Fuzzy classifier design for network intrusion detection using the gravitational search algorithm

To cite this article: M B Bardamova et al 2019 J. Phys.: Conf. Ser. 1145 012008

View the article online for updates and enhancements.
Fuzzy classifier design for network intrusion detection using the gravitational search algorithm

M B Bardamova, A A Konev, I A Hodashinsky, A A Shelupanov

Tomsk State University of Control Systems and Radioelectronics, 40, Lenina Ave.,
Tomsk, 634050, Russia

1E-mail: hodashn@rambler.ru

Abstract. The purpose of the work is to describe the method of constructing systems for
detection network intrusion based on an effective fuzzy classifier. Use of the gravitational
search algorithm for feature selection and tuning the parameters of the fuzzy classifier is
described. The effectiveness of the developed fuzzy classifiers was investigated using the KDD
Cup 1999 data set.

1. Introduction
An important challenge of information security is the protection of information transmission networks. When gaining access to the network, an intruder may be able to violate the integrity, confidentiality and availability of data and information processing systems.

Intrusion detection systems are used to detect facts of unauthorized access to a computer system or a network. One of the main components of intrusion detection systems is the classifiers, trained to distinguish between normal connection and suspicious traffic. Various methods can be used for classification: support vector machines, artificial neural networks, decision trees, fuzzy systems, and others. The methods of fuzzy sets and fuzzy logic are widely used in the field of data classification. The merits of fuzzy classifiers include their good interpretability [1].

When constructing a classifier for detecting attacks, it is necessary to ensure not only the high accuracy of the model but also to prevent the presence of excessive complexity. As a rule, the data on which traffic is monitored is of high dimensionality. For example, the most common KDD Cup training data set 1999 has 41 features [2]. However, not all features have a positive effect on classification. The presence of non-informative data leads to the construction of an overly complex model. Therefore, it is required to use a feature selection.

Feature selection methods can be divided into two categories: filters and wrappers [3-5]. Filters are based on certain metrics, e.g., entropy, probability distribution, and mutual information, and do not use classification algorithms. In turn, wrappers employ classifiers to estimate feature subsets with the classifier itself being "wrapped", so to speak, in the feature selection loop. Filters and wrappers have their own pros and cons. The advantage of filters is their scalability and low computational complexity. Their common disadvantage is that the absence of communication with the classifier, as well as ignoring the inter-feature relationships, results in low classification accuracy, which, however, differs for different classifiers. The advantage of wrappers is that they work in cooperation with a particular classification algorithm and take into account the synergetic effect from the joint use of
features selected. Their disadvantages are a higher risk of overfitting and higher time cost, which is due to the necessity of estimating classification accuracy [6].

This paper presents a method for constructing intrusion detection systems based on a fuzzy classifier with using the gravitational search algorithm for tuning system parameters and feature selection.

2. Fuzzy classifier

The fuzzy classifier is based on a set of fuzzy terms and the "if-then" fuzzy rules [7]. Each rule contains an assertion about the values of the input variables and specifies the value of the output variable as a class label:

$$R_{ij}: \text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } x_3 = A_{3i} \text{ AND } \ldots \text{ AND } x_n = A_{ni} \text{ THEN class} = c_j,$$

where $X = \{x_1, x_2, \ldots, x_n\}$ is the set of input features, $A_{ki}$ is the fuzzy term characterizing the $k$th feature in the $i$th fuzzy rule ($i \in [1, R]$), $R$ is the number of fuzzy rules and $C = \{c_1, c_2, \ldots, c_m\}$ is the set of classes.

The output of the classifier is determined by the degree of belonging of the input data from the observation table to their fuzzy terms:

$$j^* = \arg \max_{1 \leq j \leq m} \beta_j(x),$$

$$\beta_j(x) = \sum_{R_{ij}} \prod_{k=1}^{n} A_{jk}(x_k), \quad j = 1, 2, \ldots, m.$$

So the fuzzy classifier can be represented as a function that assigns a class label to a point in the feature space with a certain evaluable confidence:

$$f: \mathbb{R}^n \rightarrow [0,1]^m.$$

The criterion for the quality of classification is an accuracy. Assume we have an observations table $\{(x_p; c_p), p = 1, 2, \ldots, z\}$. The classification accuracy can be defined as

$$E(\theta, S) = \sum_{z=1}^{z} \left\{ \begin{array}{ll} 1, & \text{if } c_p = f(x_p, \theta, S) \\ 0, & \text{else} \end{array} \right.,$$

where $\theta = [\theta^1, \theta^2, \ldots, \theta^\theta]$ is the vector of terms parameters, $S = (s_1, s_2, \ldots, s_n)$ is the binary vector of features. If $s_i = 1$, the feature is used in classification, if $s_i = 0$, this feature is discarded.

