedure, but to enter a small random text, saving the user and the system from the need to store passwords. This type of analysis requires a longer period of training the system to build the most complete and accurate biometric image.

In addition, it should be noted that such systems can be used both on the corporate network or the Internet, where many users are registered and a large number of biometric images are generated, and for individual use to protect personal information. In the first case, the task of the system consists mainly in classifying the keystroke dynamics image among registered users. To train classifier for a new user can be used existing biometric images as images of “foreign”. Thus, the larger the database of images, the more data is provided to the system for training. In turn, if there is only one registered user, there is no database of “foreign” images. In this case, the task is to assess the similarity measure of the received and stored biometric images and determine the confidence interval, according to which an authentication or identification decision is made.

The main purpose of the research is developing a keystroke dynamics analysis system, based on such methods that support the identification procedure for both a single and a network user. To do this, it is necessary to consider the concept of keystroke dynamics and determine its main characteristics.

2.2 Keystroke dynamics characteristics
The basic metrics for entering text on the keyboard are timestamps of two types of events: pressing a key and releasing a key. Let’s denote \( R_t(k) \) as the key release event at time \( t \) and \( P_t(k) \) as the key release event at time \( t \).
Based on these events, a number of features can be derived that can be the basis for the analysis of Keystroke dynamics. The primary features are time intervals between a pair of events of various types. Such intervals include: key holding time, delay, and time between keystrokes.

The holding time of the i key can be defined as the interval between releasing and pressing the same key:

$$DT_i = R_i(t) - P_i(t)$$

The interval between keystrokes is defined as the interval between consecutive events of keystrokes:

$$FT_i = P_{i+1}(t) - P_i(t)$$

Next, let’s define the delay as the interval between releasing the previous key and pressing the next one:

$$LT_i = P_{i+1}(t) - R_i(t)$$

In the case of intersection of the key holding time, this feature will have a negative value. The described events and primary features can be represented graphically, as shown in Figure 1.

![Figure 1 Primary features of keystroke dynamics](image)

Based on a combination of the considered primary features, composite features can be formed: digraphs, trigraphs, etc. The specific form of a composite feature depends on the method used to analyze keystroke dynamics [1].

3. Background research

One of the most common areas of behavioral biometrics is the analysis of keystroke dynamics. The issues of keystroke dynamics analysis were first considered in the works of R. Joyce and G. Gupta in 1990, who used the method of estimating the intervals between keystrokes when entering a fixed passphrase and comparing with a reference sample [2].

Today, there are many studies in the field of analysis of Keystroke dynamics, both among Russian and among foreign scientists[3]. Among Russian developments, the works of R.V. Yeruslanov, Yu.A. Bryukhomitsky, M.N. Kazarin, who in their works use various methods of analysis of keystroke dynamics: poly-Gaussian distribution, estimation of probability density, Haar decomposition[4, 5, 6, 7].

In addition, the research of I.G. Sidorkina and A.N. Savinov, in which the separation of the sample into features with and without overlapping, is applied using the calculation of the Euclidean measure for features and the poly-Gaussian distribution [8].

In the works of D.S. Krutokhvostov and V.E. Khitsenko suggested frequency histograms of using alternative keys and other features based on chi-square statistics. The empirical distribution functions of the legal and analyzed user are compared (by key holding time) based on Kolmogorov-Smirnov statistics[9].
Among foreign studies, one can note P. Borus, S. Mondal, Syed Idrus, Syed Zulkarnain, H. Barghouthi, who use many methods in their work, among which are [10, 11, 12, 13]:

- Weighted Euclidean distance estimate for a single key;
- A weighted combination of algorithms (for individual keys): ANN, CPANN and SVM. Weights are optimized by the genetic algorithm
- An algorithm for digraphs based on two distance metrics: pairwise weighted Euclidean and correlation distance.

In the works of A. Bhatia, M. Hanmandl, keystroke dynamics is analyzed by entering a passphrase using methods: SVM, a random forest classifier for transformed characters [14].

It is also worth noting the work of Sampath K. Vuyyuru, Vir V. Phoha, Shrijit S. Joshi, Shashi Phoha, Asok Ray, which use the hidden Markov model apparatus for continuous analysis of keystroke dynamics [15].

4. Statement of the problem

The aim of the research is to develop a system of dynamic user identification by keystroke dynamics that implements a client-server architecture. The system should perform user registration, creating a biometric image of keystroke dynamics, and implement dynamic user identification according to the temporal characteristics of typing on the keyboard during the entire session [16].

