CONSTRUCTING A FUZZY NETWORK INTRUSION CLASSIFIER BASED ON DIFFERENTIAL EVOLUTION AND HARMONIC SEARCH

I. A. Hodashinsky, M. A. Mech

Department of Complex Information Security Tomsk State University of Control Systems and Radioelectronics (TUSUR), Tomsk, Russia

ABSTRACT

This paper presents a method for constructing intrusion detection systems based on efficient fuzzy rule-based classifiers. The design process of a fuzzy rule-based classifier from a given input-output data set can be presented as a feature selection and parameter optimization problem. For parameter optimization of fuzzy classifiers, the differential evolution is used, while the binary harmonic search algorithm is used for selection of relevant features. The performance of the designed classifiers is evaluated using the KDD Cup 1999 intrusion detection dataset. The optimal classifier is selected based on the Akaike information criterion. The optimal intrusion detection system has a 1.21% type I error and a 0.39% type II error. A comparative study with other methods was accomplished. The results obtained showed the adequacy of the proposed method.

KEYWORDS

Intrusion detection; fuzzy classifier; differential evolution; feature selection; binary harmonic search; Akaike information criterion

1. INTRODUCTION

Network communication has become an integral part of everyday life, both for large organizations and ordinary people. Yet, the number of network security threats has increased as well, jeopardizing the confidentiality, integrity, and availability of information and undermining the operability of information systems.

Intrusion detection systems are designed to analyze (recognize) network traffic and classify it, based on a certain set of features, as normal or abnormal. Such recognition systems can be constructed based on various methods: Bayes classifiers, support vector machines, metrical classifiers, neural networks, and fuzzy systems.

The main advantage of fuzzy systems over other solutions is a simple interpretation of results, which improves the reliability of classification and makes it easier to diagnose and fix errors that may occur when designing the intrusion detection system [1,2].

This paper presents a method for constructing intrusion detection systems based on an efficient fuzzy classifier.

2. FORMULATION OF THE PROBLEM

Assume we have a universum$U = (A, C)$, where $A = \{x_1, x_2, \ldots, x_n\}$ is the set of input features and $C = \{c_1, c_2, \ldots, c_m\}$ is the set of classes. Let $\mathbf{X} = x_1 \times x_2 \times \ldots \times x_n \in \Re$ be an n-dimensional feature

DOI: 10.5121/ijcnc.2018.10208

space. Each object u of the universum$U$ is characterized by its own feature vector. Thus, the classification problem is reduced to predicting the class of the object u based on its feature vector. In this case, 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:

$$f : \Re^n \to [0,1]^m .$$

The fuzzy classifier uses production rules of the form

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

whereA_{ki} is the fuzzy term characterizing the k-th feature in the i-th fuzzy rule ($i \in [1, R]$) and R is the number of fuzzy rules.

In this work, the class is assigned based on the principle "winner takes all:"

$$\text{class} = c_{j^*},$$

$$j^* = \arg\max_{1 \le j \le m} \beta_j$$

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

Assume we have an observations table $\{(\mathbf{x}_p; c_p), p = 1 ,..., z\}$. With the unit function

$$delta(p, \boldsymbol{\theta}) = \begin{cases} 1, & \text{if } c_p = f(c_p, \boldsymbol{\theta}) \\ 0, & \text{else} \end{cases}, \; p = 1,2,..., z$$

defined, the fitness function (measure of classification accuracy) can be written as

$$E(\boldsymbol{\theta}) = \frac{\sum_{p=1}^{z} delta(p, \boldsymbol{\theta})}{z}.$$

The problem of constructing the fuzzy classifier is reduced to finding the maximum of the fitness function in the parameter space $\boldsymbol{\theta} = \|\theta^1, \theta^2, \ldots, \theta^D\|$:

$$\max(E(\boldsymbol{\theta})), \theta^i \in \{ \Theta^i : \Theta^i_{min} < \Theta^i < \Theta^i_{max}, i = 1,2, \ldots, D\},$$

where D is number of parameters to be optimized; Θ^i is the value of the parameter θ^i on the interval $[\Theta^i_{min}, \Theta^i_{max}]$, while Θ^i_{min} and Θ^i_{max} are the lower and upper limits for all θ^i, respectively.

To optimize the parameters $\boldsymbol{\theta}$, the differential evolution algorithm is used, while the binary harmonic search algorithm is used to select relevant features.

