Dendritic Cell Algorithm with Fuzzy Inference System for Input Signal Generation*

Noe Elisa, Jie Li, Zheming Zuo, and Longzhi Yang

Northumbria University, Newcastle upon Tyne, NE1 8ST, UK
{ noe.nnko, jie.li, zheming.zuo, longzhi.yang}@northumbria.ac.uk

Abstract. Dendritic cell algorithm (DCA) is a binary classification system developed by abstracting the biological danger theory and the functioning of human dendritic cells. The DCA takes three signals as inputs, including danger, safe and pathogenic associated molecular pattern (PAMP), which are generated in its pre-processing and initialization phase. In particular, after a feature selection process for a given training data set, each selected attribute is assigned to one of the three input signals. Then, these input signals are calculated as the aggregation of their associated features, usually implemented by a simple average function followed by a normalisation process. If a nonlinear relationship exists between a signal and its corresponding selected attributes, the resulting signal using the average function may negatively affect the classification results of the DCA. This work proposes an approach named TSK-DCA to address such limitation by aggregating the assigned features of a signal linearly or non-linearly depending on their inherit relationship using the TSK+ fuzzy inference system. The proposed approach was evaluated and validated using the popular KDD99 data set, and the experimental results indicate the superiority of the proposed approach compared to its conventional counterpart.

Keywords: Dendritic cell algorithm, TSK+ fuzzy inference system, information aggregation, danger theory.

1 Introduction

Intrusion detection systems (IDSes) are of paramount importance in computer network security as the number of cyber attacks grow in prominence every year. Over the last three decades, artificial immune systems (AISes) have been proposed primarily for intrusion detection in computer systems. Self-nonself is the first biological model used in computer security domain to develop AISes such as clonal selection, negative selection and positive selection algorithms [1]. Self-nonself model is built upon the observation that the natural immune system provides protection based on the discrimination between self (own body cells) which is tolerated and nonself (foreignness) which is the source of attack [2].

* This work is supported by the Commonwealth Scholarship Commission (CSC) and Northumbria University in the United Kingdom.

Self-nonself based algorithms often fail to provide beneficial advantages to computer network security systems like those provided by the natural self-nonself model to the natural immune systems [1].

Inspired by the danger theory [3] and the behaviour of dendritic cells (DCs), the dendritic cell algorithm (DCA) was developed to address the aforementioned limitation [4]. Briefly, the DCA first transfers the values of the most relevant features from a given training dataset to its input signals, termed as safe, danger, and pathogenic associated molecular pattern (PAMP). Then, the DCA classifier takes those input signals to produce a binary output. Conventionally, a linear average aggregation method is commonly used to aggregate the values from the assigned features to form each input signal. However, if a non-linear relationship exists between the selected attributes and the resulting signals, the average approach will adversely impact the performance of the DCA. A fuzzy inference system is then adopted in this paper to compute the value of each DCA input signal to generalise the linear average aggregation method.

Fuzzy inference systems are built upon fuzzy logic to map from the input space to the output space. They have been widely applied in solving either linear or non-linear problems of arbitrary complexity, such as [5, 6]. The two most widely used fuzzy inference systems are the Mamdani fuzzy model and TSK fuzzy model. Compared with the Mamdani fuzzy model, which is more intuitive and commonly utilised to deal with human natural language, the TSK fuzzy model is more convenient to be employed when crisp output values are required. Both of these conventional fuzzy inference systems are only workable with a dense rule base by which the entire input domain is fully covered. Fuzzy interpolation enhances the power of the conventional fuzzy inference systems by relaxing the requirement of dense rule bases [7, 8]. In other words, the conventional fuzzy inference systems fail to generate a conclusion when a given observation does not overlap with any rule antecedents in the rule base, but fuzzy interpolation can still approximate the conclusion. Various fuzzy interpolation methods have been developed in the literature, such as [9–17].

This paper proposes the TSK-DCA approach for aggregating the assigned features of each input signal, either linearly or non-linearly, to generate DCA inputs using the TSK+ fuzzy inference approach. In particular, the TSK-DCA uses three TSK+ fuzzy inference systems to deal with the three DCA input signals. In order to implement the proposed TSK-DCA, a data-driven rule base generation method is firstly employed to generate three sub-TSK fuzzy rule bases, corresponding to the three input signals. Then, the TSK+ fuzzy inference approach is applied to compute the value of each input signal from the assigned features for each data instance, before the application of the DCA. TSK-DCA has been validated and evaluated by a well-known benchmark dataset, KDD99. Experimental results indicate that the TSK-DCA performs better than the conventional one.