Learning and debugging methods require a large amount of test data, what basically requires the presence of user. For long-term practical tests it will be extremely difficult to ensure the continual presence of users on whom the recognition system will be debugged. Based on this, it is proposed to use the method of simulation of a biometric image of a Keystroke dynamics based on one or more test samples of user input. The resulting image can then be used to simulate the input of real user data for modeling the system of keystroke dynamics analysis. In addition, in the process of simulation, it is possible to perform the process of debugging the system by adding distortions or noise into the original model to simulate various external factors, which were considered earlier.

Thus, the data approach allows you to generate images of “foreign” to simulate attempts to log in an unauthorized user or substitute an operator.

4.1 System model

A keystroke dynamics analysis system consists of several components. The collection component intercepts keyboard events and transfers them to the preprocessing unit, which converts keyboard events and generates features. This component transfers the generated features to the server component, which builds the image of the keystroke dynamics, and trains the user identification model [17, 18].

Schematically, the proposed user identification system can be represented, as shown in Figure 2.

The data source of the collection component is the keyboard. The simulation model, in turn, generates events that are identical to this component, replacing it. In order for the component of the simulation model to generate events with time characteristics corresponding to a specific user, a pre-received sample of input from this user should be used as the initial data for the model [19]. Figure 3 represents a system model in which a collection component is replaced by a simulation model:
Figure 2 Model of keystroke dynamics analysis system

Figure 3 Model of keystroke dynamics analysis system in which a collection component is replaced by a simulation model
Let’s consider the methods used to represent events, features, and the biometric image of keystroke dynamics in this system.

5. The concept of state context

In contrast to passphrase based identification in which the sequence of events is predefined, continuous analysis involves the identification based on a sequence of events that occur in an arbitrary way. To solve this problem, it is necessary to define a basic feature consisting of one or more consecutive events, on which the keystroke analysis will be based on. The events of pressing two keys are considered to be a basic sequence. Thus, the basic sequence consists of four events: P₁, R₁, P₂, R₂. The sequence of these events may vary. There are the possible options for following these events, which correspond to different ways of pressing a pair of keys.

The simplest sequence model assumes that there is no intersection of the key holding interval: the keys are pressed one after the other. Graphically, this model can be represented as shown in Figure 4. Thus, the sequence of events for the presented model will be as follows: P₁ → R₁ → P₂ → R₂. This model is typical for most cases of pressing symbol keys.

![Figure 4 Sequence model without intersection](image)

Other considered models suggest the intersection of the key holding interval. In one case, pressing the second key occurs while holding the first, and releasing after releasing the first. We designate this model as a model with partial intersection. Figure 5 shows a graphical representation of this process. The sequence of events will be as follows: P₁ → P₂ → R₁ → R₂. This sequence is often found when using utility keys, such as Alt + Shift.

![Figure 5 Sequence model with partial intersection](image)

Another option for intersecting intervals is to completely overlap with the holding interval of the first key of the holding interval of the second. Graphically, this process can be represented as shown in Figure 6. The sequence of events for this model will be as follows: P₂ → P₂ → R₂ → R₁. A typical example is the typing of an extra key character using the Shift key.
Thus, to form the most complete image of the keyboard, it is proposed to define these models and their temporal characteristics. To solve this task, the following model for the presentation of a biometric image of keystroke dynamics is proposed.

5.1 Keyboard state Representation Format

To simplify the keyboard model, represent it as a fixed set of keys, each of which at any given time can be in one of two states: pressed or released. Depending on the specific keyboard, the number of keys may vary. To build a model, we restrict the set of keys to only those that, as a rule, are involved in typing. Such keys include: symbolic and numeric keys (47 keys of the main keyboard), as well as service keys: Space, Enter, Right Ctrl, Left Ctrl, Right Shift, Left Shift, Right Alt, Left Alt, Caps Lock, Tab, Back Space. Thus, the keyboard model will contain 58 keys, based on which the time characteristics will be evaluated.

Represent the state of the keyboard as a vector, each element of which contains the state of a separate key. Set of states, based on the proposed model, consists of two elements:

\[ \text{State} = \{\text{Pressed}, \text{Released}\} \]

Therefore, the keyboard state vector will look like this:

\[ \text{Keyboard} = [\text{State}_1, \text{State}_2, ..., \text{State}_i, ..., \text{State}_{58}] \]

where \( i \) is the key number.