3. THEORETICAL PART

A. Differential Evolution

The design process of fuzzy rule-based classifiers from a given input-output data set can be presented as a feature selection and parameter optimization problem. A rule parameter optimization is the optimization of the location and form of the curves which describe the fuzzy terms. A parameter optimization remains in any case a nonlinear optimization problem. Parameter optimization is parameter tuning framework that tries to cope with the stochastically distributing results of search heuristics by repeated evaluations. We solve parameter optimization problem using population-based direct global optimization method called Differential Evolution.

Differential evolution, introduced by Storn and Price [3], is a simple yet efficient evolutionary algorithm for global optimization problems in continuous search domain. It deals with a real-coded population, instead of a binary, and devises its own crossover and mutation in the real space. Differential evolution begins by initializing a population of candidate solutions. The quality of each solution is evaluated by a fitness function. Differential evolution selects a set of solutions as parents and evolves the parents through mutation and crossover, which generate a child. A selection process is applied to compare each pair of a predefined parent and its corresponding child in terms of the fitness function, and then the promising one is selected to survive to the next iteration. A child that survives to the next iteration is called a successful solution. Differential evolution repeats this procedure until a predefined termination criterion is reached. The best solution found by this procedure is expected to be a near-optimal solution for the optimization problem [4].

Differential evolution has the following parameters: the number of iterations (N), the number of chromosomes in a population (S), the crossover probability (Cr), and the coefficient used to generate a new chromosome (F).

Below is the pseudocode of the differential evolution based parameter optimization algorithm for the fuzzy rule-based classifier.

Input: N,S,F
Output:$\boldsymbol{\theta}_{best}$.
$Popul$:= $\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2, \ldots,\boldsymbol{\theta}_S\}$;
  loop until (N>0)
    loopfor $p$from 1 to S
      $\boldsymbol{\theta}_{cur}$:= $Popul[p]$;
      $\boldsymbol{\theta}_a,\boldsymbol{\theta}_b$:= Random_choose($Popul$);
      θ_{best}:= Search_best($Popul$);
CR := rand(0;1);
      loopfor d from 1 to $|\boldsymbol{\theta}_i|$
        if (rand(0; 1) < CRthen
        $\boldsymbol{\theta}_{new}[d]$:=$\boldsymbol{\theta}_{best}[d]+F*(\boldsymbol{\theta}_a[d]-\boldsymbol{\theta}_b[d])$;
        else $\boldsymbol{\theta}_{new}[d]$:=$\boldsymbol{\theta}_{cur}[d]$;
      end loop
        if ($E(\boldsymbol{\theta}_{new})$ <$E(\boldsymbol{\theta}_{cur})$) then
          $\boldsymbol{\theta}_{cur}$:= $\boldsymbol{\theta}_{new;}$
      N:=N-1;
    end loop
  end loop
output$\boldsymbol{\theta}_{best}$:= Search_best($Popul$).

B. Feature selection. Binary harmonic search

In intrusion detection systems, features may contain false correlations, which hinder the underlying process and in general, the learning task to be carried out. Some features may be irrelevant and some others may be redundant. These extra features can increase computation time, and can have an impact on the accuracy of the classifier built. Feature selection is primarily performed to select relevant and informative features, but it can have other motivations, including general data reduction, feature set reduction, performance improvement and better data understanding. There are two main models dealing with feature selection: filter methods and wrapper methods. Wrapper methods optimize a classifier as part of the selection process, while filter models rely on the general characteristics of the training data to select the best features with

the independence of any classifier. Wrapper methods use a classifier and a search technique to score subsets of features according to their predictive power [5].

The Binary Harmonic Search is used in order to optimize the feature subspace of fuzzy classifier. This feature selection strategy is wrapper type approach.

Harmony Search is a meta-heuristic algorithm firstly developed by Geem et al. in 2001 [6]. It imitates the musician seeking to find pleasing harmony determined by an aesthetic standard as the optimization method seeks to find the global optimal solution determined by an objective function. Unlike the traditional optimization algorithms based on the gradient and Newton's methods, harmony search uses a stochastic search instead of a gradient-based search, and therefore the derivative information is unnecessary In 2005, Geem firstly adopted the standard harmony search with binary-coding to solve water pump switching problems in which the pitch adjustment operator was discarded [7].

In binary harmonic search, an individual is formed by the binary-string. The binary harmonic search has the following parameters: the constants used to generate a new vector ($HMCR \in [0; 1]$ – the probability of picking up a value from harmony memory and $PAR \in [0; 1]$ – the pitch adjustment operator), the harmonic memory size (*HMS*), and the number of iterations (*N*).
Below is the pseudocode of the Binary Harmonic Search based feature selection algorithm for the fuzzy rule-based classifier [8].