The accuracy depends on the location of linguistic terms, so optimizing the vector of term parameters $\theta$ leads to an increase of the accuracy. For this purpose, the continuous gravitational search algorithm was applied [8]. The binary gravitational search algorithm was used to feature selection.

3. Gravitational search algorithm

The meta-heuristic gravitational search algorithm is based on the laws of gravitation and motion of Isaac Newton. The algorithm works either with vectors of terms parameters in the case of a continuous algorithm or with feature vectors in the case of a binary version. A population of optimizable vectors consists of a system of particles with gravitational forces acting between them [9].

At the stage of primary generation of the fuzzy classifier, carried out by the algorithm on the basis of extremums of classes, the system parameter vector $\theta_1$ is created. Next, the stage of feature selection using the binary gravitational algorithm is performed. The input data are the following parameters: vector $\theta_1$, population size $P$, the maximum number of iterations $T$, the initial value of gravitational constant $G_0$, coefficient $\alpha$ and small constant $\varepsilon$. The population of binary vectors $S = \{S_1, S_2, \ldots, S_P\}$ is randomly generated.
On the base of each particle, a fuzzy classifier is constructed to estimate the accuracy of the classification.

The mass of the $i$th particle is calculated with an account of the classification accuracy:

$$m_i(t) = \frac{(1 - E[S_j(t), \theta] - E[S_{\text{worse}}(t), \theta])}{(E[S_{\text{best}}(t), \theta] - E[S_{\text{worse}}(t), \theta])},$$

where $m$ is the mass of the particle, $i$ is the number of iteration, $S_{\text{worse}}(t)$ and $S_{\text{best}}(t)$ are vectors with the best and the worst accuracy on this iteration respectively. Particles that are the best in terms of accuracy, have a large mass and attract smaller particles. But since the forces of attraction act on all particles, the heavy ones also move, thereby realizing a local search.

The total force acting on the particle conveys its acceleration to it:

$$a_i^d(t) = \sum_{j=1, j \neq i}^P \text{rand}(0; 1) \cdot G[t] \cdot \frac{M_j[t] \cdot (S_j^d[t] - S_i^d[t])}{\|S_j[t] - S_i[t]\| + \varepsilon},$$

where $d = 1, |S_i|$ is the number of the element in the vector $S$, rand$(0; 1)$ is a random number in the interval $[0, 1]$, $M_j(t) = m_j(t) / \sum_{k=1}^P m_k(t)$ is the normalised value of the mass of the $j$th particle, $i = 1, P; G(t) = G_0 (t/T)^\alpha$ is the value of the gravitational constant at the iteration $t$.

The particle velocity is determined as follows:

$$V_i^d(t+1) = \text{rand}(0; 1) \cdot V_i^d(t) + a_i^d(t).$$

Elements of the vector are updated by converting the numerical value of velocity into a binary equivalent using the probability transformation function:

$$\begin{cases} 
\text{IF} \ (\text{rand}(0; 1) < \frac{1}{1 + e^{-V_i^d(t+1)}}) \ \text{THEN} \ S_i^d(t+1) = 0, \\
\text{ELSE} \ S_i^d(t+1) = 1
\end{cases}$$

The iteration of the algorithm completes after updating all elements of the vectors and calculating the accuracy values on the updated population. The output of the binary algorithm is $S_{\text{best}}$. $S_{\text{best}}$ is a vector of features with the highest classification accuracy.

The classifier is again constructed on the obtained feature vector and the vector $\theta_1$ is updated. At the parameter tuning stage, the continuous gravitational search algorithm generates a population $\theta = \{\theta_1, \theta_2, \ldots, \theta_P\}$ on the base of the vector $\theta_1$, with a certain deviation.