Since the proposed model has 58 elements, each of which can be in one of two states, the total number of possible states of the model is \( 2^{58} \), which is more than a quadrillion states. Obviously, processing such a number of states is practically impossible and the vast majority of states will never occur. Therefore, to simplify the model and reduce the number of possible states, we introduce a limit on the number of keys that can be in pressed state by two. This restriction allows you to almost completely describe the process of typing text on the keyboard. An exception is the use of control combinations, which involve the simultaneous pressing of three or more keys. These combinations can be excluded from the model or considered separately as additional features. Thus, taking into account the described limitation, the number of possible states can be calculated as one initial state when all keys are released, the number of possible states when one key is pressed and the number of states with two simultaneously pressed keys:

\[ 1 + \binom{58}{2} + \binom{56}{2} = 1712, \]

which is a more acceptable amount for building a model of keystroke dynamics.

Thus, this approach allows us to present the process of typing on the keyboard not as a sequence of events of pressing or releasing an arbitrary key, but as a sequential change in the state of the keyboard, presented in the form of fixed size vectors, each of which is associated with a time stamp. The time intervals between two consecutive states are features on the basis of which a biometric image of keyboard writing can be built.
However, in this paper, a different approach is proposed that would allow one to take into account the described models of keystroke sequences. For example, consider a keyboard model that consists of only two keys: A and B. Based on this, we present the state matrix for this model, which is shown in Table I. Columns A and B indicate the state of the key: 0 - released, 1 - pressed. The set of states of individual keys is designated as state $S_i$.

Table 1 Matrix of states for the keyboard model consisting of two keys

|   | A | B | State |
|---|---|---|-------|
| 0 | 0 |   | $S_0$ |
| 0 | 1 |   | $S_1$ |
| 1 | 0 |   | $S_2$ |
| 1 | 1 |   | $S_3$ |

In this case, the sequence of states for two consecutive keystrokes without intersections will be as follows: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_0 \xrightarrow{\Delta t_3} S_1 \xrightarrow{\Delta t_4} S_0$. Graphically, this process can be represented as shown in Figure 7.

For the second model with incomplete overlap, the sequence will have the following form: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_3 \xrightarrow{\Delta t_3} S_1 \xrightarrow{\Delta t_4} S_0$. Graphically, this process is presented in Figure 8.

Sequence of states for full intersection model will look as: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_3 \xrightarrow{\Delta t_3} S_2 \xrightarrow{\Delta t_4} S_0$, what shown in Figure 9.
5.2 The concept of state context

In each of the considered models, the initial and final states $S_0$ coincide, and the maximum number of states does not exceed 4. Based on this, it is proposed to determine the context of the state chain, which contains 4 consecutive states, between which three time intervals are calculated. Thus, the composite feature of the keyboard handwriting will have the following form:

$$c_{ijkl} = \{S_i, S_j, S_k, S_l; \{\Delta t_{ij}, \Delta t_{kj}, \Delta t_{kl}\}\},$$

where $i, j, k, l$ are the state codes of the keyboard.

This approach will allow not only extracting the primary features of keystroke dynamics, but also to separate them depending on the model of state sequence that were considered earlier. It is assumed that this method allows you to build the most complete and accurate model of the keyboard handwriting of users [13].

5.3 Initiation model

As described earlier, according to the developed system architecture, the collection component captures keyboard events and transforms the received information to the set of states, calculating the time interval between them. Based on this, the simulation model should generate a sequence of states with parameters identical to user parameters.

The basic requirements for the simulation model are:

- The sequence of states must be random;
- The vectors of two consecutive states must differ from each other by one bit.

The first requirement ensures the independence of the model from the generated text, which allows modeling a set of sequences of arbitrary length.

The second requirement means that only one event has occurred between two consecutive states: the key is pressed or released, which corresponds to the proposed model of keystroke dynamics based on the state contexts presentation.

In order for the simulation model to repeat the temporal characteristics between keyboard events and intersection model of keystrokes, it is proposed to use the keystroke dynamics representation based on the state contexts described earlier.

We denote $S = [S_0, S_1, S_2, ...]$ as the state vector of unlimited length, where $i$ is the index of the state in the generated sequence. Upon transition to the next state, an output signal is generated that corresponds to the transition time between states. The sequences of states, as well as time intervals, are determined by the parameters of a pre-formed set of contexts of a particular user.
The initial data for the imitation model are formed on a test sample of user text input. From this sample a sequence of states is extracted, and contexts are formed. According to the above definition, the context consists of four successive states, between which the time interval is calculated in the process of user typing text. For contexts with the same sequence of states, the mathematical expectation and standard deviation for each transition are calculated.

