Application of binary dynamical systems in the problem of classification of Boolean vectors

G A Oparin, V G Bogdanova and A A Pashinin

Matrosov Institute for System Dynamics and Control Theory SB RAS, Lermontov St. 134, Irkutsk, Russia, 664033

bvg@icc.ru

Abstract. The article proposes a method based on using binary dynamical systems in the classification problem for Boolean vectors (binary feature vectors). This problem has practical application in various fields of science and industry, for example, bioinformatics, remote sensing of natural objects, smart devices of the Internet of things, etc. Binary synchronous autonomous nonlinear dynamic models with an unknown characteristic matrix are considered. Matrix elements are chosen in such a way that the Boolean reference vectors are equilibrium states of the binary dynamic model. The attraction regions of equilibrium states act as classes (one reference vector corresponds to each class). The classified vector is the initial state of the model. Simple and aggregated classifiers are considered. The proposed method is demonstrated using an illustrative example.

1. Introduction
The development of methods based on using dynamic models is one of the promising research areas in solving the classification problem for various object types [1]. The principle of these methods is that the memorized reference objects are the equilibrium states of a dynamic model, and the classes represent attraction regions of the corresponding equilibrium states. The tuning process for the dynamic model consists of selecting such values of its parameters that ensure in this model the presence of the required set of equilibrium states (attractors) with given quantitative characteristics of their regions of attraction (basins), as well as the required dynamics of the model's functioning.

The study aims to apply this approach to the Boolean feature vectors classification problem based on parameterized binary dynamic models. The use of such models determines the novelty of our method. Boolean feature vectors that need to be matched with one class or another are the initial conditions for a binary dynamical system. The classifier work result is the equilibrium state, to which the trajectory converges with the given initial conditions. If this equilibrium state belongs to the set of reference Boolean vectors, then the classification problem has a positive solution. Otherwise, the input binary feature vector does not belong to any of the predefined classes.

Many practical problems in various fields of science and technology, in particular, bioinformatics [2], remote sensing of natural objects [3], smart devices of the Internet of things [4], and others are reduced to the classification of Boolean vectors.

The development and analysis of a Boolean features system are critical in the classification theory [5], but they are subject-oriented and are not considered further.

The article is structured as follows. Section 2 provides a brief overview of the use of dynamic models in solving the classification problem. The statement of the classification problem for Boolean vectors is formulated in Section 3. In Section 4, the method for solving this problem is proposed. This method application using a nonlinear binary dynamic model is demonstrated with an illustrative example in Section 5. In the final section 6, the advantages of the proposed method are presented.

2. Related work

The well-known Hopfield neural network [6] and its development, the Hamming neural network [7], are among the most famous examples of the application of dynamical systems in the problem of classifying binarized objects. In the Hopfield model, reaching an equilibrium state does not guarantee the correct response of the network (the network converges to the so-called false attractors). The correct answer is an equilibrium state that matches one of the reference binary vectors. The Hamming network, unlike the Hopfield network, does not give out the reference vector but its number. If there are several such numbers, a conclusion is made about the impossibility of assigning the input vector to a specific class. However, the obtained numbers indicate the most similar reference vectors. Compared to the Hopfield network, the Hamming network is characterized by lower memory and computational costs. Improved Hopfield and Hamming network models have found the application for solving a wide range of problems from various fields of knowledge [2-4, 8, 9].

In addition to traditional models of neural networks, in which fixed points are attractors, models of so-called chaotic neural networks [10] are proposed, the most famous of which is Freeman's KIII model [11]. In such networks, the reference vector does not correspond to an equilibrium state, but to a cycle (oscillation between the reference and its inversion). For the "I don't know" state the dynamics of the network is chaotic (chaotic attractor). Chaotic neural networks are a special kind of dynamic neural network models in which deterministic chaos plays a key role in the classification process. Some areas of their application can be found in works [12-13].

In practice, using both types of networks is associated with high computational costs for computer modeling of systems of differential or difference equations. One of the main reasons for this is the need to use data of a real type and, accordingly, slow operations on them. The use of binary dynamic models allows the reducing computational costs. These models provide all transformations necessary for solving the classification problem using high-speed Boolean operations on binary variables.

