Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 2373–2391
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection

Chi-Ho Tsang, Sam Kwong*, Hanli Wang

Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, PR China

Received 7 June 2005; received in revised form 10 July 2006; accepted 13 December 2006

Abstract

Classification of intrusion attacks and normal network traffic is a challenging and critical problem in pattern recognition and network security. In this paper, we present a novel intrusion detection approach to extract both accurate and interpretable fuzzy IF–THEN rules from network traffic data for classification. The proposed fuzzy rule-based system is evolved from an agent-based evolutionary framework and multi-objective optimization. In addition, the proposed system can also act as a genetic feature selection wrapper to search for an optimal feature subset for dimensionality reduction. To evaluate the classification and feature selection performance of our approach, it is compared with some well-known classifiers as well as feature selection filters and wrappers. The extensive experimental results on the KDD-Cup99 intrusion detection benchmark data set demonstrate that the proposed approach produces interpretable fuzzy systems, and outperforms other classifiers and wrappers by providing the highest detection accuracy for intrusion attacks and low false alarm rate for normal network traffic with minimized number of features.

Keywords: Fuzzy classifier; Genetic algorithms; Multi-objective optimization; Feature selection; Intrusion detection

1. Introduction

Intrusion detection based on statistical pattern recognition approaches has attracted a wide range of interest over the last 10 years in response to the growing demand of reliable and intelligent intrusion detection systems (IDS), which are required to detect sophisticated and polymorphous intrusion attacks. In general, intrusion detection approaches are usually categorized into misuse and anomaly detection approaches in the literature. Misuse detection approach can reliably identify intrusion attacks in relation to the known signatures of discovered vulnerabilities. However, emergent intervention of security experts is required to define accurate rules or signatures, which limits the application of misuse detection approach to build intelligent IDS. On the other hand, the anomaly detection approach usually deals with statistical analysis and pattern recognition problems. It is able to detect novel attacks without a priori knowledge about them if the classification model has the generalization capability to extract intrusion pattern and knowledge during training. Unfortunately, it commonly suffers from high false positive rate (FPR) on classifying normal network traffic nowadays. To overcome the anomaly intrusion detection problem, the data mining [1], machine learning [2] and immune system [3] approaches have been proposed in recent years.

Learning classification rules from network data is one of the most effective methods to automate and simplify the manual development of intrusion signatures, and predict novel attacks if the generalized knowledge can be extracted from data. One of the key challenges in building an anomaly rule-based IDS is to ensure that it can automatically extract optimal classification rules from training data, and the extracted rules should be (i) accurate and sufficient to detect both known and unseen intrusion attacks and recognize normal network traffic, and (ii) linguistically interpretable for human comprehension. To extract rule-based knowledge from network data, Lee et al. [4] propose to apply association rules to capture the behaviours and relations in programs execution and user activities, and use frequent episodes to model the sequential patterns in system

* Corresponding author. Tel.: +852 2788 7704; fax: +852 2788 8614.
E-mail address: cssamk@cityu.edu.hk (S. Kwong).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.12.009

audits or network data. However, since the quantitative features in the intrusion data are partitioned into the interval with crisp boundary, there might exist a sharp boundary problem for classification. In order to solve this problem, the fuzzy logic [5], which provides the partial membership in set theory, is applied in Ref. [6] to integrate with the association rules and frequent episodes. The application of fuzzy logic in intrusion detection can also be found in Ref. [7], which effectively detects port scanning and denial-of-service attacks.

Genetic algorithm (GA) [8] has been successfully applied to solve many combinatorial optimization problems. The application of GA to the evolution of fuzzy rules can be found in Refs. [7,9] for intrusion detection. In Ref. [9], a simple GA is applied to generate and evolve the fuzzy classifiers that use complete expression tree and triangular membership function for the formulation of chromosome. To evaluate the fitness of individual solutions, the weighted sum of fitness values of multiple objective functions is proposed in Ref. [9] where the proposed weights are user-defined and cannot be optimized dynamically for different cases. In Ref. [10], a large number of fuzzy rules are first generated for each class with the use of fuzzy association rules. After that, a boosting GA based on the iterative rule learning approach is applied for each class to search its fuzzy rules required for classification, in which the rules can be extracted and included in the system for evaluation. However, it only optimizes classification accuracy and omits the necessity of interpretability optimization. In Ref. [11], a simple GA is employed as the searching strategy in a feature selection wrapper that applies RIPPER [12] as the induction algorithm for rule learning and classification. The above-mentioned works have somehow successfully demonstrated the effectiveness of applying GA to select feature subset and generate fuzzy rule-based IDS, however, only on the optimization of classification accuracy.

In general, there is always a trade-off between the accuracy and interpretability such that the acquisition of fuzzy IF–THEN rules, which achieves good accuracy, does not imply the fuzzy system is interpretable for human comprehension. As discussed in Refs. [13,14], besides the importance of classification performance, it is also desirable to obtain highly interpretable knowledge in IDS to assist security experts for intrusion analysis. Therefore, the optimizations of both accuracy and interpretability should be necessarily taken into account for building anomaly rule-based IDS. To achieve this goal, a multi-objective genetic fuzzy intrusion detection system (MOGFIDS) is proposed in this work, which applies an agent-based evolutionary computation framework to generate and evolve an accurate and interpretable fuzzy knowledge base for classification. To the best of our knowledge, this is the first work in applying multi-objective genetic fuzzy system concerning with both accuracy and interpretability for anomaly rule-based intrusion detection.

In addition, the proposed MOGFIDS can be considered as a genetic wrapper that searches for a near-optimal feature subset from network traffic data. This helps to reduce the computational overhead for classification and improve the generalization capability of MOGFIDS. Feature selection (FS), which is known to be an NP-hard problem [15], has been extensively studied in the last two decades. Given a set of N features, the goal is to select a desired subset of size M from 2^N possible subsets in order to minimize the classification error and alleviate the curse of dimensionality for computational cost. In general, the optimality of feature subset can substantially improve the interpretability of rule-based classifiers since the optimal minimal number of features minimizes the number of classification rules generated from data. The FS techniques can be broadly classified into filter-based and wrapper-based approaches in the supervised learning paradigm. The filter-based approaches select features using estimation criterion based on the statistics of learning data, and are independent of the induction classifier. The wrapper-based approaches employ induction classifier as a black box using cross-validation or bootstrap techniques to evaluate the feature subset candidates suggested by different search algorithms, such that the accuracy of the classifier can often be maximized. Wrapper-based approaches generally produce better subsets than filter-based approaches, but they are more computationally expensive than filter-based approaches due to the repeated runs of classifier, in particular for very high-dimensional feature domains. As there is no single FS technique that has proven superior for all problem domains, the first sub-goal of this work is to search for a near-optimal feature subset using some well-known filter-based approaches as a baseline reference, and the second sub-goal is to evaluate the effectiveness of MOGFIDS in comparison with some wrapper-based approaches, in searching near-optimal feature subset for intrusion detection.

The rest of this paper is organized as follows. Section 2 highlights the interpretability issues of genetic fuzzy rule-based system (GFRBS). Our proposed multi-objective genetic fuzzy rule-mining approach is described in Section 3 in detail. Section 4 discusses the experimental results including the performance comparisons of MOGFIDS with other feature selection approaches for intrusion detection. Finally, we draw the conclusions in Section 5.

2. Genetic fuzzy rule-based systems and interpretability

Fuzzy rule-based systems, inspired by the fuzzy set theory [5], have been successfully applied to solve many complex and non-linear problems by constructing fuzzy IF–THEN rules for classification and modeling control. GFRBSs employ evolutionary approach to learn and extract knowledge from training data. The optimization criteria in GFRBS include linguistic variables, parameters of fuzzy membership functions, fuzzy rules and the number of rules. In traditional GFRBS, the classification performance and interpretability (also known as transparency), which are often contradictive to each other, are not addressed simultaneously. Redundant fuzzy rules and fuzzy sets, as well as inappropriate fuzzy set topology would be undesirably constructed if the interpretability criterion is not optimized. The poor interpretability of such fuzzy systems can potentially degrade the performance as well as the usefulness of fuzzy rule-based IDS. In this section, we briefly discern the interpretability with the following factors, which are discussed in our previous work [16] in detail.

2.1. Completeness and distinguishability

Partitioning of fuzzy sets for each fuzzy variable should be complete and well distinguishable. For each input variable x_i in a feature vector $X=[x_1, x_2, \ldots, x_n]^{\mathrm{T}}$, there exists M_i fuzzy sets represented by $A_1(x), A_2(x), \ldots, A_{Mi}(x)$. Partitioning of fuzzy sets is complete if the following condition holds true, in which at least one fuzzy set is triggered for each input:

$$\forall x_i \in U_i, \quad i \in [0, \ldots, n]; \\ \exists A_j(x_i) > 0, \quad j \in [1, \ldots, M_i], \tag{1}$$

where U_i is the universe of x_i. Completeness and distinguishability can be interpreted by the fuzzy similarity measure [17], which identifies (i) the similarity between two fuzzy sets for a fuzzy variable; (ii) the similarity of a fuzzy set to the universal set U; and (iii) the similarity of a fuzzy set to a singleton set. The similarity between two fuzzy sets A and B can be calculated using the following computationally efficient method:

$$S(A, B) = \frac{\sum_{j=1}^{m}[u_A(x_j) \wedge u_B(x_j)]}{\sum_{j=1}^{m}[u_A(x_j) \vee u_B(x_j)]} \tag{2}$$

on a discrete universe $U=\{x_j | j=1, 2, \ldots, m\}$ where $\wedge$ and $\vee$ represent the minimum and maximum operations, respectively. If $S(A, B)$ is larger than a given threshold, then the partitioning of these two fuzzy sets are not well distinguishable from each other resulting in a bad topology.

2.2. Consistency, compactness and utility

Fuzzy rules are consistent if they are not contradictive. If two or more rules with similar antecedents are triggered simultaneously, then their consequents should also be similar. The detailed issues about consistency can be found in Ref. [18]. Degree of consistency can depend on the inclusion relation. Given two fuzzy rules R_i and R_j:

R_i: If x_1 is $A_{i1}(x_1)$ and x_2 is $A_{i2}(x_2)$
and $\ldots x_n$ is $A_{in}(x_n)$,
then y_1 is $B_{i1}(y_1)$ and $\ldots y_m$ is $B_{im}(y_m)$,
R_j: If x_1 is $A_{j1}(x_1)$ and x_2 is $A_{j2}(x_2)$
and $\ldots x_n$ is $A_{jn}(x_n)$,
then y_1 is $B_{j1}(y_1)$ and $\ldots y_m$ is $B_{jm}(y_m)$.

Suppose their antecedents are compatible with an input vector, and the antecedents of R_i are included in those of R_j, then R_i should be updated with a larger weight than R_j to calculate the output. The traditional weight (fire-strength) u_i of the ith rule can be defined as follows:

$$u_i(x) = u_{A_{i1}}(x_1) \wedge u_{A_{i2}}(x_2) \wedge \cdots \wedge u_{A_{in}}(x_n), \\ i = 1, 2, \ldots, R, \tag{3}$$

where R is the number of fuzzy rules in the rule base. We introduce an inclusion factor which is given by

$$\lambda_i = \prod_{R_k \subseteq R_i} (1 - u_k(x)), \quad k = 1, 2, \ldots, R, \quad k \neq i. \tag{4}$$

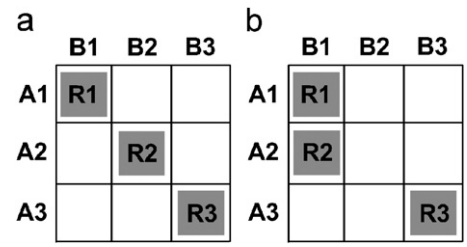

Fig. 1. Example of fuzzy system has three fuzzy rules and two input features each of which has three fuzzy sets. (a) Sufficient utility and (b) insufficient utility.