Thus, from a set of contexts $C$ for $C_{wxyz} = \{[s_w, s_x, s_y, s_z]; [\Delta t_{w}, \Delta t_{x}, \Delta t_{y}, \Delta t_{z}]\}$, a descriptor $\mathcal{C}$ is formed as follows:

$$\mathcal{C}_{wxyz} = \{[s_0^w, s_1^x, s_2^y, s_3^z]; [\mu_0^w, \sigma_0^w]; [\mu_1^x, \sigma_1^x]; [\mu_2^y, \sigma_2^y]\}$$

The initial state is the state of the keyboard on which no key is pressed:

$S_1 = 0, i = 0$

Next, from the set of descriptors, $\mathcal{C}_{ijkl}$, is randomly selected, in which the first state $s_0^w$ coincides with $S_0$. Thus, $S_{i+1} = s_0^w, \exists \mathcal{C}_{wxyz}: S_0^w = 0$. Subsequent generated states correspond to the internal states of the current context. At the initial step, the internal state index $k$, which is initially assigned the value 1. The subsequent event is formed depending on the current internal state index according to the following rules:

$$S_{i+1} := \begin{cases} s_2^i; k := 2, k = 1 \\ s_3^i; k := 3, k = 2 \\ s_4^i; k := 1, k = 3 \text{ if } \mathcal{C}_{ijkl}^{ij+1}, S_0^w = s_3^i \\ \end{cases}$$

If the index $k$ is 1 or 2, then in this case the internal state of the current context, the index is incremented. If the index is 3, $\mathcal{C}$ for which the first internal state is equal to the subsequent state of the current context, is randomly selected and stored. The next generated state is the second internal state of the selected descriptor. Schematically, the process of generating states by forward transitions through context descriptors can be represented, as shown in Figure 10.

![Figure 10 The process of generating states by forward transitions](image)

Thus, the generation of a sequence of states on the basis of contexts ensures compliance with the set conditions for the states model: a random sequence of events of unlimited length is generated, and due to their “overlapping” in the described way, there is only one event between two successive states.

When transitioning between states within the context, a random interval is generated that has a normal distribution with mathematical expectation and variance parameters in the corresponding context descriptor.

$$T_{i+1} = T_i + norm\left(\mu_j^{\Delta t_{k-1}}, \sigma_j^{\Delta t_{k-1}}\right)$$
During the transition between states, a situation where there is no context descriptors for which $s^w_0$ coincides with $S_0$ is possible. In this case, a reverse transition occurs within the current context until an initial state is reached that corresponds to the situation when no key is pressed. Based on this, in the process of forward transitions when moving to the next context descriptor, the previous descriptor is pushed onto the stack. In the process of reverse transitions, descriptors are sequentially extracted from it.

The reverse generation process can be represented, as shown in Figure 11.

![Diagram showing reverse generation process](image)

Figure 11 The process of generating states by reverse transitions

Upon reaching the internal state corresponding to the initial one, the forward generation process begins again.

To complicate the simulation model, it is proposed to expand the state space by comparing the transition between two states with several intervals allocated as a result of cluster analysis of the intervals of the original user sample.

4.4 The process of modeling and debugging a model

As a result, a component of imitation modeling was developed according to the proposed mathematical model. According to the test results, this component allowed to generate about 20 events per second, which is about 6-12 times faster than the average continuous typing speed (at a speed of 200-400 characters per second). In addition, tests showed that the resulting model generates states that are approximately 80% present in the model and, as a result, are involved in making an identification decision.

6. Conclusions

Thus, for today, behavioral biometry and analysis of the user's keyboard handwriting is an important and urgent task of ensuring the security of information systems and user information.

As part of a study on the development of methods for the dynamic analysis of keystroke dynamics, a method is proposed for representing the process of typing on the keyboard in the form of a successive change of states. Based on states it is proposed to define certain sequences called the state context as feature of a biometric image.

Based on this representation, an imitation model of keystroke dynamics was developed, which allows replacing the physical source of biometric data for the purpose of testing and debugging the identification system of keystroke dynamics.
The simulation component based on the presentation of state contexts has the following advantages:

- Allows you to repeat the model of key press overlapping as described earlier;
- The generated events form the contexts that are in the user's stored biometric image;
- Allows you to generate a random sequence of events of unlimited length.

However, the main drawback of this approach is that the generated sequence of characters does not correspond to the real typing of meaningful text.

One of the directions for further development is increasing in the performance of the simulation imitation component in order to increase the number of generated events and reduce the time of one debug iteration and optimize the model debugging process [20-21]. In addition, in this area, it is necessary to develop a method for introducing distortions into the generated image in order to debug analysis algorithms for periodic short-term and long-term changes in the user's keyboard handwriting.

References

[1] D. V. Pashchenko, E. A. Balzannikova , I. G. Sergina, "User identification method by means of biometric image of keystroke dynamics with double-chained representation" Вопросы радиоэлектроники, № 12, C 83–89, 2018, DOI 10.21778/2218-5453-2018-12-83-89.