3. Problem statement

We consider a nonlinear binary dynamic system of the following form:

$$x(t+1) = A \otimes x(t), x(0) = x^0, \tag{1}$$

where $t = 0,1,2,...$ is the discrete time, $x(t) = col(x_1(t), x_2(t),...x_n(t))$ is the state vector, $x(t) \in B^n, B = \{0,1\}$, B^n is the state space of system (1), A is an unknown binary matrix of dimension $n \times n$, the symbol $\otimes$ means the conjunctive product of two matrices defined as

$$C = A \otimes B, \text{ where } c_{ik} = \bigvee_j (a_{ij} \wedge b_{jk}).$$

A set of reference Boolean vectors is given as

$$P = \{p^1, p^2,...,p^m\}, \text{ where } p^i = col(p_1^i, p_2^i,...,p_n^i), \ p_j^i \in \{0,1\}, \ i = \overline{1,m}, \ j = \overline{1,n}.$$

It is required for system (1) to find such matrix A^* that the set P belongs to the set of equilibrium states of the system

$$x(t+1) = A^* \otimes x(t). \tag{2}$$

Further, using system (2), it is necessary to match the Boolean vector $x^0 \in B^n$ to one of the reference Boolean vectors from the set P (one reference vector from P corresponds to each class), or to conclude that the vector x^0 does not correspond to any of the reference vectors.

4. Solution method

Each reference vector from the set P must be an equilibrium state of the system (1), and therefore satisfy the system of nonlinear Boolean equations

$$p^i = A \otimes p^i, i = \overline{1,m} \tag{3}$$

with n^2 unknown variables, elements of the matrix A. The number of scalar equations in system (3) is equal to $m \cdot n$.

It is known that system (3) is reduced to a single Boolean equation $\Phi_1(A) = 1$. The function $\Phi_1(A)$ is easy to represent in conjunctive normal form, which makes it possible to use efficient AllSAT solvers to find solutions [14].

Depending on a given set of reference vectors P, an equation $\Phi_1(A) = 1$ can have a single solution, many solutions, and none. In the last case, the classification problem for the set of reference Boolean vectors P cannot be solved using the proposed method.

With a plurality of solutions, it is possible to add to constraint (3) additional requirements regarding the dynamics of system (1) and the size of the region of attraction of reference vectors:

1. Each row of the matrix A must contain exactly one 1. The corresponding Boolean constraint is written as $\Phi_2(A) = 1$. This condition ensures that for each scalar equation of system (1) there is only one variable in the right side, which makes it possible to solve the classification problem for any $x^0 \in B^n$ in one step (the radius of the region of attraction of each equilibrium state, including for the reference vector, is equal to one).

2. The number k of zero columns in the matrix A must be maximum ($k = k_{\max}$). This condition allows to obtain the maximum region of attraction, equal $2^{k_{\max}} - 1$ states, for each reference vector. We can write the Boolean constraint on the presence of k zero columns in the matrix A as an equation $\Phi_3^k(A) = 1$.

All three constraints are reduced to one Boolean equation

$$CNF = \Phi_1(A) \wedge \Phi_2(A) \wedge \Phi_3^k(A) = 1. \tag{4}$$

An iterative process for searching $k_{\max}$ is organized for the parameter k using the well-known interval halving method. In a considered problem, such an interval is a set of integers from 0 to n-1. The calculation of all solutions to equation (4) (or identification of their absence) for a given k is performed by the AllSAT solver [14].

If $k_{\max} = 0$ and equation (4) is solvable with respect to A, the solution matrix A^* will be permutable (exactly one 1 in a row and the absence of zero columns). In this case, all equilibrium states will be isolated (with empty regions of attraction) and a one-step classification is meaningless.

If $k_{\max} > 0$ two ways are possible:

1) The equation (4) has one solution A^*. Then if for a given initial vector $x^0 \in B^n$ there is a vector $x^1 = A^* \otimes x^0 \in P$, then the classification problem for the vector x^0 is solved,

otherwise none of the reference vectors corresponds to the vector x^0. Such a classifier with a single matrix A^* is called a simple classifier.