Therefore, the weight of the rule R_i with the inclusion factor can be updated as

$$\breve{u}_i = \lambda_i u_i(x), \quad i = 1, 2, \ldots, R. \tag{5}$$

Compactness of fuzzy systems plays an important role in the interpretability of fuzzy systems [16,18–21]. A compact fuzzy system indicates that it is easy to be comprehended. There are three aspects which are closely related to the compactness of fuzzy systems [20]: (i) a small number of fuzzy sets for each fuzzy variable, (ii) a small number of fuzzy rules in rule base, and (iii) a small number of conditions in the rule premise. Regarding the number of fuzzy sets, it is relatively easier for users to discern a fuzzy variable with small number of linguistic labels. The second aspect of compactness is the number of fuzzy rules. It is easier for users to comprehend and recognize a compact fuzzy rule base than a rule base with more fuzzy rules. Compactness of fuzzy rules becomes more important when the system involves a large number of dimensions [18,19]. The third aspect of compactness is the number of conditions in the antecedent part of fuzzy rules. If unrepresentative fuzzy sets are not used in the fuzzy rules then the fuzzy system can become more compact and thus easier to be understood. Based on the foregoing analysis, high compactness is desired to improve the interpretability of fuzzy systems and reduces the computational cost of the fuzzy inference process. Finally, even if the fuzzy system can be complete and distinguishable, each fuzzy set may not be used by at least one fuzzy rule. As depicted in Fig. 1(a), a fuzzy system is of sufficient utility if all the fuzzy sets (A_1–A_3, B_1–B_3) are utilized as antecedents or consequents by fuzzy rules (R_1–R_3). On the contrary, the utility is of insufficient if there is at least one fuzzy set such as B_2 that is not utilized by any rule in Fig. 1(b). To better utilize fuzzy systems, it is necessary to remove the unused fuzzy sets from rule base.

3. Agent-based genetic fuzzy rule-based knowledge extraction

3.1. Overview of MOGFIDS

In this work, an agent-based evolutionary computation framework is proposed to construct a GFRBS concerning with both accuracy and interpretability for IDS. The MOGFIDS can be viewed as a multi-agent learning system, which consists of the fuzzy set agent (FSA) and arbitrator agent (AA). The

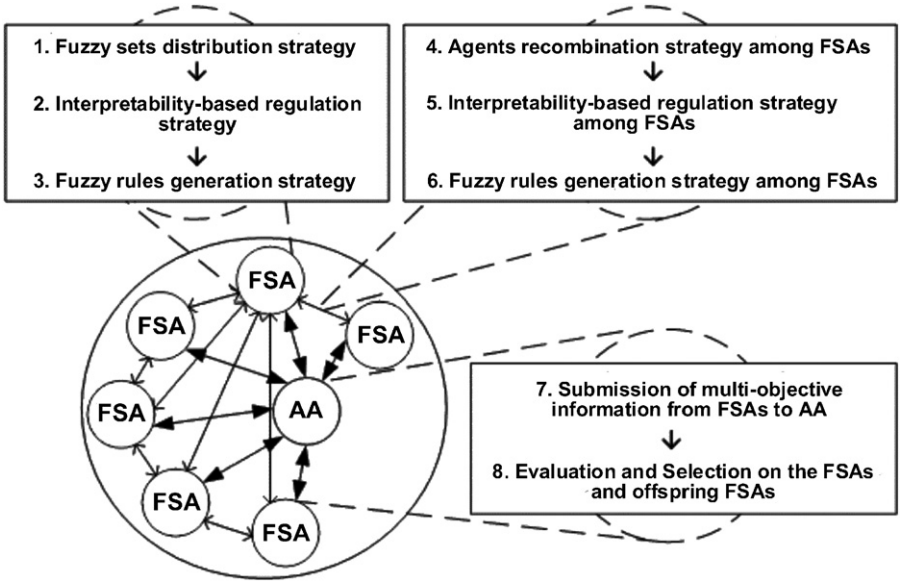

Fig. 2. Multi-agent system framework and highlighted processes of agent evolution.

agent-based framework is illustrated in Fig. 2 and summarized as follows. Each autonomous FSA employs three main strategies to construct and evolve its fuzzy systems. It initializes its own fuzzy sets information using the fuzzy sets distribution strategy. Then the interpretable fuzzy rule base is generated through the use of interpretability-based regulation strategy and fuzzy rules generation strategy according to the initialized fuzzy sets. In order to find the global optimal fuzzy rule base, in each generation FSAs generate their offspring by cooperatively exchanging their fuzzy sets information, and applying genetic crossover and mutation operations to the chromosomes of hierarchical formulation. The fuzzy rule bases of the offspring FSAs are generated using the previously mentioned strategies in a similar manner. Finally, FSAs submit their fitness values about the accuracy and interpretability to the AA for evaluation. Different from the parallel GA, our agent-based approach does not exchange individual FSA but only exchanges the multi-objective information about the fuzzy sets. The AA applies the robust multi-objective optimization algorithm NSGA-II [22] to evaluate the parent and offspring FSAs based on their fitness assessments in both accuracy and interpretability criteria. As a result, the elitist FSAs are retained and the low-fitness FSAs are removed in each generation.

3.2. Intra-behaviors of FSA

In this section, the three main intra-behavioral strategies are discussed in detail.

3.2.1. Fuzzy sets distribution strategy

Minimal number of fuzzy sets and rules can be effectively searched without a priori knowledge of the fuzzy set topology by the use of hierarchical GA (HGA) [23–25]. In HGA each chromosome is formulated in a multi-layer structure consisting of control genes and parameter genes. As depicted in Fig. 3,

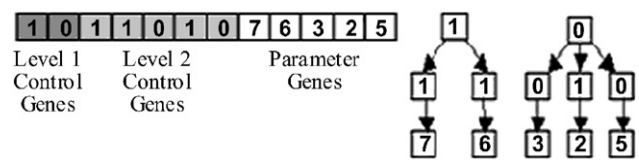

Fig. 3. Example of hierarchical chromosome. Three-level gene structure has a phenotype value (7,6) as activated by the top-level control genes.

the activations of parameter genes are managed by the control genes, e.g., a control gene with binary value "1" can activate its associated parameter genes, and so forth. As the chromosome and genotype structure are not fixed in HGA, it can perform well in the structure and topology optimization, and also help optimize the distribution of fuzzy sets.

To sufficiently represent each fuzzy variable x_i, a possible maximal number of fuzzy sets M_i is determined. For N-dimensional problem, totally $P = M_1 + M_2 + \cdots + M_N$ possible fuzzy sets require P binary-valued control genes to manage the activation of their parameter genes. Gaussian combinational membership functions (abbreviated as Gauss2mf) can cover the universe sufficiently and enforce the completeness of fuzzy systems, hence it is applied to formulate the antecedent fuzzy sets in parameter genes. The Gauss2mf is defined by four parameters a_1, a_2, a_3 and a_4, where a_1, a_2, a_3 and a_4 are the lower bound, left center, right center and upper bound of the definition domain, respectively ($a_1 \leqslant a_2 \leqslant a_3 \leqslant a_4$). The application of HGA in Gauss2mf is shown in Fig. 4 for illustration. The FSAs randomly initialize the values of both control genes and parameter genes at the beginning of run.

3.2.2. Interpretability-based regulation strategy

Since the distinguishability of fuzzy partitioning cannot be guaranteed in the initialization of fuzzy sets, the interpretability-based regulation strategy is applied in the following steps to

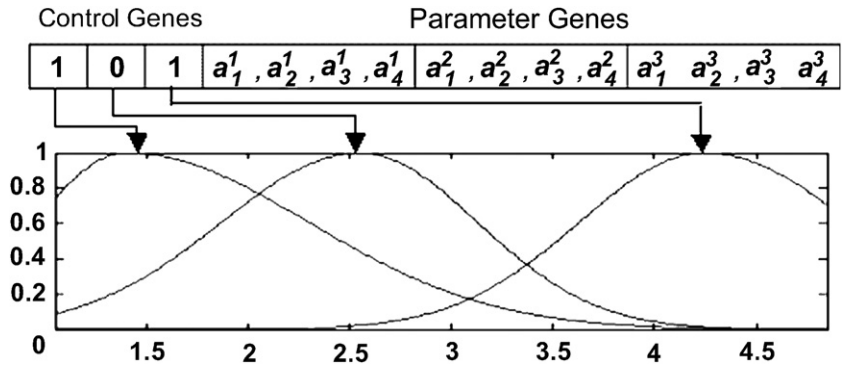

Fig. 4. Example of Gauss2mf encoded by control genes and parameter genes.

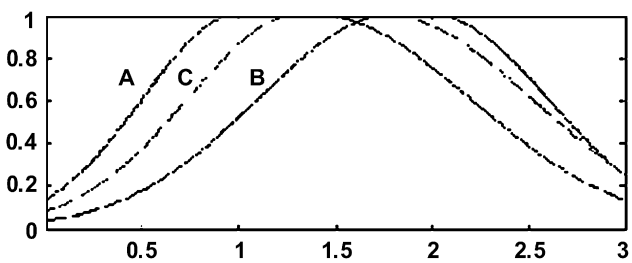

Fig. 5. Merging fuzzy sets A and B to form fuzzy set C.

establish a more compact fuzzy system with more appropriate distribution of fuzzy sets.

3.2.2.1. Merging similar fuzzy sets. Recall that the similarity measure defined for two fuzzy sets is given in Eq. (2). If $S(A, B)$ is greater than a given threshold, then the fuzzy sets A and B will be merged and become a new fuzzy set C. Suppose A and B have the membership functions $u_A(x; a_1, a_2, a_3, a_4)$ and $u_B(x; b_1, b_2, b_3, b_4)$, the resulting fuzzy set C with the membership function $u_C(x; c_1, c_2, c_3, c_4)$ is defined from merging A and B by

$$\begin{aligned} c_1 &= \min(a_1, b_1), \\ c_2 &= \lambda_2 a_2 + (1-\lambda_2) b_2, \\ c_3 &= \lambda_3 a_3 + (1-\lambda_3) b_3, \\ c_4 &= \max(a_4, b_4). \end{aligned} \tag{6}$$

The parameters $\lambda_2, \lambda_3 \in [0, 1]$ determine the relative influence of A and B on the generation of C. The threshold for merging similar fuzzy sets plays an important role in the improvement of interpretability. According to our experience, values in the range [0.4,0.7] can be a good choice. We set the threshold to 0.45 in this work. Fig. 5 illustrates a concrete example for merging A and B to form C.

3.2.2.2. Remove fuzzy sets similar to universal set or singleton set. Furthermore, if the similarity of a fuzzy set to the universal set $U(u_U(x) = 1, \forall x \in X)$ is larger than an upper threshold (θ_U) or smaller than a lower threshold (θ_S), then we can remove it from the rule base. The fuzzy set in the former case is very similar to the universal set, and in the latter case similar to a singleton set. Neither of these cases is desired to generate interpretable rule bases. We set θ_U to 0.8 and θ_S to 0.05 in this work. If a fuzzy set is removed, then the corresponding control gene will update its value from 1 to 0, and the rule antecedents associated with this fuzzy set will be removed from the corresponding rules.

3.2.3. Fuzzy rules generation strategy

Genetic optimizations of fuzzy rule base can be classified into three distinct approaches, which differ in how GA is applied in the learning process, including Michigan approach, Pittsburgh approach and iterative rule learning approach [26]. In Pittsburgh approach, each chromosome is encoded as an entire knowledge base, and combination of existing rules and generation of new rules can be easily done by genetic crossover and mutation operations. Therefore, in the current work the Pittsburgh approach is applied to extract rules from training data. Suppose there are N fuzzy variables, M_i^a is the number of active fuzzy sets for variable x_i. In addition, the "do not care" conditions are included for incomplete rules, hence the maximum number of possible fuzzy rules is $(M_1^a + 1) \times (M_2^a + 1) \times \cdots \times (M_N^a + 1)$ for N-dimensional problems. In order to search for a minimal number of fuzzy rules considering both accuracy and interpretability, FSAs perform the following tasks to achieve this goal.