The rest of this paper is organised as follows: Section 2 describes the background theories, including TSK+ fuzzy inference approach and the DCA algorithm. Section 3 details the proposed TSK-DCA approach. Section 4 reports the experimentation and analyses the results; and Section 5 draws the conclusion and points out future research directions.

2 Background

2.1 TSK+ Fuzzy Inference System

The original TSK inference system generates a crisp inference result as the weighted average of the sub-consequences with the firing strength of the fired rules as weights [18]. Obviously, no rule will be fired if a given input does not overlap with any rule antecedent. As a consequence, the TSK inference cannot be performed. TSK+ was proposed to address such issue which generates a consequence by considering all the rules in the rule base [19]. Suppose that a sparse TSK rule base is comprised of n rules:

$$\begin{aligned} R_1 : &\textbf{IF } x_1 \text{ is } A_1^1 \text{ and } \cdots x_j \text{ is } A_j^1 \cdots \text{ and } x_m \text{ is } A_m^1 \\ &\textbf{THEN } z = f_1(x_1, \cdots, x_m), \\ &\dots\ \dots \\ R_n : &\textbf{IF } x_1 \text{ is } A_1^n \text{ and } \cdots x_j \text{ is } A_j^n \ \cdots \text{ and } x_m \text{ is } A_m^n \\ &\textbf{THEN } z = f_n(x_1, \cdots, x_m), \end{aligned} \tag{1}$$

where $A_j^i, (i \in \{1, 2, \cdots, n\}$ and $j \in \{1, 2, \cdots, m\})$ represents a normal and convex polygonal fuzzy set that can be denoted as $(a_{j1}^i, a_{j2}^i, \cdots, a_{jv}^i)$, v is the number of odd points of the fuzzy set. Given an input $I = (A_1^*, A_2^*, \cdots, A_m^*)$ in the input domain, a crisp inference result can be generated by the following steps:

Step 1: Identify the matching degrees between the given input $(A_1^*, A_2^*, \cdots, A_m^*)$ and rule antecedents $(A_1^i,\ A_2^i,\ A_3^i, \cdots, A_m^i)$ for each rule R_i by:

$$S(A_j^i, A_j^*) = \left(1 - \frac{\sum\limits_{q=1}^{v} |a_{jq}^i - a_{jq}^*|}{v}\right) \cdot (DF)\ , \tag{2}$$

where DF is a $distance\ factor$, which is a function of the distance between the two concerned fuzzy sets:

$$DF = 1 - \frac{1}{1 + e^{(-cd+5)}}\ , \tag{3}$$

where c is a sensitivity factor, and d represents the Euclidean distance between the two fuzzy sets for a given defuzzification approach. in particular, c is a positive real number. Smaller value of c leads to a similarity degree which is more sensitive to the distance of two fuzzy sets, and vice versa.

Step 2: Determine the firing degree of each rule by aggregating the matching degrees between the given input and its antecedent terms by:

$$\alpha_i = S(A_1^*, A_1^i) \wedge S(A_2^*, A_2^i) \wedge \cdots \wedge S(A_m^*, A_m^i)\ , \tag{4}$$

where $\wedge$ is a t-norm operator usually implemented as a minimum operator.

Step 3: Generate the final output by integrating the sub-consequences from all rules by:

$$z = \sum_{i=1}^{n} \alpha_i \cdot f_n(x_1, \cdots, x_m) \;/\; \sum_{i=1}^{n} \alpha_i \;. \tag{5}$$

2.2 Dendritic Cell Algorithm

In order to detect anomaly for any given inputs, the DCA creates a population of artificial DCs to form a pool from which a number of DCs are selected to perform antigens (data items) sampling, signals categorization (into PAMP, DS and SS) and antigens identification [4]. While in the pool, DCs are exposed to the current signal values and the corresponding antigen data from the data source. Each DC has the ability to sample multiple antigens, so during the classification an aggregated sampling value from different DCs regarding a particular antigen is computed which is used to classify the antigen as normal or anomalous [4, 20].

The inputs to the DCA are signals from all categories and data items. Signals are represented as an aggregation of real-valued numbers from their corresponding associated features, while antigens are identified by the data item IDs such as process ID or any other unique nominal attributes. As a binary classifier, the DCA classifies each antigen as either normal (semi-mature) cell context, or as anomalous (mature) cell context. So, the DCA output is the antigen normality or abnormality context which is represented as a binary value 0 for normality or 1 for abnormality. After the pre-processing and initialisation, the DCA goes through three phases as detailed below.