2) Equation (4) has many solutions $A_1^*, A_2^*, ..., A_q^*$. This case corresponds to an aggregated classifier, which is an ensemble of simple classifiers:

$$x_1^1 = A_1^* \otimes x^0, x_2^1 = A_2^* \otimes x^0, ..., x_q^1 = A_q^* \otimes x^0.$$

If these vectors $x_1^1, x_2^1, ..., x_q^1$ are pairwise equal and $x_1^1 \in P$, then belonging of x^0 to a certain class is established. If for all $i = 1, 2, ..., q$ $x_i^1 \notin P$, then none of the reference vectors is comparable to the vector x^0. We will say that the aggregated classifier produces a deterministic solution if one of these two conditions is satisfied. Otherwise, we have a probabilistic solution when, based on the analysis of vectors $x_1^1, x_2^1, ..., x_q^1$, the probability of a vector x^0 belonging to a particular class or the probability of the absence of a property of belonging to any class is calculated. In this case, the aggregated classifier produces a probabilistic solution. Suppose the input vector x^0 is comparable with reference vectors $b \in P$ and $c \in P$ with probabilities correspondently of 0.3 and 0.5. Additionally, x^0 is not matched with any reference vector from the set P with a probability of 0.2. The sum of these probabilities is 1.

The software implementation of the proposed technology for classifying Boolean vectors is based on the HPCSOMAS framework [15] in the form of a set of composite microservices [16]. These microservices implement some functions of the Boolean constraints method in the qualitative analysis of binary dynamical systems [17].

HPCSOMAS automates the creation of microservices based on application modules and managing their execution in a distributed computing environment. This framework also provides post-processing tools, in particular, graphical visualization of the results.

5. Illustrative example

It is required to construct a classifier for five ($m = 5$) reference Boolean vectors $p^1, p^2, ..., p^5$ of dimension $n = 9$ (table 1). The list of input vectors to be classified is given in table 2.

Table 1. Reference vectors.

Reference	Binary features								
	1	2	3	4	5	6	7	8	9
P^1	0	1	0	0	1	0	0	1	0
P^2	1	1	1	1	0	1	1	0	1
P^3	1	0	1	1	1	1	1	0	1
P^4	1	1	1	1	0	0	1	0	0
P^5	0	0	0	0	1	0	0	0	0

We synthesize CNF in the DIMACS format [18] from equation (4) and start the search k_{max} procedure with simultaneous calculation of all solutions to equation (4) using the AllSAT solver. Applying this procedure, we obtain $k_{max} = 4$ and 8 matrices A^* shown in figure 1. Each matrix has exactly one 1 in each row and 4 zero columns.

Table 2. Input vectors.

x^0	Binary features								
	1	2	3	4	5	6	7	8	9
s^1	0	0	1	1	1	1	1	0	1
s^2	0	0	1	0	1	0	0	0	0
s^3	0	0	0	0	1	0	0	1	0

Let us consider the construction of a simple classifier using a matrix A_3^* as an example (figure 1). Based on the matrix A_3^*, we write the binary dynamical system (2) in scalar form:

$$x_1^{t+1} = x_7^t, \quad x_2^{t+1} = x_2^t, \quad x_3^{t+1} = x_7^t,$$
$$x_4^{t+1} = x_7^t, \quad x_5^{t+1} = x_5^t, \quad x_6^{t+1} = x_6^t, \tag{5}$$
$$x_7^{t+1} = x_7^t, \quad x_8^{t+1} = x_8^t, \quad x_9^{t+1} = x_6^t.$$

A_1^*
	1	2	3	4	5	6	7	8	9
1	0	0	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0

A_2^*
	1	2	3	4	5	6	7	8	9
1	0	0	0	1	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0

A_3^*
	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	1	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0

A_4^*
	1	2	3	4	5	6	7	8	9
1	0	0	0	1	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

A_5^*
	1	2	3	4	5	6	7	8	9
1	0	0	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

A_6^*
	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

A_7^*
	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0

A_8^*
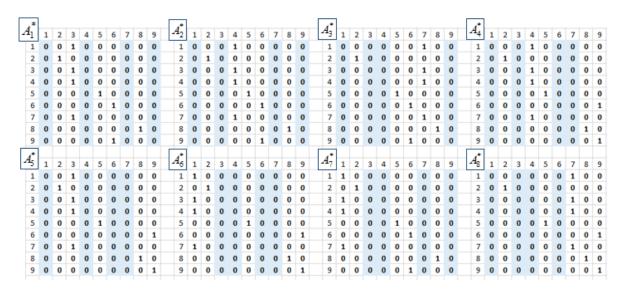|	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	1	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

Figure 1. All founded matrixes $A_1^*, A_2^*, ..., A_8^*$.