3.2.3.1. Initialization of rule base population. In MOGFIDS, each fuzzy rule is encoded as a string of length N, where the ith element has a value c_i: $0 \leqslant c_i \leqslant M_i^a$ which indicates the c_ith fuzzy set is triggered $(c_i > 0)$ or the ith fuzzy variable does not play a role in rule generation $(c_i = 0)$. After that, FSA defines the population size N_{pop}, i.e., the number of fuzzy rule sets that represent a complete rule base. Each individual of fuzzy rule sets population is represented as a concatenated string of the length $N \times N_{rule}$, where N_{rule} is a predefined integer specifying the size of the initial fuzzy rule base. In this concatenated string, each substring of length N represents a single fuzzy rule. The heuristic procedure [27,28] is applied to generate the rule consequents for classification such that the consequents are not coded as parts of the concatenated string. The fuzzy rule sets are randomly initialized so that the value of the concatenated string can present one of the fuzzy sets of the corresponding fuzzy variable, or is equal to zero indicating "do not care" conditions.

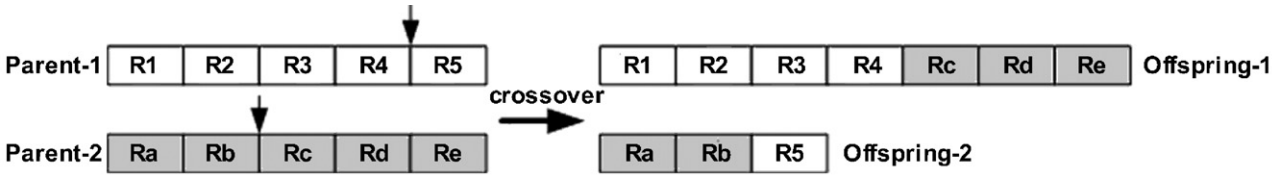

Fig. 6. Example of crossover operation on the rule sets.

3.2.3.2. Crossover and mutation. New offspring rule sets are generated by crossover and mutation. The one-point crossover operation is implemented due to its simplicity, which can randomly select different cutoff points for each parent to generate offspring rule sets. An example of crossover operation is given in Fig. 6. The mutation operation randomly replaces an element of the rule sets with another linguistic value if a simple probability test is satisfied. Elimination of existing rules and addition of new rules can also be used as mutation operations. As a result, the number of rules in the rule sets string can be changed accordingly. Note that since crossover and mutation operation may introduce the same redundant rules, FSA checks the rule sets and maintains single among all the rules in order to guarantee the consistency of fuzzy systems.

3.2.3.3. Evaluation criteria and selection mechanism. FSA applies three criteria to evaluate fuzzy rule set candidates: (i) Accuracy in terms of classification rate; (ii) number of fuzzy rules; and (iii) total length of fuzzy rules, i.e., the total number of rule antecedents in rule base. For each N-dimensional training sample $X^i = [x_1^i, x_2^i, \ldots, x_N^i]$, the fire-strength of rule R_i considering the inclusion relation is calculated in Eq. (5), and the sum of the fire-strength related to the rule R_i for class c is

$$\beta_{Class\ j}(R_i) = \sum_{X^k \in Class\ j} u_i(X^k), \quad j = 1, 2, \ldots, c. \tag{7}$$

The class C_j that has the maximum value of $\beta_{Class\ j}(R_i)$ can be found as the consequent of rule R_i by

$$\beta_{Class\ j_i}(R_i) = \max\{\beta_{Class\ 1}(R_i), \beta_{Class\ 2}(R_i), \ldots, \beta_{Class\ c}(R_i)\}. \tag{8}$$

If the maximum value cannot be uniquely found, i.e., there are some classes obtaining the same maximum value, then the fuzzy rule R_j should be removed from the rule base. After the rule base is constructed, the classification accuracy can be calculated using a single winner rule method [29]. For each training sample X^i, the winner rule R_i is determined as

$$\breve{u}_i(X^i) = \max\{\breve{u}_k(X^i) | k = 1, 2, \ldots, R\}, \tag{9}$$

where R is the number of rules. If the predicted class is not the actual class, or two or more fuzzy rules have the same maximum fire-strength, then the classification error increases one.

All the fuzzy rule-based candidates are evaluated by FSAs using NSGA-II algorithm. The preference for the multi-criteria can be defined in this approach for different trade-off requirements. Considering the basic requirement of IDS, it is essential to accurately classify large amount of normal traffic connections and intrusion attacks, accuracy is set the first priority and the other two objectives related to interpretability are set the same second priority. Suppose there are $N_{pop} + N_{offs}$ candidates, where N_{pop} is the size of parent population and N_{offs} is the size of offspring population resulting from crossover and mutation operations. FSAs employ elitism strategy to select N_{pop} best candidates from the mixed populations.

3.3. Agents interaction

The FSAs interact with one another for exchanging their fuzzy sets information and generating offspring agents. Assume the number of offspring agents N_{offs}^a is less than or equal to that of parent agents N_{curr}^a, N_{offs}^a FSAs can be randomly selected from the current agent population with the restriction that they should be different with one another. Therefore, N_{offs}^a offspring FSAs can be generated using crossover and mutation, in which two parent agents generate two offspring agents. As depicted in Fig. 2, the offspring agents also apply interpretability-based regulation strategy and fuzzy rules generation strategy to generate interpretable rule bases. After that, FSAs send their fitness information to AA, which applies NSGA-II algorithm to evaluate both parent and offspring FSAs and selects N_{curr}^a best agents to be the next generation population. The elitist FSAs considering both accuracy and interpretability can survive from the competition, while the low-fitness FSAs are discarded.

4. Experiments and evaluations

4.1. Data description, pre-processing and performance measurement

The KDD-Cup99 data set from UCI repository [30] is widely used as the benchmark data for IDS evaluation. In our experiments, we apply its 10% training data consisting of 494 021 connection records for training. Each connection record represents a sequence of packet transmission starting and ending at a time period, and can be classified as normal traffic, or one of 22 different classes of attacks. All attacks fall into four main categories:

- Denial-of-service (DOS)—Denial of the service that are accessed by legitimate users, e.g., SYN flooding.
- Remote-to-local (R2L)—Unauthorized access from a remote machine, e.g., password guessing.

Table 1
Feature set of preprocessed KDD-Cup99 data

Index	Feature name	Index	Feature name	Index	Feature name	Index	Feature name
1	duration	14	flag = RSTR	27	numRoot	40	sameSrvRate
2	protocolType = tcp	15	flag = OTH	28	numFileCreations	41	diffSrvRate
3	protocolType = udp	16	srcBytes	29	numShells	42	srvDiffHostRate
4	protocolType = icmp	17	dstBytes	30	numAccessFiles	43	dstHostCount
5	flag = SF	18	land	31	numOutboundCmds	44	dstHostSrvCount
6	flag = REJ	19	wrongFragment	32	isHostLogin	45	dstHostSameSrvRate
7	flag = S0	20	urgent	33	isGuestLogin	46	dstHostDiffSrvRate
8	flag = S1	21	hot	34	count	47	dstHostSameSrcPortRate
9	flag = S2	22	numFailed	35	srvCount	48	dstHostSrvDiffHostRate
10	flag = S3	23	loggedIn	36	serrorRate	49	dstHostSerrorRate
11	flag = SH	24	numCompromised	37	srvSerrorRate	50	dstHostSrvSerrorRate
12	flag = RSTO	25	rootShell	38	rerrorRate	51	dstHostRerrorRate
13	flag = RSTOS0	26	suAttempted	39	srvRerrorRate	52	dstHostSrvRerrorRate

- User-to-root (U2R)—Unauthorized access to gain local super-user (root) privileges, e.g., buffer overflow attack.
- Probing (Probe)—Surveillance and probing for information gathering, e.g., port scanning.

To prevent performance deterioration due to class imbalance problem in training, a random sub-sampling method is applied to the three largest classes: 'normal', 'Neptune' and 'Smurf', which have already contained 98% records of the whole training data set. The new training data contains 10^4 records of normal class and 10^3 records for each of the Neptune and Smurf classes, while the number of records of other classes remains intact. As a result, total 20 752 records are applied for training. To make the detection task more realistic, MOGFIDS is evaluated using KDD-Cup99 independent test data that contains 311 029 records with different class probability distribution and additional 14 unseen attack types. As each network connection record contains 34 continuous features and seven nominal features, the nominal features such as protocol (TCP/UDP/ICMP), service type (http/ftp/telnet/...) and TCP status flag (SF/REJ/...) are first converted into binary numeric features. Since the feature "service type" can be expanded into 71 binary features that can heavily increase the dimensionality as well as the initial rule length, this single feature is not applied in this work. Thus, totally 52 numeric features are constructed and normalized to the interval [0, 1] before processing by the proposed MOGFIDS. They are indexed and given in Table 1. Since the benchmark test data, similar to training data, has the class imbalance problem with skewed class distribution, accuracy alone is not sufficient for evaluation. Therefore, classification performance of MOGFIDS is measured by the precision, recall and F-measure that are commonly used to evaluate the rare class prediction. It is desirable to achieve a high recall without loss of precision. F-measure is a weighted harmonic mean that assesses the trade-off between them. They can be calculated using the confusion matrix in Table 2, and defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{10}$$

Table 2
Confusion matrix

		Predicted class	
		Positive class	Negative class
Actual Class	Positive class	True positive (TP)	False negative (FN)
	Negative class	False positive (FP)	True negative (TN)

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{11}$$

$$\text{F-measure} = \frac{(\beta^2 + 1)(\text{Precision} \cdot \text{Recall})}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad \text{where } \beta = 1, \tag{12}$$

$$\text{Overall accuracy} = \frac{TP + TN}{TP + TN + FN + FP}. \tag{13}$$

The receiver operating characteristics (ROC) analysis, which is originated from the signal theory and widely applied in medical data analysis, is adopted to depict the trade-off between FPR and true positive rate (TPR). Since the ROC analysis is insensitive to the changes in class distribution, even the proportion of positive (attack) to negative (normal) samples is changed in IDS testing, it will not change the ROC curves. This provides a promising index to evaluate the effectiveness of IDS classifiers.

4.2. Fuzzy systems analysis of MOGFIDS

4.2.1. Optimization results of MOGFIDS

We apply 12 FSAs each of which has 10 fuzzy rule sets solutions, therefore there are 120 fuzzy systems generated in total for initialization. The MOGFIDS is trained using 80 iterations for the evolution of FSAs. The trends of the multiple objectives against the number of iterations are plotted in Fig. 7(a–d), in which the objectives include the average overall-accuracy, average number of fuzzy sets, average number of fuzzy rules and average total-length of fuzzy rule-base among all the FSAs. The results demonstrate that the agents can continuously improve the average accuracy using the elitism strategy in each generation. In particular, the accuracy can be greatly increased