Input: *N*, *HMCR*, *PAR*.
Output:$\mathbf{A}_{best}$.
  $HM := \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{HMS}\}$;
  loop until ($N > 0$)
    $\mathbf{A}_r$:= Random_choose(*HMS*)
    loop for *d* from 1 to *N*
      if (rand(0; 1) <*HMCR*) then
        $\mathbf{A}_{new}[d] := \mathbf{A}_r[d]$
        if (rand(0; 1) <*PAR*) then
          $\mathbf{A}_{new}[d] := \mathbf{A}_r[d]$
      elseif (rand(0; 1) ≤ 0.5) then
          $\mathbf{A}_{new}[d] := 0$;
      else$\mathbf{A}_{new}[d] := 1$;
    end loop;
      if ($E(\mathbf{A}_{new}) > E(\mathbf{A}_{worst})$) then
        $\mathbf{A}_{worst} := \mathbf{A}_{new}$;
    $N := N - 1$;
  end loop;
  output$\mathbf{A}_{best}$:= Search_best(*HMS*).

Table I. Test Results Of Fuzzy Intrusion Detection Classifiers

Characteristic	Classifier ID				
	1	**2**	**3**	**4**	**5**
Number of features F_S	24	19	17	22	10
Percentage of correctly classified instances on the training sample	98.74	99.1	97.85	98.27	98.94
Percentage of correctly classified instances on the test sample	98.71	99.08	97.84	98.25	99.05
Type I error ER_1	1.58	1.06	1.01	1.14	1.21
Type II error ER_2	0.12	0.36	6.87	4.25	0.39

4. EXPERIMENTAL RESULTS

The performance of the fuzzy classifier constructed using the algorithms described above was evaluated on the KDD Cup 1999 intrusion detection dataset [9]; this dataset contains 41 features characterizing different types of network connection, including 23 network attack classes and one normal connection class.

The value of the differential evolutionparameters is
$N = 1000$; $S = 20$; $F = 0.7$.
The value of the binary harmonic search parameters is
$HMCR = 0.8$; $PAR = 0.25$; $HMS = 20$; $N = 20000$.

Using the KDD' 99 dataset, several intrusion detection classifiers were constructed that differed in the number of features, accuracy on training and test data, and value of type I and II errors. Table I shows the test results of five fuzzy classifiers.

5. DISCUSSION

The identification of fuzzy classifiers from training data needs to consider an important tradeoff well known in the statistical modeling community – the tradeoff between data fitness and model complexity [10]. To find the optimal classifier, the Akaike information criterion [11] adapted for the problem at hand was used:

$$AIC = \ln(ER_1 + k \cdot ER_2) + \frac{2}{z}(1 + Fs)$$

whereFs isthenumber of features; z isthe volume of observations table and k is the coefficient setting the priority of the type II error over the type I error (in this work,$k = 2$). Based on the Akaike information criterion, classifier 5 was found to be the optimal one.
We compare the performance of the proposed method with other classification methods for the intrusion detection case study.

In [5, 12], an approach for intrusion detection in computer networks is introduced. This method – called FVQIT (Frontier Vector Quantization using Information Theory) – uses a modified clustering algorithm to split up the feature space into several local models, in each of which the classification task is performed independently. The FVQIT Method is used in combination with the discretization algorithms – PKID (Proportional *k*-Interval Discretization) and EMD (Entropy Minimization Discretization) – and filter methods (Consistency-based Filter and INTERACT). INTERACT is a method based on the interaction between features, from an Information Theory point of view; while Consistency-based follows a more classical approach, evaluating consistency between classes [12]. In the combinations (discretizer + filter + classifier) used also two classifiers that can deal with both numerical and symbolic attributes and no conversion is needed: C4.5 and naive Bayes (NB).

In [13], an effective intrusion detection framework by using an adaptive, robust, precise optimization method, namely, time-varying chaos particle swarm optimization (TVCPSO) to simultaneously do parameter setting and feature selection for multiple criteria linear programming (MCLP) and support vector machine (SVM) is proposed.

In [14], authors used a novel clustering algorithm, Affinity Propagation (AP) and its extension in streaming environments. AP clustering has no need to define the number of clusters beforehand and this is an important advantage for autonomic intrusion detection because it is very difficult to

have a priori knowledge for the data, especially for a very large amount of streaming data that always evolve over time [14].
In [14], authors compared Affinity Propagation model to other methods: *k*-NN (*k*-Nearest Neighbor) model and PCA (Principal Component Analysis) model.