Using HPCSOMAS services, we perform the following actions:

- Search for equilibrium states;
- Verifying the presence of reference vectors among these states;
- Verifying the isolation property for reference vectors;
- Search for its attraction regions (basins);
- Visualization of reference vector basins.

Figure 2 shows 32 found equilibrium states in the output DIMACS format of SAT solver. In these format, variables $x_1^0, x_2^0, ..., x_9^0, x_1^1, x_2^1, ..., x_9^1$ are encoded by one-dimensional indexes $1, 2, ..., 9, 10, 11, ..., 18$. Indexes are calculated for x_i^t using the formula $i + nt, i = \overline{1,9}, t = \overline{0,1}, n = 9$.

These indexes are positive or negative if the x_i^t is equal correspondently to 1 or 0. For example, the reference vector p^5 (0 0 0 0 1 0 0 0 0) is shown for $t = 0$ as (-1 -2 -3 -4 5 -6 -7 -8 -9) in figure 2. The value 0 denotes the end of a string in DIMACS format. Reference vectors are highlighted in bold.

The isolation property for these vectors is not satisfied. Therefore reference vectors are attractors of the system (5). Figure 3 shows the basins of these attractors including 15 states. In figure 3, the decimal values of the states are calculated by the formula $\sum_{i=1}^{9} a_i * 2^{i-1}$, where $a_1 a_2 .. a_9$ - the corresponding binary state values.

The analysis of the attractor regions in Figure 3 shows that the classifier has the property of correctly recreating the reference vector for a highly distorted input Boolean vector x^0.