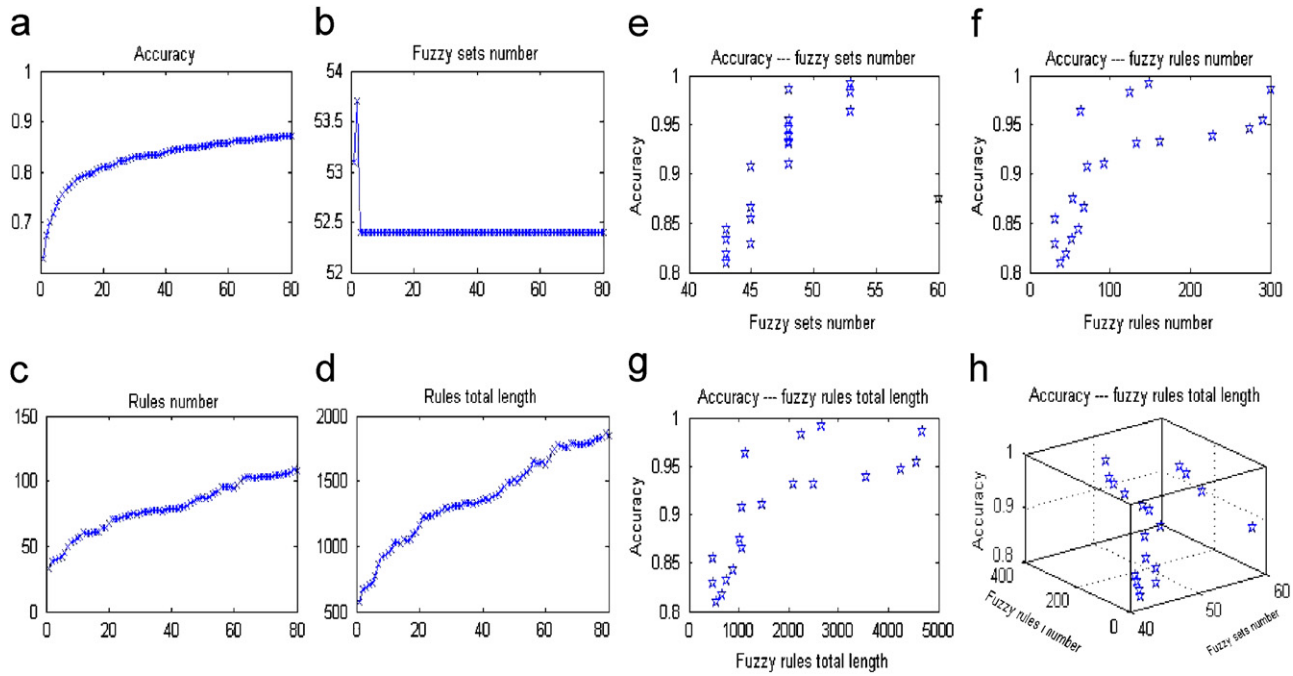

Fig. 7. MOGFIDS using 12 FSAs. (a–d) Trends of average overall-accuracy, fuzzy sets number, rules number and total length of rule base. (e–h) Non-dominated Pareto fronts about the fuzzy systems of MOGFIDS on KDD-Cup99 training data.

during the first 20 generations. As the first priority of optimization is given for accuracy in NSGA-II, it is found that the number of rules and their total rule length do not desirably decrease during the first 80 iterations, due to the complexity of rules extraction from intrusion detection data. However, on average stable accuracies can be obtained using about 106 rules each of which has the approximate rule length of 17 only, the rule base is acceptable for accurate classification.

The trade-offs among the multiple objectives within the non-dominated fuzzy system solutions are given in Fig. 7(e–h). The results show that, out of 120 fuzzy systems, there are 19 non-dominated solutions found from the training data. The average overall-accuracy obtained from the agents varies from 80% to 99%, with the number of rules ranging from 30 to 300. As widespread non-dominated solutions can be obtained in our approach considering both accuracy and interpretability, they can assist security experts to comprehend the intrusion attacks recognized by the fired rules, based on the interpretable knowledge provided therein. Regarding the accuracy as an important criterion for intrusion detection, the best resulted fuzzy system, which extracts 148 fuzzy rules from training data and obtains the peak accuracy (99.2417%) and lowest false positive rate (1.1%) for classifying normal network traffic, can be selected and applied for testing.

4.2.2. Comparison with the MOGFIDS without agent-based evolutionary computation framework

To demonstrate the effectiveness of agent-based evolutionary computation framework in MOGFIDS, two comparative experiments are carried out to evaluate multi-agent optimization approach and FSA intra-behavioral optimization strategies. First, the MOGFIDS is trained with only one FSA for baseline comparison. The FSA contains 10 fuzzy rule set solutions such that it has exactly 10 fuzzy systems used for learning. The results given in Fig. 8(a–d) show that the average classification accuracy is lower than that of MOGFIDS using 12 FSAs, and the average rates of increase on both number of rules and their total length are higher than those of 12 agents based MOGFIDS. As shown in Fig. 8(e–h), the non-dominated and the most accurate fuzzy system can only obtain about 88% accuracy using an unacceptably large amount of fuzzy rules for classification. These results demonstrate that the agent cooperation and competition can effectively exchange fuzzy set information for constructing accurate and compact fuzzy systems.

Second, MOGFIDS is trained without using fuzzy sets distribution strategy and interpretability-based regulation strategy for baseline comparison. Since our proposed fuzzy sets distribution strategy is not applied in this comparative experiment, fuzzy sets are encoded in simple chromosome formulation of simple GA instead of HGA. We apply 12 FSAs each of which contains 10 fuzzy rule set solutions such that there are 120 fuzzy systems used for learning. To demonstrate the importance of compactness in ensuring interpretability of GFRBS, the above-mentioned strategies that can optimize the distribution and compactness of fuzzy sets are not applied in this experiment. Fuzzy sets distribution strategy is not used in intra-behavior of all FSAs, and interpretability-based regulation strategy is not used in both the intra-behavior and inter-behavior of all FSAs. The comparative results show that, although number of fuzzy sets obtained by the proposed MOGFIDS in Fig. 7(b) is not significantly smaller than that obtained by this experiment in Fig. 9(b), there is a large difference between their numbers of fuzzy rules and total rule length as shown in Figs. 7(c–d) and 9(c–d). It indicates that the fuzzy sets ob-

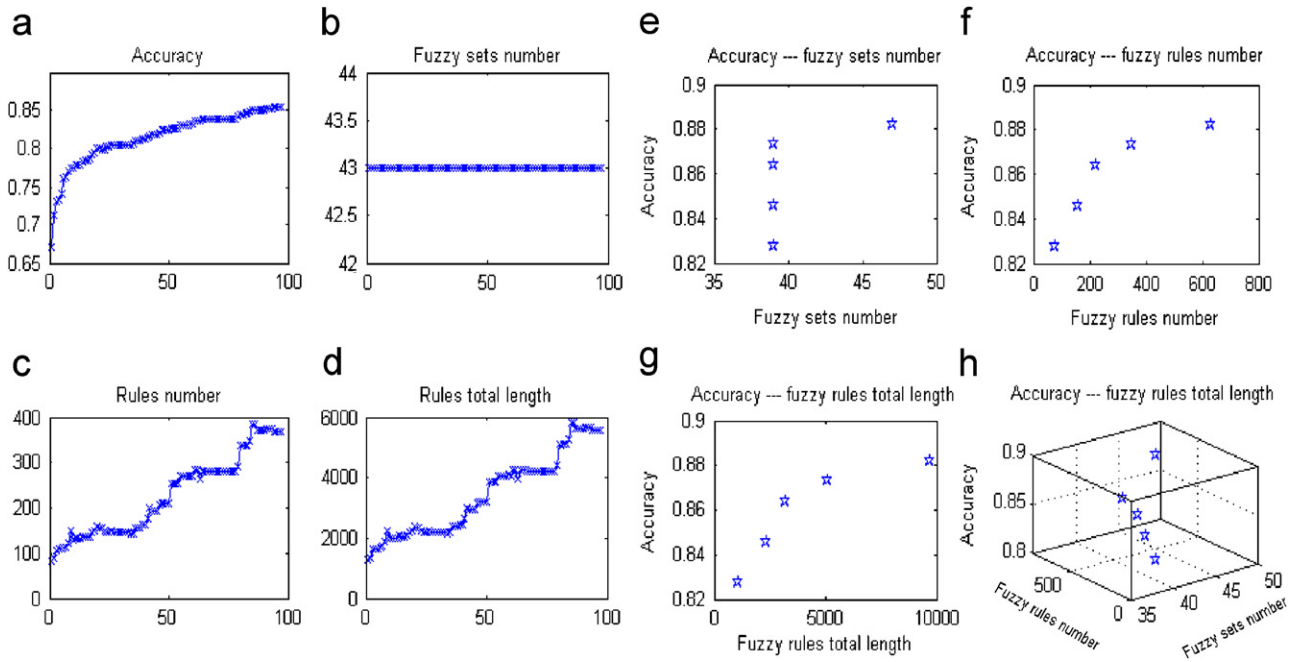

Fig. 8. MOGFIDS using one FSA. (a–d) Trends of the average performance. (e–h) Non-dominated Pareto fronts.

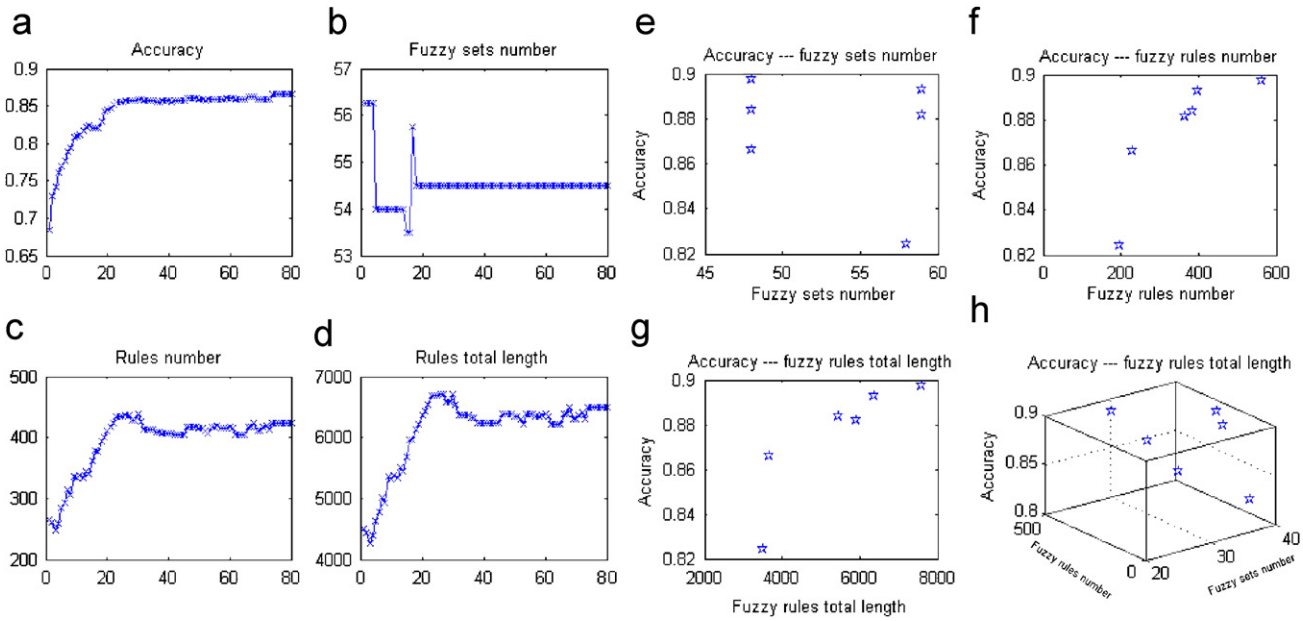

Fig. 9. MOGFIDS using 12 FSAs, without applying fuzzy sets distribution strategy and interpretability-based regulation strategy. (a–d) Trends of the average performance. (e–h) Non-dominated Pareto fronts.

tained by the proposed MOGFIDS are more distinguishable, hence the fuzzy systems are more compact, and the number of fuzzy rules and total rule length can be desirably small.

The comparative results in Figs. 7(e–h) and 9(e–h) further show that the non-dominated solutions of the proposed MOGFIDS can obtain higher accuracy using smaller number of fuzzy rules and total rule length, as compared to the results obtained by the non-dominated solutions in this experiment. As discussed in Section 2, it is desired to achieve a compact fuzzy system that has a small number of fuzzy sets, fuzzy rules and conditions in the rule premise. Compact fuzzy rules are easier to be interpreted if they can be defined by the most relevant fuzzy variables and appropriate fuzzy sets. Compact fuzzy system is easier to be understood if the number of fuzzy rules can be small using high compactness of fuzzy rule base. The comparative results indicate that MOGFIDS is able to obtain a compact

```
set all weights W[A] = 0.0;
for i = 1 to m do
   begin
      randomly select a sample R_i;
      find its k nearest hits H_j;
      for each class C≠class(Ri) do
        find its k nearest misses M_j(C);
      for A = 1 to N do
            W[A] = W[A] − Σ_{j=1}^{k} diff(A,R_i,H_j)/(m·k)  + Σ_{C≠(Rclass)} [P(C)/(1 − P(class(R))) Σ_{j=1}^{k} diff(A,R_i,M_j(C))]/(m·k);
   end
```

Fig. 10. Relief-F algorithm.

fuzzy system, which is interpretable for human user to analyze and understand the high-level knowledge of the classification results.

