A Modification of Training and Recognition Algorithms for Recognition of Abnormal Behavior of Dynamic Systems

Victor Shcherbinin and Valery Kostenko
Department of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia

Keywords: Genetic Algorithm, Machine Learning, Supervised Learning, Recognition Algorithm, Dynamic System, Training Set, Algebraic Approach.

Abstract: We consider the problem of automatic construction of algorithms for recognition of abnormal behavior segments in phase trajectories of dynamic systems. The recognition algorithm is constructed using a set of examples of normal and abnormal behavior of the system. We use axiomatic approach to abnormal behavior recognition to construct abnormal behavior recognizers. In this paper we propose a modification of the genetic recognizer construction algorithm and a novel DTW-based recognition algorithm within this approach. The proposed modification reduces search space for the training algorithm and gives the recognition algorithm more information about phase trajectories. Results of experimental evaluation show that the proposed modification allows to reduce the number of recognition errors by an order of magnitude and to reduce the training time by a factor of 2 in comparison to the existing recognizer and recognizer construction algorithm.

1 INTRODUCTION

Consider a dynamic system information about which can be accessed by reading data from sensors surrounding the system. The sensor readings are obtained from sensors with a fixed frequency $1/\tau$.

A *multidimensional phase trajectory* in the space of sensor readings is an ordered set of vectors $X = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_k})$, where $\overline{x_i} \in \mathbb{R}^s$ is a vector of sensor readings at $t = t_0 + i \cdot \tau$.

We assume that at any given moment of time the system can be in one of two states:

- *Normal state.* In this state, the system is fully functional, and there are no signs that it is going to lose any of its functionality any time soon.
- *Abnormal state.* In this state, the system is not fully functional or is going to lose some of its functions soon.

The behavior (trajectory in the space of observed parameters) that precedes a transition of the system from normal state to abnormal state is called *abnormal behavior*. We suppose that there are L classes of abnormal behavior, each of these classes is characterized by a phase trajectory X^l_{Anom} called *a reference trajectory*.

The observed phase trajectory X of the system can have segments of abnormal behavior which are distorted compared to the reference trajectories. The distortions can be classified as amplitude distortions and time distortions. We say that a segment of abnormal behavior is distorted by amplitude compared to a reference trajectory if values in some points of the segment differ from those in the corresponding points of the reference trajectory. We say that a segment of abnormal behavior is distorted by time compared to a reference trajectory if there are missing or extra points in the segment compared to the reference trajectory. An example of an amplitude distortion is a stationary noise.

The problem of recognizing abnormal behavior can be defined as follows. We have

- An observed multidimensional trajectory X;
- A set of L classes of abnormal behavior for each of which reference trajectory X^l_{Anom} is defined;
- Recognition accuracy requirement:

$$e_I \leq const_1, e_{II} \leq const_2 \quad (1)$$

Here e_I is the number of type I errors, e_{II} is the number of type II errors, $const_1$ and $const_2$ are given numerical constraints.

We need to recognize abnormal behavior of the system, i. e. to find abnormal behavior segments in the trajectory X and abnormal behavior class number for each segment found.

Shcherbinin V. and Kostenko V..
A Modification of Training and Recognition Algorithms for Recognition of Abnormal Behavior of Dynamic Systems.
DOI: 10.5220/0004552201030110
In *Proceedings of the 5th International Joint Conference on Computational Intelligence* (ECTA-2013), pages 103-110
ISBN: 978-989-8565-77-8

This problem belongs to a class of pattern recognition problems. A wide variety of methods are used in the pattern recognition field, including the methods based on artificial neural networks (Haykin, 1998), k-nearest neighbour algorithm (Cover and Hart, 1967), algorithms based on Singular Spectrum Analysis (Hassani, 2007), etc. However, application of these methods and algorithms to this particular problem is complicated because of the presence of non-linear amplitude and time distortions of abnormal behavior segments in the observed phase trajectory X. To overcome these difficulties (emerging from the properties of dynamic systems in question) a parametric family of recognition algorithms based on algebraic approach was introduced in (Kovalenko et al., 2005). The idea of this parametric family is based on the idea of using algebraic approach to label planar configurations described in (Rudakov and Chekhovich, 2003). A genetic training algorithm for the parametric family was suggested in (Kovalenko et al., 2010). Results from (Kostenko and Shcherbinin, 2013) show that this parametric family of recognition algorithms demonstrates high tolerance to non-linear amplitude and time distortions of abnormal behavior segments compared to other approaches. In this paper we describe a modification of recognition algorithms from this parametric family and a modification of the genetic training algorithm from (Kovalenko et al., 2010).