All variables for the continuous gravitational search algorithm are calculated similarly. But there are slight differences. The distance between the two particles in the binary algorithm is given as the Euclidean distance. In the continuous algorithm, it is given as the Hamming distance. After calculating the velocity in a continuous algorithm, the elements of the vectors are updated as follows:

$$\theta_i^d(t+1) := \theta_i^d(t) + V_i^d(t+1),$$

where $d = 1, |\theta|$ is the number of the element in the vector $\theta$. The output of this algorithm is the vector $\theta_{\text{best}}$, which has the highest classification accuracy.

Table 1 shows the pseudocodes of the described algorithms.

### 4. Experiment

The experiment consisted in constructing a fuzzy classifier with using the binary and continuous gravitational search algorithm on the KDD Cup 1999 data set. KDD Cup 1999 allows evaluating the effectiveness of the proposed algorithms for use as an element of the intrusion detection system. The data set consists of 41 features and contains information about network connections collected in the local network [2].

In addition to the percentage of correct classification as a criterion of accuracy, it is necessary to consider the percentage of errors of the first and second type. The type I error ($ER_1$) shows, what
percentage of instances with the label of a normal connection was accepted by the classifier as an attack, the type II error \( (ER_2) \) shows the percentage of incorrectly defined attack instances.

Table 1. The pseudocodes of the gravitational algorithms.

| The binary gravitational search algorithm | The continuous gravitational search algorithm |
|------------------------------------------|---------------------------------|
| 1. **Input**: \( \emptyset, P, T, G_0, \alpha, \varepsilon \). | 1. **Input**: \( S, P, T, G_0, \alpha, \varepsilon \). |
| 2. **Population** := \{\( S_1, S_2, \ldots, S_p \)\}; | 2. **Population** := \{\( \emptyset_1, \emptyset_2, \ldots, \emptyset_p \)\}; |
| 3. \( t := 1 \); | 3. \( t := 1 \); |
| 4. while \( (t \neq T) \) | 4. while \( (t \neq T) \) |
| 5. \( G[t] := G_0 \cdot (t / T)^n \); | 5. \( G[t] := G_0 \cdot (t / T)^n \); |
| 6. fori = 0 to \( P \) | 6. fori = 0 to \( P \) |
| 7. \( m_i[t] := \frac{(1 - E[S_i[t], \emptyset] - E[S_{worst}[t], \emptyset])}{(E[S_{worst}[t], \emptyset] - E[S_{worst}[t], \emptyset])}; \) | 7. \( m_i[t] := \frac{(1 - E[S_i[0], \emptyset] - E[S_{worst}[0], \emptyset])}{(E[S_{worst}[0], \emptyset] - E[S_{worst}[0], \emptyset])}; \) |
| 8. forj = 0 to \( P \) | 8. forj = 0 to \( P \) |
| 9. \( R(i, j) := |S_j - S_i|; \) | 9. \( R(i, j) := \|\emptyset_j - \emptyset_i\|; \) |
| 10. end for; | 10. end for; |
| 11. end for; | 11. end for; |
| 12. fori = 0 to \( P \) | 12. fori = 0 to \( P \) |
| 13. forj = 0 to \( P \) | 13. forj = 0 to \( P \) |
| 14. ford = 1 to \(|A|\) | 14. ford = 1 to \(|A|\) |
| 15. \( a'_i[t] := \sum_{j = 1}^{A} \text{rand}(0; 1) \cdot G[t] \cdot M_i[t] \cdot (S'_i[t] - S'_j[t]) / (|S_i[t] - S_j[t]| + \varepsilon) \); | 15. \( a'_i[t] := \sum_{j = 1}^{A} \text{rand}(0; 1) \cdot G[t] \cdot M_i[t] \cdot (\theta'_i[t] - \theta'_j[t]) / (|\emptyset_i[t] - \emptyset_j[t]| + \varepsilon) \); |
| 16. \( V^d_i[t + 1] := \text{rand}(0; 1) \cdot V^d_i[t] + a'_i[t]; \) | 16. \( V^d_i[t + 1] := \text{rand}(0; 1) \cdot V^d_i[t] + a'_i[t]; \) |
| 17. \( F^d_i[t + 1] := 1 / (1 + e^{-V^d_i[t + 1]}); \) | 17. \( \theta^d_i[t + 1] := \theta^d_i[t] + V^d_i[t + 1]; \) |
| 18. if \( \text{rand}(0, 1) \) \( \leq F^d_i[t + 1] \) then \( S'_i[t + 1] := 0 \); | 18. end for; |
| 19. else \( S'_i[t + 1] := 1 \); | 19. end for; |
| 20. end for; | 20. end for; |
| 21. end for; | 21. \( t := t + 1; \) |
| 22. end for; | 22. end for; |
| 23. \( t := t + 1; \) | 23. \( \text{output} \emptyset_{best} := \text{Search_best}(Population) \). |
| 24. end for; | 25. end for; |
| 26. \( \text{output} \emptyset_{best} := \text{Search_best}(Population). \) | |