4.3. Experiments and evaluations on filter-based approaches

4.3.1. Brief descriptions of filter-based approaches

The feature selection experiments are first conducted using filter-based approaches on the training data. The objective of this experiment is to find out which filter can achieve the best performance for intrusion detection data, and to suggest a good feature subset that contains relevant features with relative order of importance for baseline reference. Various filtering criteria and a well-known feature selection algorithm are applied in this experiment to measure the relevance of features from training data. Hence, the features can be ranked according to their relevance values, in order to determine their orders of importance. These filtering criteria are briefly discussed as follows:

- Information gain (IG), which is also known as mutual information, measures the expected reduction in entropy of class before and after observing features. Larger difference indicates that the selected feature is more important to contain the class discriminatory information. IG is measured as

$$InfoGain(S, F) = Entropy(S) - \sum_{v \in V(F)} \frac{|S_v|}{|S|} \cdot Entropy(S_v), \tag{14}$$

  where S is the pattern set, $|S|$ is the number of samples in S, v is value of feature F, and S_v is the subset of S where feature F has value v. The entropy of class before observing features is defined as

$$Entropy(S) = \sum_{c \in C} -\frac{|S_c|}{|S|} \cdot \log_2 \frac{|S_c|}{|S|}, \tag{15}$$

  where C is the class set and S_c is the subset of S belonging to class c. IG is the fastest and simplest ranking method, however, the drawback is that it flavors the features with many number of values.
- Gain ratio (GR) normalizes the IG by dividing it by the entropy of S with respect to feature F, in order to discourage the selection of features with many uniformly distributed values. GR is measured as

$$GainRatio(S, F) = InfoGain(S, F)/SplitInfo(S, F), \tag{16}$$

$$SplitInfo(S, F) = \sum_{i=1}^{n} -\frac{|S_i|}{|S|} \cdot \log_2 \frac{|S_i|}{|S|}, \tag{17}$$

  where S_i is the subset of S where feature F has its ith possible value, and n is the number of subclasses split by feature F. The drawback of this method is that if many S_i have a particular value for a feature F, then the *SplitInfo* value will be very small, and hence *GainRatio* value will be undesirably large.
- Chi-square (CS) measures the well-known χ^2 statistics of each individual feature with respect to the classes. The features are ranked by the descending order of their χ^2 values, in which large χ^2 values obtained by the features reveal their strong correlation with the classes. The χ^2 of a feature F is measured as

$$\chi^2(F) = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}, \tag{18}$$

$$E_{ij} = \frac{R_i \cdot C_j}{|S|}, \tag{19}$$

  where m is the number of intervals discretized from the numerical values of F, k is the number of classes, A_{ij} is the number of samples in the ith interval with jth class, and E_{ij} is the expected occurrence of A_{ij} in which R_i is the number of samples in the ith interval and C_j is the number of samples in the jth class.

  These information-theoretical and statistical criteria have been empirically proved to be effective to select relevant features from some high-dimensional real-world domains, however, the major limitation of their applications is that they assume the independence property of features. As the interaction of relevant features is not taken into account, filtering on highly correlated features might degrade performance of classifiers.
- Relief-F is an instance-based feature-ranking algorithm that deals with noisy data and multi-class problem. The algorithm flavors the features that differentiate samples from different classes and have same values for samples from the same class. The high-level description of algorithm is depicted in Fig. 10. Besides that Relief-F is computationally efficient,

it can also identify irrelevant features and take into account the interaction of relevant features. However, it cannot identify the redundant features and determine a useful subset from many weakly relevant features, since distance metric combines measurements for all features in nearest neighbor algorithms, and weakly relevant features bias the similarity measure away from strongly relevant features.

4.3.2. Experimental results and analysis

The features listed in Table 1 are ranked using the filtering criteria described in the previous subsection, and the ranking results are given in Table 3. To evaluate the effectiveness of these filtering criteria, four well-known classification algorithms are employed, including (i) C4.5 decision tree; (ii) Naïve Bayes (NB); (iii) *k*-nearest neighbor (*k*-NN) and (iv) support vector machine (SVM). Note that the C4.5 is run with tree pruning, attempting to alleviate the over-fitting problem. In *k*-NN the parameter *k* is set to 5, and the SVM is trained with a fast sequential minimal optimization method [31] and a polynomial kernel of order 3. The classification accuracies obtained with these various classifiers using the ranked features are compared in Fig. 11. The results show that C4.5, *k*-NN and SVM classifiers can obtain high accuracy using the top 15–30 features ranked by IG, CS and Relief-F filtering criteria. In particular, IG outperforms all the others in terms of both the rate of accuracy improvement and minimum number of ranked features that contributes the peak accuracy for different classifiers. The accuracies obtained with classifiers using GR ranking often start with the lowest value, grow slowly, and do not catch up with that using other ranking filters until a relative large number of features are used. Therefore, GR is not effective to filter the features of network packets due to the possible drawback as discussed in the previous subsection. All ranking filters, except GR, can lead to the continuous growth of accuracy as the number of ranked features increases. It can be found that the increasing trend of accuracy is retarded and a steady state is reached when the number of features exceeds 30, implying that the last 22 individual features are not significantly relevant for classification. In particular, the last nine features ranked by IG, GR and CS are identical (in which over half of them are also the last ranked features in Relief-F ranking) and cannot improve classification accuracy. Hence, they are considered as the most irrelevant features.

As different classifiers consistently achieve high accuracy using the top 30 relevant features ranked by IG, CS and Relief-F criteria, a simple positional-scoring method—Borda preference rule [32] is applied to assign weights to these features, in order to find a collective feature subset that can be voted by these filters. It is interesting to note that the features of this collective subset, which are the commonly top features ranked by different filters as given in Table 3, are identical to the top 30 features in IG ranking. These results suggest that IG criterion is admissibly robust to filter network traffic features. While Relief-F is outperformed by IG, it reflects that many features of network packets tend to be weakly relevant for classification, and hence Relief-F cannot accurately assign relevant weights to them. In addition, as compared with both C4.5 and *k*-NN classifiers that

Table 3
Feature ranking by different filtering criteria and a collective feature subset voted by Borda rule

Filters	Feature rankings (most important feature ↔ least important feature)
Info. Gain	**16**,**34**,**46**,**35**,**44**,**45**,**51**,**47**,**17**,**52**,**41**,**40**,**48**,**38**,**43**,**49**,**23**,**2**,**36**,**5**,**4**,**19**,**21**,**39**,**42**,**50**,**24**,**3**,**37**,7,14,6,1,33,11,22,12,25,28,18,27,30,29,8,26,20,32,31,10,9,15,13
Gain Ratio	**18**,**19**,**22**,**24**,**4**,**23**,**25**,**5**,**2**,**21**,**3**,**16**,**28**,**12**,**11**,**14**,**17**,**29**,**7**,**38**,**41**,**52**,**30**,**40**,**27**,**36**,**37**,**39**,**50**,**48**,34,42,33,45,51,46,35,49,47,6,44,43,1,8,26,20,32,31,10,9,15,13
Chi square	**16**,**17**,**34**,**35**,**46**,**52**,**50**,**44**,**51**,**45**,**48**,**21**,**40**,**24**,**19**,**49**,**41**,**38**,**36**,**47**,**39**,**42**,**43**,**1**,**37**,**18**,**22**,**4**,**23**,**5**,2,3,27,29,14,12,7,25,6,30,28,11,33,8,26,20,32,31,10,9,15,13
Relief-F	**2**,**23**,**47**,**45**,**4**,**5**,**44**,**43**,**34**,**3**,**40**,**46**,**19**,**51**,**35**,**49**,**36**,**38**,**50**,**41**,**37**,**7**,**39**,**52**,**6**,**14**,**48**,**42**,**15**,**21**,33,11,12,1,24,18,25,9,17,22,10,13,30,29,28,27,16,20,26,32,31,8
Borda Rule	34,35,44, 45, 46,51,47,16,40,52,17,43,23, 41, 49,2, 19, 38,48,50,4,36,5,21,39,3,24,37, 42,7

The top 30 ranked features are bold-faced, and the underlined features have the same scores assigned by Borda rule.

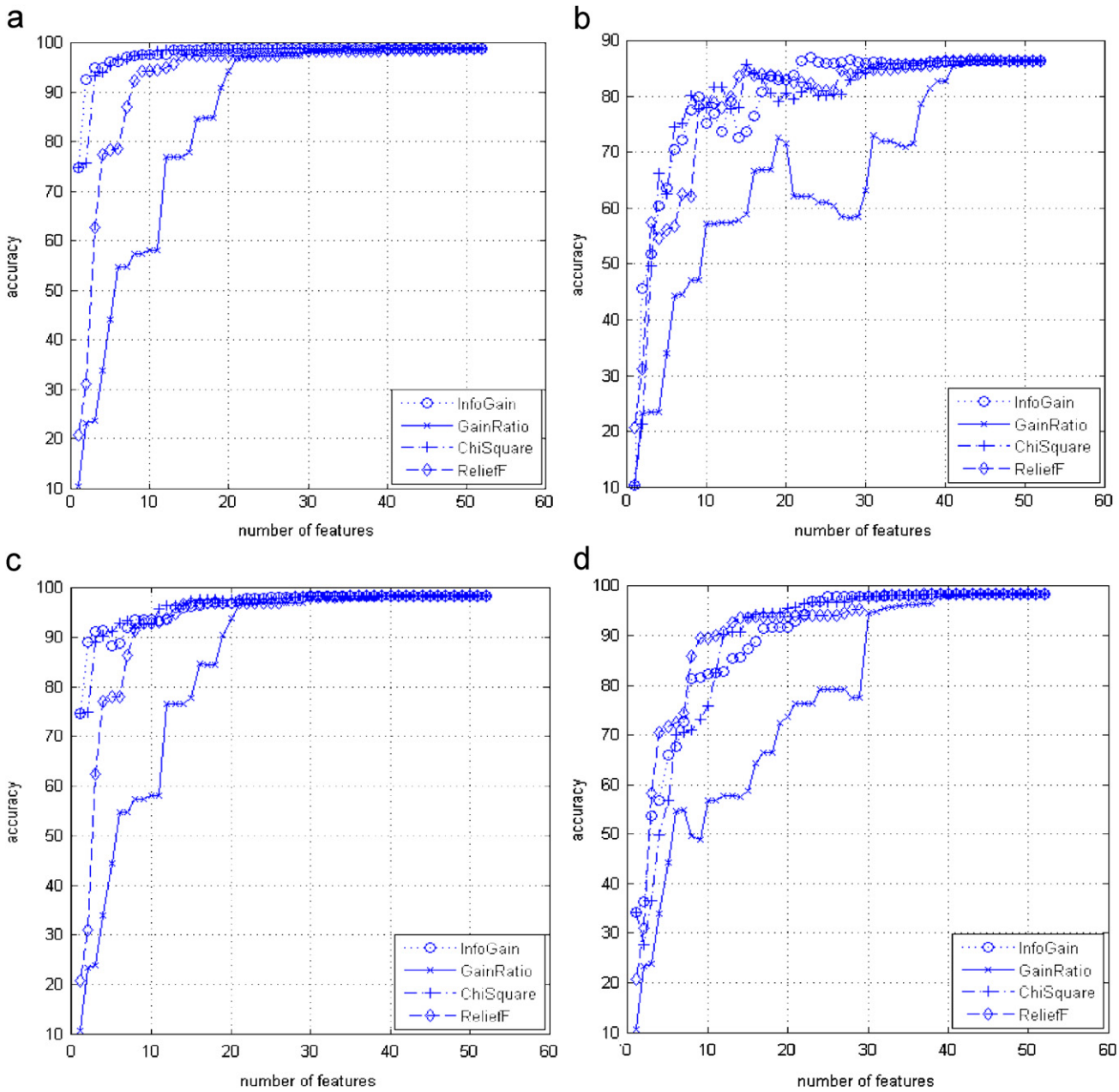

Fig. 11. Classification accuracy obtained with (a) C4.5, (b) Naïve Bayes, (c) k-NN, where $k=5$ and (d) SVM using different number of features as ranked by different filtering criteria based on 10-fold cross-validation on KDD-Cup99 training data. The number of features is increasing with the inclusion of gradually less relevant features, according to the rankings in Table 3.

achieve high accuracies, the accuracies obtained with NB using different filtering criteria are relative low ($<88\%$), and their trends fluctuate when the number of features increases. As NB strongly assumes the features are statistically independent, it can be robust with respect to irrelevant features but susceptible to correlated features. Therefore, it indicates that some packet features are highly correlated for classification (indeed for a particular class prediction as discussed in next subsection) such that NB cannot yield significant improvement on accuracy using minimum number of features ranked by filters.