2 CONSTRUCTION OF AN ABNORMAL BEHAVIOR RECOGNITION ALGORITHM USING A SET OF EXAMPLES

We call a set of dynamic system's trajectories $TS = \{X\}$ obtained in different conditions of its operation or via simulation of the system *a set of examples*. Each trajectory X from TS includes sections of normal and abnormal behavior. For each $X \in TS$ the starting point, the end point and abnormal behavior class number of each abnormal behavior segment is given.

The set of examples X is divided into three non-overlapping parts: a set of reference trajectories $\{X^l_{Anom}\}^L_{l=1}$, a training set $\widetilde{TS}$, and a validation set $\widehat{TS}$. The training set $\widetilde{TS}$ and the validation set $\widehat{TS}$ have the same size and contain trajectories that include both abnormal and normal behavior segments.

Suppose we are given an objective function $\varphi(e_I, e_{II}) : \mathbb{Z}_+ \times \mathbb{Z}_+ \to \mathbb{R}_+$ which is non-decreasing w.r.t. both its arguments. The problem of automatic construction of abnormal behavior recognition algorithm from a set of examples is formulated as follows (Kovalenko et al., 2005). Given

- a set of reference trajectories $\{X^l_{Anom}\}^L_{l=1}$,
- a training set $\widetilde{TS}$,
- a validation set $\widehat{TS}$,
- an objective function $\varphi(e_I, e_{II})$,

produce a recognition algorithm Al that satisfies the following conditions:

1. Al should show limited number of type I and type II errors on the training set $\widetilde{TS}$:
$$e_I(Al, \widetilde{TS}) \le const_1, e_{II}(Al, \widetilde{TS}) \le const_2 \quad (2)$$
Here $e_i(Al, TS)$ is the number of type i errors that Al makes on the trajectories from TS.
2. Al should minimize the objective function $\varphi(e_I, e_{II})$ on the validation set $\widehat{TS}$:
$$Al = \arg\min_{Al}(\varphi(e_I(Al, \widehat{TS}), e_{II}(Al, \widehat{TS}))) \quad (3)$$
3. Computational complexity $\Theta_{Al}(m)$ of the recognition algorithm Al on any trajectory of length less or equal to m should be limited by a given function $\theta(m)$:
$$\Theta_{Al}(m) \le \theta(m) \quad (4)$$
The function $\theta(m)$ is determined by the system operation rate and the available processing power.

The problem definition described here corresponds to the classic definition of the problem of learning from examples (a. k. a. supervised learning problem) described in (Vorontsov, 2004) and (Vapnik, 1998).

3 AXIOMATIC APPROACH TO ABNORMAL BEHAVIOR RECOGNITION

In this section we describe the parametric family of algorithms for recognition of abnormal behavior of dynamic systems introduced in (Kovalenko et al., 2005).

3.1 Basic Notions

Let $X = (x_1, x_2, \ldots, x_k)$, be a one-dimensional trajectory, $x_t \in \mathbb{R}$.

An elementary condition $ec = ec(t, X, p)$ is a function defined on a point t and its neighborhood on a trajectory X. It depends on a set of parameters p and takes either true value or false value.

An example of an elementary condition is

$$ec(t,X,p) = \begin{cases} true, & \text{if } \forall i \in [t-l, t+r] \\ & a \le x_i \le b, \\ false, & \text{otherwise.} \end{cases} \quad (5)$$

Here $p = \{a, b, l, r\}$ is the set of parameters of this elementary condition, $a, b \in \mathbb{R}$, $a < b$, $l, r \in \mathbb{N}^+$.