Step 1. Detection The DCA processes the input signals using the following equation to obtain three cumulative output signals termed as CSM, mDC and $smDC$:

$$C[CSM, smDC, mDC] = \frac{(W_{PAMP} * C_{PAMP}) + (W_{SS} * C_{SS}) + (W_{DS} * C_{DS})}{W_{PAMP} + W_{SS} + W_{DS}} * \frac{1+I}{2}, \tag{6}$$

where C_{PAMP}, C_{DS} and W_{SS} are PAMP, DS and SS signal values respectively which are generated by aggregating the assigned attributes. The weights (W_{PAMP}, W_{SS} and W_{DS}) are pre-defined weights or can be derived empirically from the data. Each selected DC from the pool is assigned a migration threshold in order to determine the lifespan for antigen sampling and the amount of data items it can collect. Each DC computes its CSM value and compares it with the migration threshold. If the CSM of a DC exceeds the migration threshold, the DC ceases to sample data items and thus signals.

Step 2. Context Assessment The cumulative values of $smDC$ and mDC obtained from the detection phase are used to perform context assessment. If the antigens collected by a DC has a greater mDC than its $smDC$ value, it is assigned a binary value of 1, and 0 otherwise.

Step 3. Classification All the collected antigens are analysed by deriving the Mature Context Antigen Value (MCAV) for each presented antigen, which is used to assess the degree of an anomaly of a given antigen. Firstly, the anomaly threshold of MCAV is derived from the test data set. Then, the MCAV value of each antigen is calculated as dividing the number of times it is presented in the mature context by the total number of presentations in DCs. Antigens with MCAVs greater than the anomaly threshold are classified into the anomalous class while the others are classified into the normal one.

3 The Proposed Approach

The proposed TSK-DCA system is depicted in Figure 1. In particular, given a training dataset, a feature selection process is first performed to select the most significant features. The selected features are then categorised into three groups, representing the three input signals. From this, three TSK+ fuzzy models can be generated using the given training data set for the aggregation of the three input values. Given an input, the TSK+ inference systems take place to aggregate the given inputs to the three DCA input signals. Then, the output of the TSK-DCA classifier is generated by the DCA model. Each of these key components of the proposed system is detailed in the following subsections.

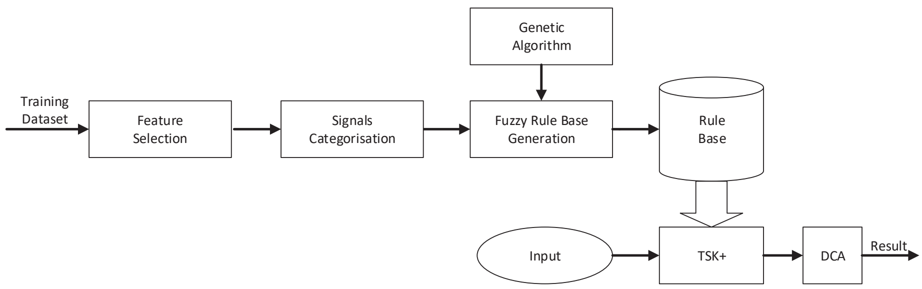

Fig. 1. The overall TSK-DCA system

3.1 Data Pre-processing

This study adopted the information gain approach to decide which features are more important than others during the data pre-processing stage, although any other feature selection approach is also applicable here. Briefly, the information gain of an attribute indicates the amount of information with respect to the classification a particular attribute provides, which can be obtained by [21]:

$$G(D, A) = E(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} * E(D_v), \tag{7}$$

where $values(A)$ represent all the possible values of attribute A, D_v is a subset of D each taking value v for attribute A, G is the gain, and E is the entropy. In particular, the entropy E is computed as:

$$Entropy(D) = \sum_{i=1}^{i=2} -p_i * log_2 p_i, \tag{8}$$

where p_i is the proportion of elements being classified as i in the data set D. The higher the entropy the more information an attribute contains. Given a threshold, the attributes with higher gains than the given threshold are selected.

3.2 Signal Categorisation

The selected features are analysed using their histograms with respect to the two class labels (normal and abnormal) presented in the training dataset. The frequency of occurrence of the largest values presented in each attribute from each class is used to decide its signal category. If the largest values of an attribute have a high frequency of occurrence in normal class than that in anomalous class, the attribute will be categorized as safe signal. If the largest values of an attribute have a higher frequency of occurrence in anomalous class and significant lower frequency of occurrence in the normal class, it is categorised to PAMP signal. Otherwise, it is assigned to DS signal.

3.3 Signal Generation Using TSK+

Once the selected features are categorised into the three input signals, the TSK+ approach is applied to generate the input signals of the DCA. In order to apply the TSK+ approach as introduced in Section 2.1, a rule base needs to be generated first, which is outlined in Figure 2 in two key steps as detailed below.