4.4. Experiments and evaluations on wrapper-based approaches and MOGFIDS

4.4.1. Brief descriptions of wrapper-based approaches and feature selection in MOGFIDS

Various wrappers using different classification algorithms and searching strategies are employed in this experiment for baseline reference. The objective of this experiment is to evaluate the effectiveness of MOGFIDS by comparing it with the baseline wrappers and well-known classifiers in terms of the feature selection performance and the classification results, re-

Table 4
Parameter settings of GA applied in the baseline wrappers

Parameter	Value
Population size	80
One-point crossover probability	0.8
Mutation probability	0.01
Selection scheme	Tournament selection of size 2
Number of generation	100

spectively. In our proposed MOGFIDS, the fuzzy variables are selected and removed through the crossover and mutation operations on the control genes during the FSAs evolution, such that the MOGFIDS can heuristically search a desired feature subset that minimizes the classification error and improves the interpretability of fuzzy system accordingly. In the wrapper-based approaches, the C4.5, NB, *k*-NN and SVM are employed as the classification algorithms for comparison. As the optimal feature subsets can only be guaranteed by applying exhaustive or branch-and-bound search, four heuristic search strategies are adopted in the wrappers to find near-optimal feature subsets from training data. They are briefly discussed as follows:

- Best first (BF) search starts with empty feature set and explores all possible feature subsets by adding features one by one. The feature subset with highest accuracy is then further expanded until no improvement is found. After that, BF backtracks the second best subset and expands it iteratively. If no improvement is found in the limited *k* expansion, the search will be terminated and the best subset is returned.
- Forward sequential selection (FSS) starts with empty feature subset and adds features one by one in the growing set. The best subset with highest accuracy is considered as the base subset in next iteration. The search will be terminated after the accuracy of current subset cannot be increased with adding feature.
- Backward sequential selection (BSS), on the contrary, starts with full feature subset and removes features one by one in the shrinking set. It iteratively removes feature that yields the maximal performance increase.
  The FSS and BSS tend to become trapped on the local maxima since they cannot modify the previous subsets for re-evaluation. Hence, many evolutionary algorithms have been proposed to search global optimal feature subsets.
- GA is first proposed in Ref. [33] for feature selection. Each chromosome represents a feature subset candidate, where each feature is encoded as a gene with binary value. GA uses the classification accuracy as the fitness value and selects the fittest chromosomes to survive in the next generation. Feature subsets are explored and exploited by applying genetic operators, such as crossover and mutation operators, probabilistically on the chromosomes. In this work, the parameters of GA applied in the wrappers are given in Table 4.

4.4.2. Experimental results and analysis

Table 5 presents the feature subsets found by different wrappers using various search strategies, their overall classification accuracy (ACC), and the FPR on classifying normal traffic from training data. The detailed classification performances are shown in Table 6 for different wrappers using GA search strategy and MOGFIDS on both the training and test data. Table 7 summarizes the available results obtained with other recently proposed classifiers in the IDS literature. The confusion matrices obtained with MOGFIDS in training and testing are given in Table 8. Following are the detailed analysis about the extensive experimental results.

4.4.2.1. Feature selection results of baseline wrappers and MOGFIDS. Regarding the results obtained with classifiers C4.5, *k*-NN and SVM in Table 5, BSS and GA outperform BF and FSS, in terms of both ACC and FPR measures. Among all the searching strategies, even though GA search may not improve ACC significantly, it is very effective to remove the irrelevant features for different wrappers due to the following two reasons. First, the size of optimal subsets searched by GA is smaller than that of the entire feature sets while ACC is improved simultaneously. Second, the FPR on classifying normal network packets can be obviously reduced by GA. These can substantially reduce large amount of false intrusion alarms and improve the real-time performance of IDS, in particular for the high-speed network with bulk of daily normal packets nowadays.

Among all the baseline wrappers, the application of C4.5 and GA produces a compact subset that results in the lowest classification error (0.6%) and FPR (1.4%) on KDD-Cup training data as shown in Table 5. However, the result in Table 6 shows that C4.5 with tree pruning encounters learning difficulty on the attack classes with small number of training samples, which also coincides with its common issue of over fitting, causing the Recall and F-measure on classifying DOS and R2L attacks in test samples are relatively lower than those of other baseline wrappers and MOGFIDS. According to Table 5, regardless of the search strategies, NB surprisingly obtains very high FPRs among all the wrappers, demonstrating that it fails to retain features that could be highly correlated for classifying normal network traffic. Both the *k*-NN and SVM wrappers give satisfactory but insignificantly improved ACC using different search strategies. The size of their feature subsets and the FPR are, respectively, larger than and higher than those of C4.5 and MOGFIDS. The MOGFIDS encouragingly obtains the second highest ACC (99.24%) as well as the lowest FPR (1.1%) among all the wrappers during training.

In addition, it is shown that only 27 out of 52 features are searched by MOGFIDS, indicating that the other 25 removed features are not significantly relevant for classification using fuzzy rules. The smallest size of subset (20 features) is found by C4.5 using GA, instead of MOGFIDS. It is due to the fact that MOGFIDS optimizes both the accuracy and interpretability simultaneously, causing feature selection bias towards the inclusion of both strongly and weakly relevant features. Comparing with the relevant and irrelevant features identified by the filters in Section 4.3, MOGFIDS includes 20 out of 30 relevant features in Borda ranking, and excludes 8 out of 9 irrelevant features ranked by IG, GR and CS. In Ref. [34], a *k*-NN/GA

Table 5
The feature subset selected by different wrappers using different classifiers and search strategies, overall accuracy (ACC), and false positive rate (FPR) on classifying normal traffic obtained based on 10-fold cross-validation on training data

Classifier	Search	Selected feature subset (number of features)	ACC (%)	FPR (%)
C4.5	BF	2,3,16,17,18,21,23,30,33,34,41,45,46,47,48,51 (16)	98.9433	2.3
	FSS	2,3,16,17,18,21,23,30,33,34,41,45,46,47,48,51 (16)	98.9433	2.3
	BSS	1,2,3,5,6,11,12,14,16,17,18,19,20,23,25,28,29,33,36,40,41,43,45,48,49,52 (26)	99.1712	1.8
	GA	1,3,4,6,9,11,14,15,16,17,18,23,25,28,33,36,40,41,45,48 (20)	99.3992	1.4
	–	Entire feature set (52)	98.9123	3.1
NB	BF	2,4,5,6,11,12,14,17,18,19,21,23,24,25,27,29,34,44,46,47,48 (21)	94.4266	20.3
	FSS	2,4,5,6,11,12,14,17,18,19,21,23,24,25,27,29,34,44,46,47,48 (21)	94.4266	20.3
	BSS	2,3,4,5,7,12,14,17,18,19,20,21,22,23,24,25,27,28,29,30,40,44,46,47,48,50 (26)	94.3126	17.2
	GA	3,4,5,6,8,11,13,14,17,18,19,20,21,22,23,25,27,28,29,30,31,33,34,44,46,47,48,50 (28)	96.1566	19.3
	–	Entire feature set (52)	86.4809	37.4
k-NN ($k = 5$)	BF	2,3,4,7,14,15,16,17,18,19,20,22,23,24,25,28,29,30,33,38,40,41,48,50,51 (25)	98.3736	5.8
	FSS	2,4,7,14,15,16,18,19,22,23,24,28,29,33,38,40,41,48,50,51 (20)	98.2700	6.0
	BSS	1,2,3,5,6,7,8,10,11,12,14,16,18,19,21,22,23,24,25,27,28,29,30,33,34,38,39,40,42,43,45,46,47,48,49,50,52 (37)	98.6118	3.9
	GA	2,3,4,5,7,14,16,17,18,19,21,22,23,24,25,27,28,29,30,33,34,38,39,40,41,42,43,45,46,47,48,49,50,51,52 (35)	98.9123	3.7
	–	Entire feature set (52)	98.4461	5.0
SVM	BF	1,2,3,4,5,16,17,19,21,22,23,24,25,28,29,30,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52 (36)	98.3218	3.9
	FSS	1,2,3,4,5,6,16,17,19,21,22,23,24,25,28,30,33,34,35,36,38,40,41,42,43,44,45,46,47,48,50,51,52 (31)	98.2803	4.0
	BSS	1,2,3,4,5,16,17,18,19,21,22,23,24,25,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52 (39)	98.3736	3.9
	GA	1,2,3,4,5,6,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52 (43)	98.7051	3.5
	–	Entire feature set (52)	98.4046	3.6
MOGFIDS	NSGA-II	2,3,4,16,17,18,19,20,22,23,24,25,28,29,33,34,36,41,43,44,45,46,47,48,49,50,51 (27)	99.2434	1.1

The result obtained with each classifier with no feature selection is shown as well.

Table 6
Recall, precision, F-measure, overall accuracy and classification cost using cost matrix [35] obtained with different wrappers/GA and MOGFIDS on classifying training and test data

	Metric	C4.5 (20-feature)		NB (28-feature)		k-NN ($k=5$) (35-feature)		SVM (43-feature)		MOGFIDS (25-feature)	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Probe	Recall	99.6286	88.4782	96.3169	78.0125	99.4738	81.9251	99.0715	86.6779	99.6104	**88.5982**
{4107:	Precision	99.5978	73.8234	96.9773	49.8161	99.5046	61.8970	99.1943	**77.3565**	99.1277	74.4003
4166}	F-measure	99.6132	80.4892	96.6460	60.8045	99.4892	70.5165	99.1329	**81.7524**	99.3685	80.8809
DOS	Recall	99.8124	97.0799	99.5544	82.4488	99.7655	97.3413	99.6717	**97.4679**	99.8537	97.2017
{5467:	Precision	100	**99.9319**	98.5376	97.9967	99.5088	99.6646	99.8590	99.8707	99.3810	99.8963
229853}	F-measure	99.9061	98.4853	99.0434	89.5529	99.6370	98.4893	99.7653	**98.6547**	99.6168	98.5306
U2R	Recall	80.7692	**16.2281**	67.3077	13.1579	48.0769	14.0351	53.8462	10.0877	78.8462	15.7895
{52:	Precision	80.7692	2.6185	42.6829	2.7855	80.6452	3.5281	96.5517	54.7619	93.1818	**61.0169**
228}	F-measure	80.7692	4.5094	52.2388	4.5977	60.2410	5.6387	69.1358	17.0370	85.4167	**25.0871**
R2L	Recall	98.7342	3.3788	97.9204	5.4173	98.7342	5.0590	98.0108	3.3603	99.2007	**11.0137**
{1126:	Precision	99.0027	28.2980	94.8336	38.6344	97.2395	55.4127	97.3070	46.2192	94.6610	**68.3928**
16189}	F-measure	98.8683	6.0368	96.3523	9.5022	97.9811	9.2715	97.6576	6.2651	96.8777	**18.9722**
Normal	Recall	98.6000	98.1318	80.7000	89.9972	96.3000	95.7668	96.5000	98.0080	98.8700	**98.3645**
{10000:	Precision	97.6238	**74.8907**	88.4868	50.5989	96.8813	73.7941	93.9630	73.4802	99.7881	74.7370
60593}	F-measure	98.1095	**84.9503**	84.4142	64.7779	96.5898	83.3568	95.2146	83.9900	99.3269	84.9382
Overall accuracy		99.3992	92.2332	96.1566	79.7996	98.9123	91.9638	98.7051	92.4663	99.2434	**92.7672**
Classification cost		0.2426		0.4853		0.2459		0.2474		**0.2317**	

The wrappers are trained with 10-fold cross-validation. The numbers of records of each class category in training and test data are given under the category name in the format of {training:test}, the best results under testing are bold-faced, and the second best results are underlined, with respect to each performance metric for each class category.