We have introduced the concept of an elementary condition for one-dimensional trajectories. However, an s-dimensional trajectory can be regarded as a collection of s one-dimensional trajectories. We introduce elementary conditions for multidimensional trajectories by adding to the elementary condition the number of one-dimensional trajectory to which it is applied as a parameter.

Let $X = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_k})$ be a multidimensional trajectory, $\overline{x_i} \in \mathbb{R}^s$.

An axiom $a = a(t, X)$ is a function defined as a Boolean formula over a set of elementary conditions defined on a point t and its neighborhood on a multidimensional trajectory X:

$$a(t,X) = \bigvee_{i=1}^{p} \bigwedge_{j=1}^{q} ec_{ij}(t, X, p_{ij}) \quad (6)$$

We call a finite collection of axioms $As = \{a_1, a_2, \ldots, a_m\}$ *an axiom system* if it meets the condition:

$$\forall X \, \forall \overline{x_t} \in X \quad \exists! \, a_i \in As : a(t,X) = true \quad (7)$$

I. e. for any point t in any trajectory X there exists one and only one axiom a_i in axiom system As that is true on point t.

A marking of a trajectory $X = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_k})$ by an axiom system $As = \{a_1, a_2, \ldots, a_m\}$ is a finite sequence

$$J = (j_1, \; j_2, \; \ldots \; , j_k)$$

of numbers of axioms from as, such that a_{j_t} is true on the point t of trajectory X.

3.2 The Recognition Algorithm

We define our parametric family of recognition algorithms S as a family of algorithms which recognize abnormal behavior segments in trajectory X by performing the following steps:

1. Perform marking of reference trajectories $\{X^l_{Anom}\}_{l=1}^{L}$ corresponding to different classes of abnormal behavior by an axiom system As.
2. Perform marking of trajectory X by an axiom system As. We denote the marking of trajectory X as J.
3. Perform fuzzy search for reference trajectory markings in marking J.

Each of the recognition algorithms from the parametric family S is defined by an axiom system and an algorithm for searching for the markings of reference trajectories in the marking of the observed trajectory.

The use of fuzzy search algorithms for searching markings allows us to tackle time distortions. Algorithms based on DTW (Keogh and Pazzani, 2001) are used for marking search.

3.3 Construction of the Recognition Algorithm

Construction of a recognition algorithm is performed in two stages:

1. The recognition algorithm is constructed for each pair (preprocessing algorithm, marking search algorithm): parameters of the preprocessing algorithm and marking search algorithm are determined, an axiom system is constructed.
2. From all constructed solutions we pick an algorithm that shows the least value of the objective function $\varphi(e_I, e_{II})$ on control sample $\widehat{TS}$.

Local optimization algorithms are used to adjust the parameters of preprocessing algorithm and marking search algorithm. The greatest difficulty is posed by construction of an axiom system.

4 AXIOM SET TRANSFORMATION

In the existing axiom system construction algorithms described in (Kovalenko et al., 2010), (Kostenko and Shcherbinin, 2013) an axiom system As is constructed by first constructing a set of axioms as for which the condition (7) is not guaranteed to hold. Then a transformation is applied to this set transforming it into an axiom system which is guaranteed to satisfy (7).

The cited works use a transformation of axiom set called *prioritizing of axioms*. Formally, prioritizing of axioms is given by

$$As = \{a'_1, \ldots, a'_m, a'_\infty\}, \quad (8)$$

where

$$\begin{aligned} a'_1 &= a_1 \\ a'_2 &= a_2 \wedge \neg a_1 \\ a'_3 &= a_3 \wedge \neg a_2 \wedge \neg a_1 \quad (9) \\ &\vdots \\ a'_m &= a_m \wedge \neg a_{m-1} \wedge \ldots \wedge \neg a_2 \wedge \neg a_1 \\ a'_\infty &= \neg a_m \wedge \neg a_{m-1} \wedge \ldots \wedge \neg a_2 \wedge \neg a_1 \end{aligned}$$

One can readily see that (7) is satisfied for As. Note that in order to apply this transformation to an axiom set as one needs to choose an order (i. e. priority) on the axiom set. That means that there are $m!$ ways to apply this transformation to a set of m axioms. This increases the search space of axiom system construction algorithms since they search for an axiom system which minimizes the objective function.