Table 2 shows three fuzzy classifiers constructed using the gravitational search algorithm (\( E_L \) is the percentage of classification accuracy in the training sample, \( E_T \) is the percentage of accuracy on the test sample, and \( F \) is the number of features).

Table 2. The designed classifiers.

| # | \( F \) | Selected features | \( E_L \) | \( E_T \) | \( ER_1 \) | \( ER_2 \) |
|---|---|---|---|---|---|---|
| 1 | 30 | 2, 4, 5, 7, 8, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 | 98.29 | 98.40 | 0.96 | 0.64 |
| 2 | 27 | 2, 3, 4, 5, 7, 8, 9, 10, 13, 14, 15, 19, 20, 21, 23, 24, 25, 26, 27, 28, 30, 31, 34, 35, 36, 37, 38 | 98.47 | 98.48 | 0.76 | 0.76 |
| 3 | 12 | 8, 10, 12, 14, 16, 17, 20, 22, 24, 26, 30, 32 | 98.78 | 98.76 | 0.10 | 1.14 |

A choice of a classifier should be based on the requirements for the final system. If confidentiality is more important, there should be the smallest error of the second kind, so it is necessary to use the
classifier # 1. If the preference is given to availability, should choose the variation # 3. It is built on the least number of features, so it is the least complicated in the computational plan. The fuzzy classifier # 2 is a balanced alternative since the errors of the first and second type are equal and sufficiently small.

The validity of the constructed classifiers can be confirmed by comparison with classifiers-analogs. The comparison was conducted with the systems of intrusion detection, constructed by the following methods: based on the composition of the methods of selection of characteristics, discretization, information theory, decision tree and Bayesian classification – PKID + Cons + FVQIT, EMD + Cons + FVQIT, PKID + Cons + C4.5, EMD + INT + C4.5 and PKID + Cons + NB [10, 11]; by the method of principal components (PCA), by the nearest-neighbour method (k-NN), by the method of dynamic clustering (AP-Affinity Propagation) [12]; based on the composition of methods of boosting, sound functions and extreme training (MARK-ELM– Multiple Adaptive Reduced Kernel Extreme Learning Machine) [13]; based on the composition of the method of chaotically digging particles, the machine of reference vectors and the method of linear programming – TVCPSO-SVM, TVCPSO-MCLP [14].

As the comparison parameters, the number of features and the percentage of errors of the first and second type are chosen. The values of the selected parameters for the classifiers constructed with the using of the gravitational search algorithm and similar classifiers are given in Table 3.

| Algorithms | #F | ER1 | ER2 |
|------------|----|-----|-----|
| GSA_B + GSA_c # 1 | 30 | 0.96 | 0.64 |
| GSA_B + GSA_c # 2 | 27 | 0.76 | 0.76 |
| GSA_B + GSA_c # 3 | 12 | 0.10 | 1.14 |
| PKID+Cons+FVQIT | 6 | 7.27 | 0.48 |
| EMD+Cons+FVQIT | 7 | 5.5 | 1.54 |
| PKID+Cons+C4.5 | 6 | 5.92 | 1.92 |
| EMD+INT+C4.5 | 7 | 8.19 | 0.49 |
| PKID+Cons+NB | 6 | 9.82 | 0.42 |
| PCA | 41 | 19.59 | 0.7 |
| k-NN | 41 | 1.22 | 1.6 |
| AP | 41 | 1.01 | 1.6 |
| MARK-ELM | 41 | 1.00 | 0.23 |
| TVCPSO-MCLP-1 | 17 | 4.81 | 4.81 |
| TVCPSO-MCLP-2 | 41 | 2.77 | 2.41 |
| TVCPSO-SVM-1 | 17 | 0.87 | 2.97 |
| TVCPSO-SVM-2 | 41 | 3.29 | 4.51 |