Table 7
Recall, precision, F-measure, overall accuracy and classification cost obtained with the recently proposed classifiers in the literature

	Metric	KDD-Cup 99 Winner [35] (all-feature)		Bayesian network [36] (17-feature)		CART [36] (12-feature)		RIPPER [13] (all-feature)		Improved PNrule [13] (all-feature)		EFRID [37] (all-feature)		Multi-classifier [38] (all-feature)	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Probe	Recall	—	83.30	98.57	—	97.71	—	100	81.16	99.61	**89.01**	50.35	—	—	88.70
	Precision	—	64.81	—	—	—	—	99.37	77.92	99.56	**82.11**	—	—	—	—
	F-measure	—	72.90	—	—	—	—	99.68	79.51	99.58	**85.42**	—	—	—	—
DOS	Recall	—	97.10	98.16	—	85.34	—	100	22.06	100	21.74	98.91	—	—	**97.30**
	Precision	—	**99.88**	—	—	—	—	99.83	95.75	99.88	96.68	—	—	—	—
	F-measure	—	**98.47**	—	—	—	—	99.91	35.86	99.94	35.50	—	—	—	—
U2R	Recall	—	13.20	60.00	—	64.00	—	98.08	11.84	90.38	11.40	88.13	—	—	**29.80**
	Precision	—	**71.43**	—	—	—	—	89.47	55.10	94.00	53.06	—	—	—	—
	F-measure	—	**22.28**	—	—	—	—	93.58	19.49	92.15	18.77	—	—	—	—
R2L	Recall	—	8.40	98.93	—	95.56	—	99.47	8.33	99.20	**13.05**	7.41	—	—	9.60
	Precision	—	**98.84**	—	—	—	—	99.56	81.85	97.47	82.37	—	—	—	—
	F-measure	—	15.48	—	—	—	—	99.51	15.12	98.33	**22.53**	—	—	—	—
Normal	Recall	—	**99.50**	96.64	—	100	—	—	—	—	—	92.78	—	—	—
	Precision	—	**74.61**	—	—	—	—	—	—	—	—	—	—	—	—
	F-measure	—	**85.28**	—	—	—	—	—	—	—	—	—	—	—	—
Overall accuracy		–	92.71	—	—	—	—	—	—	—	—	—	—	—	—
Classification cost		0.2331		—		—		—		—		—		**0.2285**	

Note that they are provided for reference instead of direct comparison, as their sampling on training and test data are different from one another. For example, [36] applies KDD-Cup99 training data set for both training and test such that the results are cited in the column "Train". The best results under testing are bold-faced, the second best results are underlined, and the hyphen marks indicate that results are not reported in the papers.

Table 8
Confusion matrix obtained with MOGFIDS in (left) training and (right) testing

	Probe	DOS	U2R	R2L	Normal
Training					
Probe	4091	4	0	4	8
DOS	4	5459	0	1	3
U2R	2	0	41	2	7
R2L	3	1	2	1117	3
Normal	27	29	1	56	9887
Testing					
Probe	3691	127	4	83	261
DOS	349	223 421	0	276	5807
U2R	134	0	36	12	46
R2L	361	3	9	1783	14 033
Normal	426	102	10	453	59 602

wrapper approach is proposed to search the top five relevant features for each attack category (total 14 features), in which 11 out of the 14 ranked features are also overlapped in the feature subset selected by MOGFIDS. Due to the space limit, among 27 feature variables selected by MOGFIDS the fuzzy distributions of only two of them with index numbers 46 and 51 are shown in Fig. 12. These two features represent the percentage of connections made to different services provided in different hosts, and the percentage of connections made to different hosts that have "REJ" errors within 100 connection windows, respectively. It demonstrates that our MOGFIDS can generate distinguishable fuzzy sets distributions easily understandable by human beings. The distributions of other features are also consistent with those shown in Fig. 12.

4.4.2.2. Classification results of baseline wrappers and MOGFIDS on different class categories. Considering the rare classes such as U2R and R2L attacks, their probability distributions of training data are different from those of test data. In particular for R2L attacks, there is a large ratio (1:14) of its sample size on training data to test data. Table 6 shows that MOGFIDS outperforms all the baseline wrappers in terms of Recall, Precision and F-measure for R2L attacks, and also achieves the highest Precision and F-measure among all the wrappers for U2R attacks, indicating that MOGFIDS can relatively alleviate the over-fitting problem when it is learned with small training samples of rare classes and evaluated with large test samples containing novel attacks.

Regarding the major classes such as Probe and DOS attacks, Table 6 shows that both SVM and MOGFIDS can achieve high F-measure rates in testing, as compared with those of C4.5, NB and k-NN wrappers. On the recognition of normal network traffic, C4.5 and MOGFIDS outperform the wrappers and obtain very good prediction results. Among all the wrappers NB does not perform well in testing and gives unacceptably high FPR (10.00%) for normal class. As discussed previously, it is due to the fact that there is a high correlation of features used for recognizing the normal network traffic. It is found that k-NN and SVM do not give significantly good results for the normal class. As MOGFIDS achieves high Recall (98.36%) and Precision (74.74%) on classifying normal network traffic, it demonstrates that MOGFIDS only generates small amount of false intrusion alarms, and most unseen and novel intrusion attacks are not classified as normal network traffic. In addition, Table 6 shows that among all the wrappers MOGFIDS achieves the highest overall-accuracy (92.77%) and the lowest classification cost (0.2317) based on the cost matrix. The detailed classification results on different class categories can be found in Table 8, which shows that most of the normal network connections can be correctly classified as normal by MOGFIDS in testing.

4.4.2.3. Comparison with other approaches. The MOGFIDS is further compared with the winner of KDD-Cup99 and six well-known classifiers reported in the IDS literature. According to Table 7, the KDD-Cup99 winner [35] gives the lowest FPR on classifying normal network connections, however, it is outperformed by our MOGFDIS in all the attack classes prediction, in terms of Recall, F-measure, overall-accuracy and classification cost. Since the six classifiers do not apply the exactly same data set as used by KDD-Cup99 participants and MOGFIDS for training and testing purposes, their sampling on training and test data are different from one another such that the comparison is provided for reference only. From Table 7, it is found that the Bayesian Network [36], CART [36], RIPPER [13] and EFRID [37] classifiers do not give significantly good results. Although the classification performances of both RIPPER and PNrule [13] are found to be better than MOGFIDS in training, their test performances indicate that they suffer from the over-fitting and generalization problem during learning. The performance of PNrule is better than MOGFIDS in Probe and R2L attack classes in testing, however its classification results for DOS attack is found disappointing among all the classifiers. The multi-classifier model [38] combines multi-layer perception neural network, Gaussian classifier and k-means clustering algorithms to maximize the classification accuracy for each attack category individually such that it can obtain high Recall rates on Probe, DOS and U2R attack classes. However, it does not take into account the minimization of FPR on classifying normal network traffic and seriously lacks interpretability for security analysis.

4.4.2.4. ROC analysis. To better understand the trade-off between the FPR and TPR of MOGFIDS and other baseline classifiers, they are evaluated by the ROC curves, which are obtained by varying their decision thresholds for classification. In general, the amount of intrusion attacks (positive samples) is usually smaller than that of normal traffic (negative samples), which dominates in the intrusion detection domain, so that the performance curve near to the northwest point (0,1) of ROC graph becomes particularly important. The ROC curve can also be used to determine the performance of IDS for different operating points so that different configurable thresholds can be used for different IDS deployment locations and strategies in a network. The results in Fig. 13 show that the ROC curve obtained with MOGFIDS compares very favorably with those of other baseline classifiers over the KDD-Cup99 test data. It

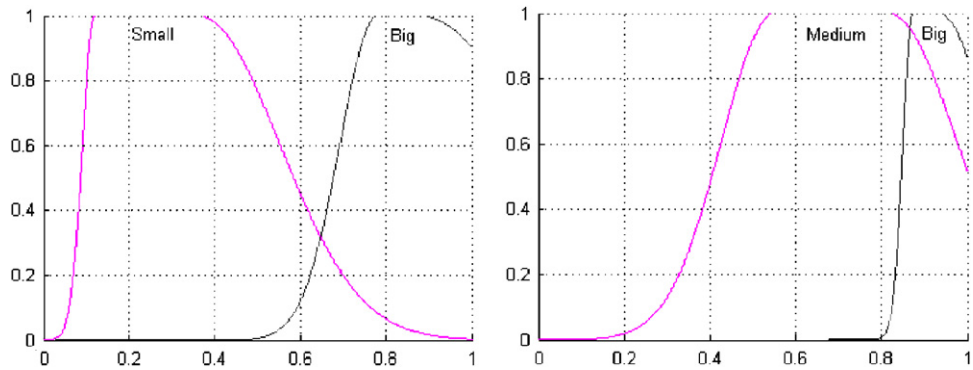

Fig. 12. Distribution of fuzzy sets of feature with (a) index #46 and (b) index #51.

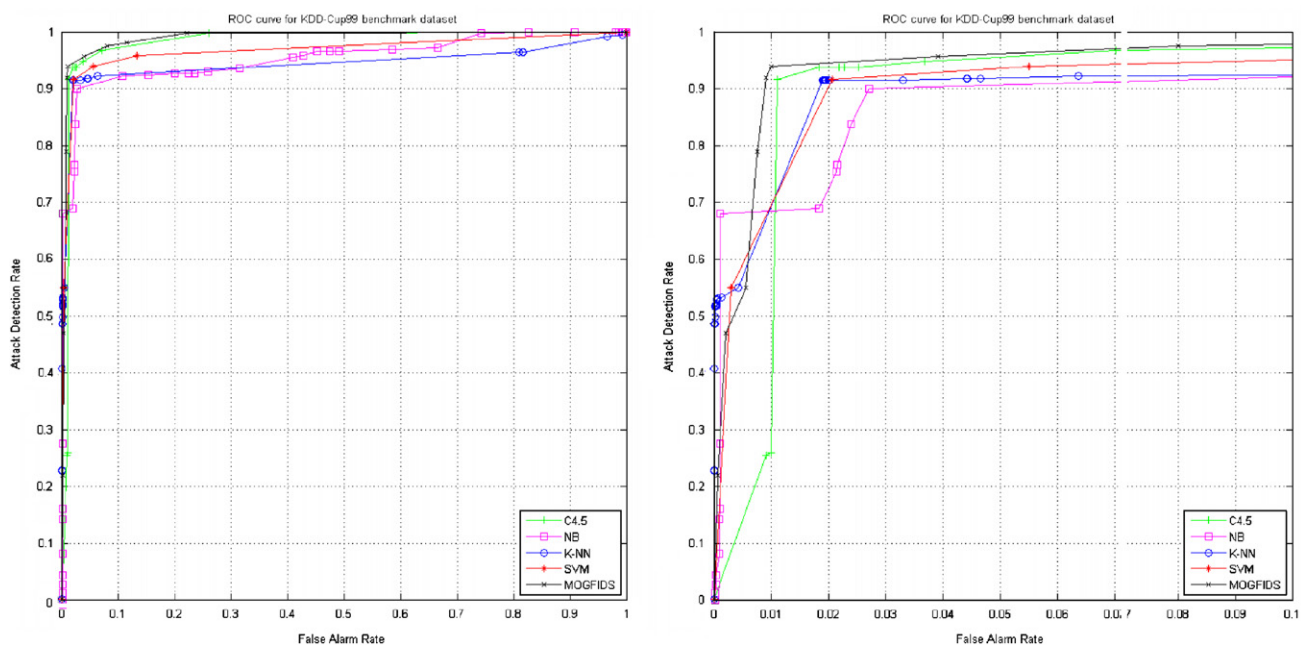

Fig. 13. ROC curves of different baseline classifiers and MOGFIDS evaluated on KDD-Cup99 test data. For clear illustration, the false alarm rate in the ROC graph is shown with (a) a full interval [0,1] and (b) a remarkable interval [0,0.1].

is found that the NB and k-NN classifiers suffer from high false alarm rates when they obtain high attack detection rates. The C4.5 outperforms SVM in overall performance, however, it is outperformed by the MOGFIDS that gains the largest area under ROC and obtains the lowest false alarm rate when all the intrusion attacks are correctly classified. A fortiori, these comparative results demonstrate the robust performance can be achieved by the proposed MOGFIDS on detecting both known and unseen attacks with high ACC, and recognizing the normal network traffic with low FPR.