In this paper we propose a different type of transformation which we call *superset construction*. The idea is to map to point t not one axiom from as but the whole subset of axioms from as which are true on point t. Formally we can construct an axiom for each subset of as in the following way:

$$\begin{aligned} As &= \{a'_0, \ldots, a'_{2^m-1}\}, \qquad (10)\\ &\text{where}\\ a'_0 &= \neg a_1 \wedge \neg a_2 \wedge \ldots \wedge \neg a_{m-1} \wedge \neg a_m\\ a'_1 &= \neg a_1 \wedge \neg a_2 \wedge \ldots \wedge \neg a_{m-1} \wedge a_m\\ a'_2 &= \neg a_1 \wedge \neg a_2 \wedge \ldots \wedge a_{m-1} \wedge \neg a_m \qquad (11)\\ a'_3 &= \neg a_1 \wedge \neg a_2 \wedge \ldots \wedge a_{m-1} \wedge a_m\\ &\vdots\\ a'_{2^m-1} &= a_1 \wedge a_2 \wedge \ldots \wedge a_{m-1} \wedge a_m \end{aligned}$$

One can readily see that the result of this transformation is guaranteed to satisfy (7), i. e. this transformation can be used in construction algorithms in place of prioritizing of axioms. The proposed transformation has the following properties:

- Each axiom $a' \in As$ bijectively corresponds to a subset $\tilde{as} \subseteq as$. The fact that a' is true on a point t of a trajectory X is equivalent to the fact that each axiom from the corresponding subset $\tilde{as}$ is true on t and each axiom from $as \backslash \tilde{as}$ is false on t. Marking search algorithms can exploit the structure of the subsets corresponding to the elements of marking to improve the recognition quality.
- There is only one way to apply superset construction to an axiom set. That means that axiom system construction algorithms that use this transformation will have lesser search space compared to ones that use prioritizing of axioms.

5 GENETIC AXIOM SYSTEM CONSTRUCTION ALGORITHM

Genetic axiom system construction algorithm described in (Kovalenko et al., 2010) uses prioritizing of axioms as the axiom set transformation. An individual in this algorithm is an ordered set of axioms. In this paper we propose a modification of this algorithm that uses superset construction as the axiom set transformation. Two principal modifications of the algorithm are:

- *Modification of the structure of population individuals*. An individual in the new algorithm is an unordeded set of aixoms, since we don't need to choose an order in axiom set to apply superset construction.
- *Modification of mutation and crossover operations*. Since the structure of an individual is changed, we need to adjust mutation and crossover operations. In particular, we need to remove priority changing from these operations.

5.1 General Scheme of the Algorithm

The goal of genetic axiom system construction algorithm is to construct an axiom system that minimizes objective function $\varphi(e_I, e_{II})$ on validation set $\widehat{TS}$ given a fixed preprocessing algorithm and a fixed marking search algorithm.

The general scheme of the proposed algorithm is as follows:

1. Generation of the initial population.
2. Iterative optimization of the population:
   (a) Mutation of individuals.
   (b) Crossover of individuals and expansion of the population.
   (c) Selection of individuals and reduction of the population.
   (d) Checking of termination condition: iterative process is repeated until the termination condition is met.

An individual in the population Pl is an (unordered) set of axioms:

$$Pl = \{as^i\}, as^i = \{a^i_1, a^i_2, \ldots, a^i_{m_i}\}$$

To compute the objective function $\varphi(e_I, e_{II})$ for an individual axiom set as^i we apply superset construction transformation described in section 4 to as^i deriving an axiom system As^i, run recognition algorithm as described in subsection 3.2 using axiom system As^i, and calculate the number of type I and type II errors and the objective function $\varphi(e_I, e_{II})$.