We use a simple classifier to match the input vectors s^1, s^2, s^3, presented in table 2, with the reference ones. As a result, the response of system (5) to the vector s^1 coincides with the reference vector p^3 and to the vector s^2 - with the reference vector p^5. The response to the vector s^3 does not coincide with any of the reference vector.

```
     -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 0
     -1 2 -3 -4 -5 -6 -7 -8 -9 -10 11 -12 -13 -14 -15 -16 -17 -18 0
p5   -1 -2 -3 -4 5 -6 -7 -8 -9 -10 -11 -12 -13 14 -15 -16 -17 -18 0
     -1 2 -3 4 5 -6 -7 -8 -9 -10 11 -12 -13 -14 -15 -16 -17 -18 0
     -1 -2 -3 -4 -5 -6 7 8 -9 -10 -11 -12 -13 -14 -15 -16 17 -18 0
     -1 -2 -3 4 5 -6 7 8 -9 -10 -11 -12 -13 14 -15 -16 17 -18 0
     -1 2 -3 -4 -5 -6 7 8 -9 -10 11 -12 -13 -14 -15 -16 17 -18 0
p1   -1 2 -3 -4 5 -6 7 8 -9 -10 11 -12 -13 14 -15 -16 17 -18 0
     -1 -2 -3 -4 -5 6 7 -8 -9 -10 -11 -12 -13 -14 15 16 -17 18 0
     -1 2 -3 4 5 6 -7 -8 -9 -10 11 -12 -13 -14 15 16 -17 18 0
     -1 -2 -3 -4 5 6 7 8 9 -10 -11 -12 -13 -14 15 16 17 18 0
     -1 -2 -3 4 5 6 -7 8 9 -10 11 -12 -13 -14 15 -16 17 18 0
     -1 -2 -3 -4 5 6 -7 -8 9 -10 -11 -12 -13 14 15 -16 -17 18 0
     -1 -2 -3 4 5 6 -7 -8 9 -10 -11 -12 -13 14 15 -16 17 18 0
     -1 -2 -3 4 5 6 7 -8 9 -10 11 -12 -13 14 15 -16 17 18 0
     -1 -2 -3 4 5 6 -7 -8 9 -10 11 -12 -13 14 15 -16 17 18 0
     1 -2 3 4 -5 -6 7 -8 9 10 -11 12 13 -14 -15 16 -17 -18 0
     1 -2 3 4 -5 6 7 -8 9 10 11 12 13 -14 -15 16 -17 -18 0
p4   1 2 3 4 -5 -6 7 -8 -9 10 11 12 13 -14 -15 16 -17 -18 0
     1 2 3 4 -5 -6 7 8 -9 10 11 12 13 14 -15 16 17 -18 0
     1 -2 3 4 5 -6 7 -8 -9 10 -11 12 13 14 -15 16 -17 -18 0
     1 2 3 4 5 -6 7 -8 -9 10 11 12 13 14 -15 16 17 -18 0
     1 2 3 4 5 -6 7 8 -9 10 11 12 13 14 -15 16 17 -18 0
     1 -2 3 4 -5 6 7 -8 9 10 -11 12 13 -14 15 16 17 18 0
     1 2 3 4 -5 6 7 8 9 10 -11 12 13 14 15 16 17 18 0
p3   1 -2 3 4 5 6 7 -8 -9 10 -11 12 13 14 15 16 -17 18 0
     1 -2 3 4 5 6 7 8 9 10 -11 12 13 14 15 16 17 18 0
p2   1 2 3 4 -5 6 7 -8 -9 10 11 12 13 -14 -15 16 -17 18 0
     1 2 3 4 5 6 7 -8 9 10 11 12 13 -14 -15 16 17 18 0
     1 2 3 4 -5 6 7 8 9 10 11 12 13 -14 15 16 17 18 0
     1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 0
```
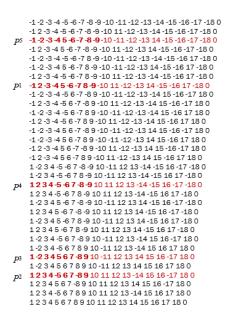

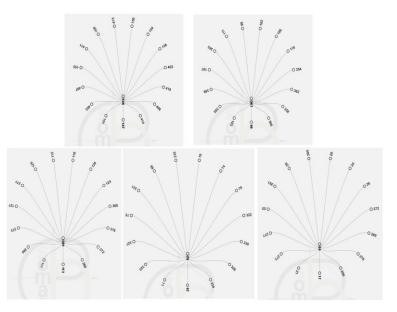**Figure 2**. Equilibrium states.   **Figure 3**. Visualization of basins of reference vectors.

We use the aggregated classifier to match the input vectors s^1, s^2, s^3, s^4, s^5, s^6 (table 3) with the reference ones.

Table 3. Input vectors.

x^0	Binary feature								
	1	2	3	4	5	6	7	8	9
s^1	0	0	1	1	1	1	1	0	1
s^2	0	0	1	0	1	0	0	0	0
s^3	0	0	0	0	1	0	0	1	0
s^4	0	1	1	1	0	1	1	0	0
s^5	0	1	1	0	1	0	0	1	1
s^6	0	1	1	1	0	1	0	0	1

The classification results are shown in table 4. The response of system (5) to the vector s^1 coincides with the reference vector p^3 with probability 3/4, to the vector s^2 - with the reference vector p^5 also with the probability 3/4, to the vector s^6 - with the reference p^2 with probability 1/2, and on the vector s^5 - with the reference p^1 with probability 3/8. The response of the system (1) to the vector s^3 does not coincide with any reference vector, and for the vector s^4 it is revealed that it belongs to different classes - p^2 and p^4 with the same probability 3/8.

Table 4. Results of the aggregated classifier operation.

x^0	Simple classifier							
	A_1^*	A_2^*	A_3^*	A_4^*	A_5^*	A_6^*	A_7^*	A_8^*
s^1	p^3	p^3	p^3	p^3	p^3	-	-	p^3
s^2	-	p^5	p^5	p^5	-	p^5	p^5	p^5
s^3	-	-	-	-	-	-	-	-
s^4	p^2	p^2	p^2	p^4	p^4	-	-	p^4
s^5	-	p^1	p^1	-	-	-	p^1	-
s^6	p^2	p^2	-	p^2	p^2	-	-	-

6. Conclusion

Boolean vectors classification method based on the use of binary nonlinear dynamical systems is proposed. The software implementation of this method is performed using the HPCSOMAS framework. An example of using the developed software is considered.