[2] R. Joyce and G. Gupta "Identity Authentication Based on Keystroke Latencies" Communications of the ACM vol. 33 no. 2 pp. 168-176, 1990, DOI: 10.1145/75577.75582

[3] Shen Teh P., Beng Jin Teoh A, Yue S. “A Survey of Keystroke Dynamics Biometrics”, The Scientific World Journal –Hindawi Publishing Corporation, 24 p., 2013.

[4] R. V. Eruslanov “Keyboard identification of the user on the basis of the multilinked presentation method”, Radio engineering and infocommunication systems, №3 pp 43 – 50, 2009.

[5] U.A. Bryukhomitsky “Bar graph keystroke dynamics recognition method” News of the Southern Federal University. Technical science, vol.112, №11, 8 p, 2010.

[6] Bryukhomitsky U.A., and Kazarin M.N. “Multi-connected representation of biometric parameters in keyboard monitoring systems”, Information counteraction to threats of terrorism. Scientific and practical journal №14, 144-148 pp., 2010

[7] Bryukhomitsky U.A., and Kazarin M.N. "The method of biometric user identification by keyboard handwriting based on the Haar decomposition and Hamming proximity measure" Izvestiya Southern Federal University. Technical science, vol. 33, no. 4, 2003, pp. 141-149.

[8] I. G. Sidorkina, A. N. Savinov. "Three algorithms of control access to the KSSI on the basics of recognition of keystroke dynamics" Bulletin of the Chuvash University, no. 3, 2013, pp. 293-301.

[9] D.S Krutovostov, V.E. Khitsenko, “Password authentication and continuous authentication by keystroke dynamics using mathematical statistics”, Cybersecurity issues, №5(24), 2017, 91 – 99 pp, DOI: 10.21681/2311-3456-2017-5-91-99.

[10] P. Bours, S. Mondal,. “Continuous Authentication with Keystroke Dynamics”. 2015DOI: 10.13140/2.1.2642.5125..

[11] Mondal, Soumik & Bours, Patrick. (2017). Person Identification by Keystroke Dynamics Using Pairwise User Coupling. IEEE Transactions on Information Forensics and Security. PP. 1-1. 10.1109/TIFS.2017.2658539.
[12] Syed Idrus, Syed Zulkarnain & Cherrier, Estelle & Rosenberger, Christophe & Bours, Patrick. (2014). Soft Biometrics for Keystroke Dynamics.

[13] P. Bours, H. Barghouthi “Continuous authentication using keystroke dynamics,” Proceedings of the Norsk Informasjonssikkerhetskonferanse (NISK'09). 1-11, 2009.

[14] A Bhatia, Hanmandlu, M. “Keystroke Dynamics Based Authentication Using Information Sets,” Journal of Modern Physics, Vol.8 No.9, August 2017, DOI: 10.4236/jmp.2017.89094.

[15] Vuyyuru, Sampath K. et al. “Computer User Authentication using Hidden Markov Model through Keystroke Dynamics.” (2006).

[16] Martyshkin, A.I. & Pashchenko, D.V. & Trokoz, D.A.. (2019). Queueing Theory to Describe Adaptive Mathematical Models of Computational Systems with Resource Virtualization and Model Verification by Similarly Configured Virtual Server. 1-6. 10.1109/RUSAUTOCON.2019.8867620.

[17] Alexey, I. Martyshkin and V. Pashchenko Dmitry. “Development and Research of Means for the Collection and Reporting of Open-Source Information for Use in Decision-Making Systems.” Journal of Computational and Theoretical Nanoscience 16 (2019): 3103-3114.

[18] Dubravin, Aleksey & Zinkin, Sergey & Paschenko, Dmitry. (2015). Formal and conceptual definitions of the hybrid model of distributed computings in networks. 1-6. 10.1109/SIBCON.2015.7147047.

[19] Pashchenko, Dmitry & Trokoz, Dmitry & Martyshkin, Alexey & Sinev, Mihail & Svistunov, Boris. (2020). Search for a substring of characters using the theory of non-deterministic finite automata and vector-character architecture. Bulletin of Electrical Engineering and Informatics. 9. 10.11591/eei.v9i3.1720.

[20] Pashchenko, Dmitry et al. “Using Algebra of Hyper-Dimensional Vectors for Heuristic Representation of Data While Training Wide Neural Networks.” 2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP) (2019): 168-171.

[21] Janarthanan, R., Balamurali, R., Annapoorani, A. and Vimala, V., 2020. Prediction of rainfall using fuzzy logic. Materials Today: Proceedings.