During generation of the initial population individual axiom sets are generated randomly. Axioms are randomly constructed from a finite set of elementary conditions using boolean operations $\{and, or, not\}$. The used set of elementary conditions is defined by the user. Axiom sets are randomly constructed from generated axioms.

The algorithm ends if any of the following conditions is met:

- We found an axiom set as for which the value of objective function $\varphi(e_I, e_{II})$ is 0.
- The number of iterations of the algorithm exceeded a predefined value I_{all}.
- The number of iterations without decreasing of the lowest objective function value in the population exceeded a predefined value I_{stop}.

5.2 Genetic Algorithm Operations

Selection operation has two parameters: N_{as}^{max} — maximal number of axiom sets in population, $p \in [0, 1]$ — the fraction of best axiom sets that survive selection.

During selection the next generation population is formed from $\lfloor N_{as}^{max} \cdot p \rfloor$ axiom sets with lowest objective function value and $N_{as}^{max} - \lfloor N_{as}^{max} \cdot p \rfloor$ axioms chosen randomly from the current population.

Mutation and crossover are defined on three levels: elementary condition level, axiom level, axiom set level.

5.2.1 Elementary Condition Level

The mutation operation at this level alters parameter values of an elementary condition:

$$ec(t, X, p) \downarrow \begin{cases} ec(t, X, p) & \text{with probability } 1 - P_{ec}^{mut} \\ ec(t, X, p') & \text{with probability } P_{ec}^{mut} \end{cases} \quad (12)$$

Here $p' = m_{ec}(p, \Delta_{ec}^{mut})$ is a new set of parameter values of elementary condition ec, $m_{ec}(p, \Delta_{ec}^{mut})$ is a mutation function that alters parameters of ec. This function is specific for each type of elementary condition.

Parameters of this operation are P_{ec}^{mut} – mutation probability, Δ_{ec}^{mut} – degree of mutation (it determines how much parameters of elementary condition change).

The crossover operation produces an elementary condition from two parent elementary conditions of the same type:

$$(ec_1(t, X, p_1), ec_2(t, X, p_2)) \to ec_{12}(t, X, p_{12}) \quad (13)$$

The new elementary condition ec_{12} has the same type as its parents, each parameter value is inherited from one of the parents. For each parameter, the parent from which to inherit its value is chosen randomly. The probability of crossover of an elementary condition P_{ec}^{cross} is a parameter of this operation.

5.2.2 Axiom Level

The mutation operation at this level works on axioms:

$$a \downarrow \begin{cases} a & \text{with probability } 1 - P_a^{mut} \\ m_a(a, \Delta_a^{mut}) & \text{with probability } P_a^{mut} \end{cases} \quad (14)$$

Here $m_a(a, \Delta_a^{mut})$ is a mutation function that randomly adds a new elementary condition to the axiom, removes an elementary condition from the axiom, replaces an elementary condition with a randomly generated one, or changes the boolean operation between two elementary conditions (i. e. replaces *and* with *or* or vice-versa, adds or removes *not*).

Parameters of this operation are P_a^{mut} – mutation probability, Δ_a^{mut} – degree of mutation (i. e. the fraction of affected elementary conditions).

The crossover operation at this level generates a new axiom containing a combination of elementary conditions of the two parent axioms:

$$(a^1, a^2) \to a^{12} \quad (15)$$

Each elementary condition in a^{12} is either inherited from one of its parents or generated as a result of crossover between two elementary conditions of the same type randomly selected from different parents. The probability of crossover of an axiom P_a^{cr} is a parameter of this operation.

5.2.3 Axiom Set Level

The mutation operation at this level is defined as follows:

$$as \downarrow \begin{cases} as & \text{with probability } 1 - P_{as}^{mut} \\ m_{as}(as, \Delta_{as}^{mut}) & \text{with probability } P_{as}^{mut} \end{cases} \quad (16)$$

Here $m_{as}(as, \Delta_{as}^{mut})$ is a mutation function that adds or removes an axiom from axiom set as.

Parameters of this operation are P_{as}^{mut} – mutation probability, Δ_{as}^{mut} – degree of mutation (i. e. the fraction of affected axioms).