The advantages of the proposed method lie in the simplicity of tuning the binary dynamic system to a given set of reference vectors, solving the classification problem in one step of the dynamic system functioning, as well as calculating the probability of belonging the input vector x^0 to each of the specified classes of the set P. Additionally, in our model (in contrast to the Hopfield model), there are no false attractors, which significantly speeds up the classification process.

Despite the fact that all actions of the proposed algorithm are performed on binary values using only Boolean algebra operations, such a classifier has good generalization abilities and is capable of correctly recreating the reference for a strongly distorted input Boolean vector x^0.

References
[1]   Andreyev Yu V, Dmitriev A S and Matveev M A 1995 Application of chaotic dynamical systems to the problems of recognition and classification *Proc. of the 3rd Int. Workshop NDES-95, Dublin, Ireland* pp 249–252
[2]   Conforte A J, Alves L, Coelho F C, Carels N and Silva F A B 2020 Modeling basins of attraction for breast cancer using Hopfield networks *Frontiers in Genet*ics **11** 314
[3]   Khristodulo O I, Makhmutov A A and Sazonova T V 2017 Use algorithm Based at Hamming Neural Network Method for Natural Objects Classification *Procedia Computer Science* **103** 388–395

[4] Boriskov P 2020 IoT-oriented design of an associative memory based on impulsive Hopfield neural network with rate coding of LIF oscillators *Electronics* **9** 1469

[5] Hancock J T and Khoshgoftaar T M 2020 Survey on categorical data for neural networks *J Big Data* **7** 28

[6] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *PNAS* **79** (8) 2554–2558

[7] Lipman R 1987 An introduction to computing with neural nets *IEEE Acoustic, Speech and Signal Processing Magazine* **4** (2) 4–22

[8] Cantini L and Caselle M 2019 Hope4Genes: a Hopfield-like class prediction algorithm for transcriptomic data *Sci Rep* **9** 337

[9] Koutroumbas K and Kalouptsidis N 2005 Generalized hamming networks and applications *Neural Networks* **18**(7) 896–913

[10] Andreyev Yu V, Belsky Yu L and Dmitriev A S 1992 Information processing in nonlinear systems with dynamic chaos *Proc. Int. Seminar Nonlinear Circuits and Systems Moscow, Russia* vol 1 pp 51–60

[11] Freeman W J 1987 Simulation of chaotic EEG patterns with a dynamic model of the olfactory system *Biological Cybernetics* **56** 139–150

[12] He Y and Wang L 2000 Chaotic neural networks and their *applications Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No. 00EX393), Hefei, China* vol 2 pp 826–830

[13] Velichko A 2020 Neural network for low-memory IoT devices and MNIST image recognition using kernels based on logistic map *Electronics* **9** 1432

[14] Toda T and Soh T 2016 Implementing Efficient All Solutions SAT Solvers *ACM Journal of Experimental Algorithmics* **21**(1) 1–44

[15] Oparin G A, Bogdanova V G, Pashinin A A and Gorsky S A 2019 Microservice-oriented approach to automation of distributed scientific computations *Proc. of the 42st Int. Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019, Opatija, Croatia* pp 253-258

[16] Oparin G A, Bogdanova V G and Pashinin A A 2020 Automated tools for the development of microservice compositions for hybrid scientific computations *Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments, Irkutsk, Russia* pp 201–213

[17] Oparin G, Bogdanova V and Pashinin A 2019 Qualitative analysis of autonomous synchronous binary dynamic systems *MESA* **10**(3) 407–419

[18] Satisfiability suggested format. [Online]. Available: http://beyondnp.org/static/media/uploads/docs/satformat.pdf

[19] Irkutsk Supercomputer Centre of SB RAS. [Online]. Available: http://hpc.icc.ru