The performance ofthe fuzzy rule-based classifier is compared with that of several classification methods in Table II. According to Table II, it is clear that using our method resulted in more reliable intrusion detection systems.

6. CONCLUSIONS

In this paper, the method for constructing fuzzy network intrusion classifiers, which includes the criterion for selecting the optimal fuzzy rule-based classifier and the algorithms for feature selection and parameter optimization, has been proposed. The method based on the combination of differential evolution, binary harmonic search and Akaike information criterion that maintains the performance results of the classifiers but using a reduced set of features. In order to help the developers of fuzzy rule-based classifiers to strive for a balance between the two conflicting modeling objectives, we propose Akaike information criteria for constructing optimal fuzzy rule-based classifiers. The experimental investigation on the KDD Cup 1999 intrusion detection dataset has been carried out. The experimental results have confirmed the efficiency of the proposed method. Parameters of differential evolution and binary harmonic search set in the research is not optimal, thus the future work

Table 2. Comparing The Performance Of Classifiers

Classifier	F_S	ER_1	ER_2
Our method	10	1.21	0.39
PKID+Cons+FVQIT [10]	6	7.27	0.48
EMD+Cons+FVQIT [10]	7	5.5	1.54
PKID+Cons+C4.5 [4]	6	5.92	1.92
EMD+INT+C4.5 [4]	7	8.19	0.49
PKID+Cons+NB [4]	6	9.82	0.42
TVCPSO-MCLP [11]	17	4.81	4.81
TVCPSO-MCLP [11]	41	2.77	2.41
TVCPSO-SVM [11]	17	0.87	2.97
TVCPSO-SVM [11]	41	3.29	4.51
PCA [12]	41	19.59	0.7
k-NN [12]	41	1.22	1.6
AP [12]	41	1.01	1.6

Should optimize the parameters according to this methods parameters and different training dataset.

In the future work, we will develop a practical real-time system for high-speed network intrusion detection. How to dynamically and automatically update the detection models for addressing the problem of concept drift is also being investigated. From a further perspective, we intend to test different methods for instance selection.

ACKNOWLEDGMENT

The Ministry of Education and Science of the Russian Federation, agreement no. 8.9628.2017/8.9, supported this work.

REFERENCES

[1] I.A. Hodashinsky and I.V. Gorbunov, "Algorithms of the tradeoff between accuracy and complexity in the design of fuzzy approximators," Optoelectronics, Instrumentation and Data Processing, vol. 49, pp. 569-577, November 2013.

[2] A.E. Anfilofiev, I.A. Hodashinsky and O.O.Evsutin, "Algorithm for tuning fuzzy network attack classifiers based on invasive weed optimization", 2014 Dynamics of Systems, Mechanisms and Machines, Dynamics 2014 – Proceedings. January 2015.

[3] R. Storn and K.V. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," (1995) Technical Report TR-95-012. ICSI (March 1995). ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.pdf

[4] S.-M. Guo, C.-C. Yang, P.-H. Hsu, and J.S.-H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," IEEE Transactions on Evolutionary Computation, vol. 19, NO. 5, pp. 717-730, October 2015.

[5] V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset," Expert Systems with Applications, vol. 38, pp. 5947-5957, May 2011.

[6] Z.W. Geem, J. Kim, and G. Loganathan, "Music-inspired optimization algorithm harmony search," Simulation, vol. 76, pp. 60-68, 2001.

[7] K. Lee and Z.W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 3902-3933, September 2005.

[8] I.A. Hodashinsky and M.A.Mekh "Fuzzy classifier design using harmonic search methods," Programming and Computer Software, vol 43, pp. 37-46, January 2017.

[9] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, May 2016.

[10] J.Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," IEEE Transactions on Fuzzy Systems, vol. 6, N. 3, pp. 362-372, August 1998.

[11] H. A. Akaike, "New look at the statistical model identification," IEEE Transactions on Automatic Control, vol. AC19, pp. 716-723, 1974.

[12] I. Porto-Diaz, D.Martinez-Rego, A. Alonso-Betanzos, and O.Fontenla- Romero, "Combining feature selection and local modelling in the KDD Cup 99 Dataset," ICANN 2009, Part I, LNCS, vol. 5768, 2009, pp. 824-833.

[13] S.M.H.Bamakan, H. Wang, T. Yingjie, and Y.Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," Neurocomputing, vol. 199, pp. 90-102, July 2016.

[14] W. Wang, A. T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia, and X. Zhang, "Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks," Knowledge-Based Systems, vol. 70, pp. 103-117, November 2014.