The crossover operation at this level generates a new axiom set from two parent axiom sets:

$$(as^1, as^2) \to as^{12}$$

Each axiom in as^{12} is either inherited from one of its parents or generated as a result of crossover between two axioms randomly selected from different parents. The probability of crossover of an axiom set P_{as}^{cr} is a parameter of this operation.

5.3 Selection of Parameters of Genetic Algorithm Operations

Each individual axiom set from the population and also each axiom from these axiom sets has parameters associated with it which control the mutation and crossover operations. To select these parameters for each individual on each step of the genetic algorithm we follow (Kovalenko et al., 2010) and introduce functions that evaluate population individuals and their parts and use them to adjust parameters of genetic algorithm operations for axiom sets and axioms. We call these functions *evaluation functions*.

We define an evaluation function M_{as} for axiom sets as follows:

$$M_{as} = c_1 e_I + c_2 e_{II} + c_3 \frac{\varphi_s(e_I, e_{II})}{\varphi_{min}(e_I, e_{II})} \quad (17)$$

Here e_I, e_{II} are the numbers of type I and type II errors; $\varphi_s(e_I, e_{II})$ is the objective function value for axiomatic set as on the training set $\widetilde{TS}$; $\varphi_{min}(e_I, e_{II})$ is the lowest objective function value in the population; c_i are given positive constants.

We define an evaluation function M_a for axioms as follows:

$$M_a = c_4 M_{as} + c_5 \left| \frac{Sec(\widetilde{TS}) - num_a}{Sec(\widetilde{TS})} \right| + c_6 \left| \frac{L - ref_a}{L} \right| \quad (18)$$

Here M_{as} is the evaluation function value for the axiom set that contains axiom a; $Sec(\widetilde{TS})$ is the number of abnormal behavior segments in the training set $\widetilde{TS}$; num_a is the number of points in the training set $\widetilde{TS}$ on which a is true; L is the number of reference trajectories; ref_a is the number of points in reference trajectories on which a is true; c_i are given constants.

Parameters of mutation and crossover operations for axiom sets and axioms are determined according to the value of evaluation functions:

$$[P_{as}^{mut}, \Delta_{as}^{mut}, P_{as}^{cr}] = F_1(M_{as}) \quad (19)$$

$$\left[P_a^{mut}, \Delta_a^{mut}, P_a^{cr}, P_{ec}^{mut}, \Delta_{ec}^{mut}, P_{ec}^{cr}\right] = F_2(M_a) \quad (20)$$

Functions F_1 and F_2 are chosen so as to satisfy the following conditions:

- All parameters of mutation and crossover operations should take a value within $[0, 1]$.
- All parameters of mutation operation: $P_{ec}^{mut}, \Delta_{ec}^{mut}, P_a^{mut}, \Delta_a^{mut}, P_{as}^{mut}, \Delta_{as}^{mut}$ should be directly proportional to the corresponding evaluation function.
- All parameters of the crossover operation: $P_{ec}^{cr}, P_a^{cr}, P_{as}^{cr}$ should be inversely proportional to the corresponding evaluation function.

Adjustment of parameters of mutation and crossover operations at every step for each individual allows to improve the algorithm convergence and to obtain better results.

6 DTW-BASED MARKING SEARCH ALGORITHM

In subsection 4 we point out that if axiom system As is a result of superset construction from axiom set $as = \{a_1, a_2, \ldots, a_m\}$, each axiom $a' \in As$ corresponds to a subset of as. We can say that marking J of a trajectory X by axiom system As corresponds to a sequence $\tilde{J}$ of sets $\tilde{j}_t \subseteq as$:

$$\tilde{J} = (\tilde{j}_1, \tilde{j}_2, \ldots, \tilde{j}_k) \quad (21)$$

Here $\tilde{j}_t = \{j : a_j(t, X) = true\}$ We will denote the set sequence $\tilde{J}$ corresponding to marking J as $s(J)$.

In this paper we propose a marking search algorithm that analyses axiom set sequences corresponding to markings. The proposed algorithm is based on DTW distance between two finite sequences of arbitrary elements (Keogh and Pazzani, 2001).