The constructed classifier uses fewer features than six similar systems. By the type I error, the fuzzy classifier # 3 surpasses all considering systems. The other fuzzy classifiers concede only the TVCPSO-SVM-1. Only four systems outperform the fuzzy classifier # 1 by the type II error. Thus, it can be concluded that a fuzzy classifier designed using the binary and continuous gravitational search algorithms can be used in intrusion detection systems.

5. Conclusions
During the experiment, three fuzzy classifiers were constructed to determine the presence of an intrusion into the information transfer network. The results of their testing are rather different. The classifier, based on a large number of features, has the smallest type II error. A decrease in number of features leads to a reduction of a type I error but at the same time a type II error increases. Thus, when choosing a final classifier, it is required to determine which characteristic is more important. Nevertheless, the reduction in the number of features compared to the initial set is achieved, and the
errors do not exceed 1.14 percent. So it can be concluded that the method of constructing a fuzzy classifier with the using the binary gravitational search algorithm for feature selection and the continuous gravitational search algorithm for the tuning of the terms parameters can be used in intrusion detection systems.

Acknowledgments
This work was partially supported by the Ministry of Education and Science of Russia, Government Order no. 2.8172.2017/8.9 (TUSUR).

References
[1] Say L, Chou D, Gemmill A, Tunçalp Ö, Moller AB, Daniels J, Gülmezoglu AM, Temmerman M and Alkema L 2014 Global causes of maternal death: a WHO systematic analysis The Lancet Global Health. 2/6 pp 323–33
[2] KDD Cup 1999, available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[3] Kohavi R and John G H 1997 Wrappers for feature subset selection Artif. Intell 97/1 pp 273–324
[4] Dash M, Liu H 1997 Feature selection for classification. Intell. Data Anal. 1 pp 131–156
[5] Bolon-Canedo V, Sanchez-Marono N and Alonso-Betanzos A 2015 Feature Selection for High-Dimensional Data (New York: Springer) p 167
[6] Veerabhadrappa Rangarajan L 2010 Multi-level dimensionality reduction methods using feature selection and feature extraction International Journal of Artificial Intelligence & Applications 1/4 pp 54–68.
[7] Hodashinsky I A and Mekh M A 2017 Fuzzy classifier design using harmonic search methods Programming and Computer Software 43/1 pp 37–46.
[8] Hodashinsky I A, Bardamova I B and Bardamova M B 2017 Complex Assessment of Coagulation Parameters in Pregnant Women Using a Fuzzy Classifier Biomedical Engineering 51/3 pp 223–28
[9] Rashedi E, Nezamabadi-pour H and Saryazdi S 2009 GSA: A Gravitational Search Algorithm Information Sciences 179 pp 223–234
[10] Porto-Diaz I, Martinez-Rego D, Alonso-Betanzos A and Fontenla-Romero O 2009 Combining Feature Selection and Local Modelling in the KDD Cup 99 Dataset JCAVNS 5768 pp 824–833
[11] Bolon-Canedo V, Sanchez-Marono N and Alonso-Betanzos A 2011 Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset Expert Systems with Applications 38 pp 5947–57
[12] Wang W, Guyet A T, Quiniou R, Cordier M-O, Masseglia F and Zhang X 2014 Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks Knowledge-Based Systems 70 pp 103–17
[13] Fossaceca J M, Mazzuchi T A and Sarkani S 2015 MARK-ELM: Application of a Novel Multiple Kernel Learning Framework for improving the robustness of Network Intrusion Detection Expert Systems with Applications 42 pp 4062–80
[14] Bamakan S M H, Wang H, Yingjie T, and Shi Y 2016 An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization Neurocomputing 199 pp 90–102