5. Conclusions

In this work, we have addressed two important issues for anomaly intrusion detection: (i) generating accurate and interpretable fuzzy systems for classification and (ii) evaluating the feature selection techniques for intrusion detection domain. Automatic generation of rule-based knowledge by data mining approaches has been widely adopted owing to its considerable classification accuracy. However, attention has not been paid to the interpretability optimization of rule-based systems, which is also important for intrusion analysis and human comprehension. We have presented a novel intrusion detection approach that extracts accurate and interpretable fuzzy rule-based knowledge from network traffic data using an agent-based evolutionary framework. The experimental results demonstrate that the agent cooperation and competition are effective to exchange fuzzy set information such that the widespread non-dominated fuzzy systems can be obtained in our approach considering both accuracy and interpretability.

Feature selection can be used not only to alleviate the curse of dimensionality and minimize classification errors, but also to improve the interpretability of rule-based classifiers. To evaluate the effectiveness of our approach in the aspect of feature

selection, it is compared with some well-known feature selection filters and wrappers in terms of the feature selection performance and classification results. It is found that the feature subset searched by our approach retains the relevant features and removes most of the irrelevant features found by different baseline filters. In addition, when our approach is compared with different classifiers and wrappers, it shows that the overfitting problem can be relatively alleviated if it is required to learn with small training samples of rare classes and evaluate with large test samples containing novel attack classes. The results demonstrate that our approach encouragingly outperforms all the baseline classifiers and wrappers by providing the highest detection accuracy and the lowest classification cost. In terms of the F-measure, it scores the best on the U2R and R2L class categories, and also the second best on the Probe, DOS and normal class categories. As our approach can obtain the largest area under ROC curve as well as the lowest false alarm rate when all the intrusion attacks can be correctly classified in the ROC graph, it further supports the robust performance of our approach. The extensive experimental results in this paper have shown the successful classification of sophisticated intrusion attacks and normal network traffic, hence there is much scope for future work to apply our approach to other complex problem domains such as face recognition and DNA computing, which can be studied with accurate and interpretable fuzzy systems.

Acknowlegment

The work described in this paper was supported by a grant from City University Strategic Grant 7001955.

References

[1] W. Lee, S.J. Stolfo, K.W. Mok, Adaptive intrusion detection: a data mining approach, Artif. Intell. Rev. 14 (6) (2000) 533–567.

[2] M. Ramadas, S. Ostermann, B. Tjaden, Detecting anomalous network traffic with self-organizing maps, in: Sixth International Symposium on Recent Advances in Intrusion Detection, RAID'03, September 2003, pp. 36–54.

[3] P.D. Williams, K.P. Anchor, J.L. Bebo, G.H. Gunsch, G.D. Lamont, CDIS: towards a computer immune system for detecting network intrusions, in: Fourth International Symposium on Recent Advances in Intrusion Detection, RAID'01, October 2001, pp. 117–133.

[4] W. Lee, S.J. Stolfo, A framework for constructing features and models for intrusion detection systems, ACM Transactions on Information and System Security, TISSEC, vol. 3(4), November 2000, pp. 227–261.

[5] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Syst. Man Cybern. SMC-3 (1973) 28–44.

[6] G. Florez, S.M. Bridges, R.B. Vaughn, An improved algorithm for fuzzy data mining for intrusion detection, in: Proceedings of North American Fuzzy Information Processing Society Conference, NAFIPS 2000, New Orleans, LA, June 2002, pp. 457–462.

[7] J.E. Dickerson, J. Juslin, O. Koukousoula, J.A. Dickerson, Fuzzy intrusion detection, in: Proceedings of IFSA World Congress and 20th North American Fuzzy Information Processing Society Conference, NAFIPS 2001, Vancouver, British Columbia, July 2001, pp. 1506–1510.

[8] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[9] J. Gomez, D. Dasgupta, Evolving fuzzy classifiers for intrusion detection, in: Proceedings of IEEE Workshop on Information Assurance, United States Military Academy, West Point, New York, June 2001, pp. 68–75.

[10] T. Özyer, R. Alhajj, K. Barker, A boosting genetic fuzzy classifier for intrusion detection using data mining techniques for rule prescreening, in: A. Abraham, M. Köppen, K. Franke (Eds.), Design and Application of Hybrid Intelligent Systems, IOS Press, Amsterdam, 2003, pp. 983–992.

[11] G. Helmer, J.S.K. Wong, V. Honavar, L. Miller, Automated discovery of concise predictive rules for intrusion detection, Syst. Softwares 60 (3) (2002) 165–175.

[12] W.W. Cohen, Fast effective rule induction, in: Proceedings of the 12th International Conference on Machine Learning, Lake Tahoe, CA, July 1995, pp. 115–123.

[13] R. Agarwal, M.V. Joshi, PNrule: a new framework for learning classifier models in data mining (a case-study in network intrusion detection), in: Proceedings of First SIAM Conference on Data Mining, Chicago, April 2001.

[14] K. Julisch, M. Dacier, Mining intrusion detection alarms for actionable knowledge, in: Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining, Edmonton, July 2002, pp. 366–375.

[15] O. Boz, Feature subset selection by using sorted feature relevance, in: Proceedings of International Conference on Machine Learning and Applications, Las Vegas, NV, USA, June 2002.

[16] H.L. Wang, S. Kwong, Y. Jin, W. Wei, K.F. Man, Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction, Fuzzy Sets and Systems 149 (1) (2005) 149–186.

[17] M. Setnes, R. Babuška, U. Kaymak, H.R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. Syst. Man Cybern.—Part B: Cybernetics 28 (3) (1998) 376–386.

[18] Y. Jin, W. von Seelen, B. Sendhoff, On generating FC3 fuzzy rule systems with data using evolution strategies, IEEE Trans. Syst. Man Cybern.—Part B: Cybernetics 29 (6) (1999) 829–845.

[19] Y. Jin, Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. Fuzzy Syst. 8 (2) (2000) 212–221.

[20] H. Roubos, M. Setnes, Compact and transparent fuzzy models and classifiers through interactive complexity reduction, IEEE Trans. Fuzzy Syst. 9 (4) (2001) 516–524.

[21] P. Meesad, G.G. Yen, Quantitative measures of the accuracy, comprehensibility, and completeness of a fuzzy expert system, IEEE International Conference on Fuzzy Systems, vol. 1, May 2002, pp. 284–289.

[22] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[23] K.S. Tang, K.F. Man, Z.F. Liu, S. Kwong, Minimal fuzzy memberships and rules using hierarchical genetic algorithms, IEEE Trans. Ind. Electron. 45 (1) (1998) 162–169.

[24] K.F. Man, K.S. Tang, S. Kwong, Genetic algorithms: concepts and applications, IEEE Trans. Ind. Electron. 43 (5) (1996) 519–534.

[25] K.F. Man, K.S. Tang, S. Kwong, Q. He, Genetic algorithms and their applications, IEEE Signal Process. Mag. 13 (6) (1996) 22–37.

[26] L. Magdalena, O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (1) (2004) 5–31.

[27] H. Ishibuchi, T. Nakashima, T. Kuroda, A hybrid fuzzy GBML algorithm for designing compact fuzzy rule-based classification systems, in: Proceedings of the Ninth IEEE International Conference on Fuzzy Systems, San Antonio, May 2000, pp. 706–711.

[28] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Trans. Fuzzy Syst. 9 (4) (2001) 506–515.

[29] H. Ishibuchi, K. Nozaki, H. Tanaka, Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems 52 (1) (1992) 21–32.

[30] UCI Machine Learning Repository (Online), Available: ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩.

[31] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185–208.
[32] J.C. de Borda, Mémoire sur les elections au scrutin, 1781, English translation by A. de Grazia, Math. Derivation Election Syst. Isis 44 (1953) 42–51.
[33] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Lett. 10 (1989) 335–347.
[34] M. Middlemiss, G. Dick, Feature selection of intrusion detection data using a hybrid genetic algorithm/KNN approach, in: A. Abraham, M. Köppen, K. Franke (Eds.), Design and Application of Hybrid Intelligent Systems, IOS Press, Amsterdam, 2003, pp. 519–527.
[35] C. Elkan, Results of the KDD'99 classifier learning, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, vol. 1(2), 2000, pp. 63–64.
[36] S. Chebrolu, A. Abraham, J. Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems, Comput. Secur. 24(4) (2005) 295–307.
[37] J. Gomez, D. Dasgupta, Evolving fuzzy classifiers for intrusion detection, in: Proceedings of IEEE Workshop on Information Assurance, United State Military Academy, West Point, NY, 2001, pp. 68–75.
[38] S. Maheshkumar, S. Gursel, Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context, in: Proceedings of International Conference on Machine Learning, Models, Technologies and Applications, Las Vegas, Nevadat, USA, CSREA Press, 2003, pp. 209–215.

About the Author—Chi-Ho Tsang received the B.Sc. degree with first-class honors in Computer Science from University of Hertfordshire, UK, in 2003 and the M.Phil. degree in Computer Science from City University of Hong Kong in 2005. From 2003 to 2005, he involved in various research projects related to genetic-fuzzy rule mining, multi-objective data clustering, ant colony clustering, anomaly intrusion detection, and evolutionary job-shop scheduling. In 2006, he joined MTR Corporation Limited, a mass transit railway corporation in Hong Kong, where he contributed to the development of automatic fare collection data analysis system, management information module, as well as data warehousing. Currently, he is a Business Analyst at Hang Seng Bank Limited, Hong Kong, where he works on credit risk management IT projects. His research interests are in the fields of pattern recognition, data mining, multi-objective genetic algorithms, bio-inspired evolutionary algorithms, mobile robotics, and network security.

About the Author—SAM KWONG received his B.Sc. degree and M.A.Sc. degree in electrical engineering from the State University of New York at Buffalo, USA and University of Waterloo, Canada, in 1983 and 1985, respectively. He later obtained his Ph.D. from the University of Hagen, Germany. From 1985 to 1987, he was a diagnostic engineer with the Control Data Canada where he designed the diagnostic software to detect the manufacture faults of the VLSI chips in the Cyber 430 machine. He later joined the Bell Northern Research Canada as a Member of Scientific staff where he worked on both the DMS-100 voice network and the DPN-100 data network project. In 1990, he joined the City University of Hong Kong as a lecturer in the Department of Electronic Engineering. He is currently an associate Professor in the Department of Computer Science. Dr. Kwong received the Best Paper Award in the BioInformatics Workshop 1999, Tokyo, for the paper entitled "*A Compression Algorithm for DNA Sequences and Its Application in Genome Comparison*" in recognition of his outstanding contribution to the conference. Currently, this algorithm has the best compression results for compressing DNA data sequences. His research interests are in Genetic Algorithms, Speech Processing and Recognition, Digital Watermarking, Data Compression and Networking.

About the Author—Hanli Wang received his B.Sc. and M.Sc. degrees, both with the First-Class Honor in electrical engineering from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively. From March 2003 to March 2004, he was a Research Assistant at the Department of Computer Science, City University of Hong Kong, Hong Kong, China. From April 2004 to July 2004, he worked at the Digital Manufacturing Lab, GE Global Research-Shanghai, Shanghai, China. He is currently pursuing the Ph.D. degree at the Department of Computer Science, City University of Hong Kong, Hong Kong, China. His research interests include digital video coding, image processing, evolutionary computation, and pattern recognition. He received a number of scholarships and awards during his Ph.D. study period such as the City University of Hong Kong Research Tuition Scholarship from 2005 to 2007, and the City University of Hong Kong Outstanding Academic Performance Award in October 2006.