The goal of a marking search algorithm is to find segments which correspond to abnormal behavior trajectory markings in the observed trajectory marking. The algorithm uses sliding window approach and consists in the following:

1. We choose the point in the observed trajectory marking J from which we start recognition:
$$t = \min_{l=\overline{1,L}} (N_l^{min}) \quad (22)$$
Here
   - $N_l^{min} = \lfloor (1-s) \cdot len(X_{Anom}^l) \rfloor$ is the minimal window length for recognition of class l abnormal behavior segments. We use not one but several window lengths to better tackle time distortions.
   - $s \in (0, 1)$ is a parameter that controls minimal and maximal window lengths (relative to the length of reference trajectory).
2. For each class l whose reference trajectory marking length is not greater than t we check the following conditions:
$$\begin{cases} DTW(s(J_{Anom}^l), s(J_{t-N_l^{min}:t})) \le p_l \\ DTW(s(J_{Anom}^l), s(J_{t-N_l^{min}-1:t})) \le p_l \\ \vdots \\ DTW(s(J_{Anom}^l), s(J_{t-N_l^{max}+1:t})) \le p_l \end{cases} \quad (23)$$
Here

- $N_l^{max} = \lceil (1+s) \cdot len(X_{Anom}^l) \rceil$ is the maximal window length for recognition of class l abnormal behavior segments.
- $J_{t-N_l^{min}:t}$ is the marking of the segment of the observed trajectory from point $(t - N_l^{min})$ to point t.
- p_l is a parameter which determines how close the marking of a segment should be to the reference trajectory marking for it to be considered an abnormal behavior segment.

If any of conditions (23) is met, the corresponding segment is considered a class l abnormal behavior segment.

3. Move to the next point ($t \leftarrow t+1$). If $t = len(X)+1$ then the algorithm stops. Otherwise the algorithm proceeds from item 2.

Note that we compute all DTW distances from (23) in one go, i. e. with time complexity $O(N_l^{max} * len(X_{Anom}^l))$, using the approach described in (Müller, 2007).

7 EXPERIMENTAL EVALUATION

During experiments we used the genetic axiom system construction algorithm described in section 5 together with DTW-based marking search algorithm described in section 6. The experiments were conducted on artificial data generated by using a software program that can generate a set of precedents with given characteristics and given reference abnormal behavior trajectories. The values in the points of normal behavior segments were generated so that they obey the Gaussian distribution. The abnormal behavior segments were generated as stretched or squeezed reference abnormal behavior trajectories with Gaussian noise applied to them.

We compared the results and training time for recognizers constructed by existing algorithm described in (Kovalenko et al., 2010) with the results and training time for recognizers constructed by the algorithm proposed in this paper.

7.1 Distance Functions

To be able to compute DTW distance between two sequences of subsets of as we need to define a distance function on 2^{as}, i. e. a function $d : 2^{as} \times 2^{as} \to [0, 1]$ which measures the degree of difference between two subsets of as. We used the following distance functions during experiments:

1. Trivial distance function:

$$d(as_1, as_2) = \begin{cases} 1 & as_1 \neq as_2 \\ 0 & as_1 = as_2 \end{cases} \quad (24)$$

2. Distance function based on Jaccard coefficient (Tan et al., 2005):

$$d(as_1, as_2) = \begin{cases} 1 - \frac{|as_1 \cap as_2|}{|as_1 \cup as_2|} & \text{if } as_1 \cup as_2 \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

3. Normalized Hamming metric (Hamming, 1950):

$$d(as_1, as_2) = 1 - \frac{|as_1 \cap as_2|}{|as|} \quad (26)$$

4. The following distance function proposed by the authors:

$$d(as_1, as_2) = \begin{cases} 1 - \frac{|as_1 \cap as_2|}{|\min(|as_1|, |as_2|)|} & \text{if } as_1 \cap as_2 \neq \varnothing \\ 1 & \text{if } as_1 \cap as_2 = \varnothing, \\ & as_1 \cup as_2 \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

7.2 Results

The results for a dataset with time and amplitude distortions of abnormal segments relative to reference trajectories being up to 10%, test trajectory length of 3000 points and 2 abnormal behavior classes are shown in table 1. The results show that we were able to achieve better recognition quality and notably higher training speed when we used Hamming metric and the proposed metric compared to the existing algorithm. The proposed metric behaved slightly better. Other metrics demonstrated results that were worse than the results for the existing algorithm.

Table 1: Results for a dataset with time and amplitude distortions up to 10%, test trajectory length of 3000 points and 2 abnormal behavior classes. e_I is the number of type I errors, e_{II} is the percent of type II errors.

	e_I	e_{II}	Training time
Existing algorithm	17	0%	2 h. 12 min.
Trivial metric	6	5%	51 min.
Jaccard metric	27	0%	1 h. 17 min.
Hamming metric	1	0%	17 min.
Proposed metric	0	0%	15 min.

The results for a dataset with higher time and amplitude distortions (up to 30%), test trajectory length of 3000 points and 2 abnormal behavior classes are shown in table 2. The results show that we were able

to achieve better recognition quality and more than twice lesser training time when we used Hamming metric and the proposed metric compared to the existing algorithm. The proposed metric again behaved slightly better.

Table 2: Results for a dataset with time and amplitude distortions up to 30%, test trajectory length of 3000 points and 2 abnormal behavior classes.

	e_I	e_{II}	Training time
Existing algorithm	18	0%	2 h. 33 min.
Trivial metric	29	0%	1 h. 33 min.
Jaccard metric	52	0%	1 h. 18 min.
Hamming metric	6	0%	54 min.
Proposed metric	1	0%	53 min.

Overall results show that recognizers that are constructed with the proposed algorithm and use either Hamming metric or metric proposed by the authors can achieve better recognition quality while requiring less time for training even in the presence of amplitude and time distortions up to 30%.

8 CONCLUSIONS

This paper considers the problem of automatic construction of algorithms that recognize segments of abnormal behavior in multidimensional phase trajectories of dynamic systems. The recognizers are constructed using a set of examples of normal and abnormal behavior of the system. We employ axiomatic approach to abnormal behavior recognition to construct recognizers of abnormal behavior. In this paper we propose a modification of the way a set of axioms is transformed into an axiom system during recognizer construction. This modification implies modification of the training and recognition algorithm. We present modified genetic recognizer construction algorithm and DTW-based search algorithm.

Results of experimental evaluation of the proposed algorithms show that they allowed to decrease the number of errors by one order of magnitude compared to the old training and recognition algorithms and recognizer training took less than half of the time it took to train a recognizer using the old algorithms.

REFERENCES

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Tech. J.*, 29:147–160.

Hassani, H. (2007). Singular spectrum analysis: Methodology and comparison. *Journal of Data Science*, 5(2):239–257.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

Keogh, E. J. and Pazzani, M. J. (2001). Derivative dynamic time warping. In *First SIAM International Conference on Data Mining (SDM2001)*.

Kostenko, V. A. and Shcherbinin, V. V. (2013). Training methods and algorithms for recognition of nonlinearly distorted phase trajectories of dynamic systems. *Optical Memory and Neural Networks*, 22:8–20.

Kovalenko, D. S., Kostenko, V. A., and Vasin, E. A. (2005). Investigation of applicability of algebraic approach to analysis of time series. In *Proceedings of II International Conference on Methods and Tools for Information Processing*, pages 553–559. (in Russian).

Kovalenko, D. S., Kostenko, V. A., and Vasin, E. A. (2010). A genetic algorithm for construction of recognizers of anomalies in behaviour of dynamical systems. In *Proceedings of 5th IEEE Int. Conf. on Bio Inspired Computing: Theories and Applications*, pages 258–263. IEEEPress.

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Rudakov, K. V. and Chekhovich, Y. V. (2003). Algebraic approach to the problem of synthesis of trainable algorithms for trend revealing. *Doklady Mathematics*, 67(1):127–130.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vorontsov, K. V. (2004). Combinatorial substantiation of learning algorithms. *Journal of Comp. Maths Math. Phys*, 44(11):1997–2009.