Clustering: The K-Means clustering algorithm is employed to each sub-dataset (i.e., danger, safe or PAMP) which includes only the associated features for the particular input signal. Note that the number of clusters has to be predefined to enable the application of the K-Means algorithm. The Elbow method, which has been used in [6, 22], is also employed in this work to determine the number of clusters.

Fuzzy Rule Extraction: Each obtained cluster is expressed as one TSK fuzzy rule. Assume that a determined cluster for a signal is associated with d features, then a TSK fuzzy rule R_i can be formed as:

$$\begin{aligned} R_i : &\textbf{IF } x_1 \text{ is } A_1^i \text{ and ... and } x_d \text{ is } A_d^i \\ &\textbf{THEN} y = f_i(x_1, ..., x_d) \ , \end{aligned} \tag{9}$$

where A_r^i ($r = \{1, ..., d\}$) is a fuzzy set as a rule antecedent. For simplicity, triangular membership functions are utilised in this work, that is $A_r^i = (a_{r1}^i, a_{r2}^i, a_{r3}^i)$. Without loss of generality, take a rule cluster c_k as an example, which contains p_k elements, such as $c_k = \{x_k^1, x_k^2, ..., x_k^{p_k}\}$. The core of

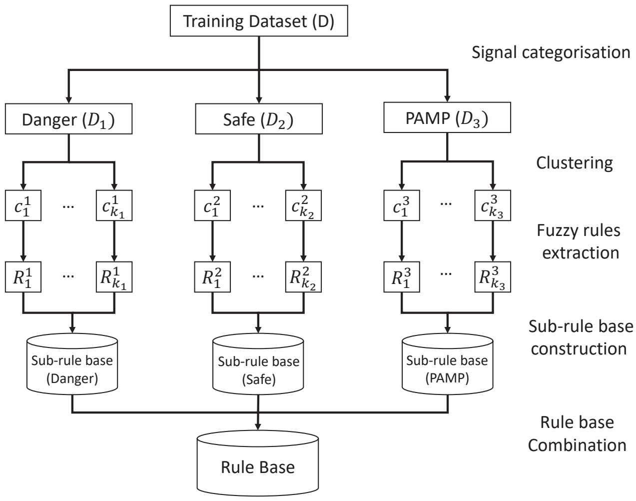

Fig. 2. The TSK+ fuzzy rule base generation

the fuzzy set is set as the cluster centre which is $a^i_{r2} = \sum_{q=1}^{p_k} x_k^{qr}/p_k$; and the support the fuzzy set is expressed as the span of the cluster, i.e. $(a^i_{r1}, a^i_{r3}) = (\min\{x_k^{1r}, x_k^{2r}, ..., x_k^{p_k r}\}, \max\{x_k^{1r}, x_k^{2r}, ..., x_k^{p_k r}\})$. The consequent of a TSK fuzzy rule is the DCA input signal values. In particular, the consequent is expressed as a first-order polynomial in this work, which can be represented as $y = f_i(x_1, ...x_d) = \beta_0^i + \beta_1^i x_1 + \beta_2^i x_2 + ... + \beta_d^i x_d$, where β_i^d is a constant parameter of the linear functions.

The rule base is optimised by employing the genetic algorithm (GA). As an adaptive heuristic search algorithm, GA has been successfully applied to find the optimised solution in the problem of fuzzy inference systems, such as [22–24]. The algorithm firstly initialises the population with random individuals. It then selects a number of individuals for reproduction by applying the genetic operators, that is mutation and crossover. The offspring and some of the selected existing individuals jointly form the next generation. The algorithm repeats this process until a satisfactory solution is generated or a maximum number of generations has been reached.

In this work, an individual (I) in a population ($\mathbb{P}$) is used to represent a potential solution that contains all the parameters of the polynomial functions in the TSK rule consequent, represented as $I = \{\beta_0^1, ..., \beta_d^1, ..., \beta_0^i, ..., \beta_d^i, \beta_0^n, ..., \beta_d^n\}$, where n denotes the total number of rules in the current rule base. Given a population, represented as $\mathbb{P} = \{I_1, ..., I_{|\mathbb{P}|}\}$, where $|\mathbb{P}|$ is the numbers of individuals, the next generation of a population is produced by applying a single point crossover and a mutation, on selected individuals. The DCA classification accuracy is used to evaluate the quality of individuals in the new generation of population. After the algorithm is terminated, the fittest individual in the current population is the optimal solution. From this, all the extracted rules are grouped together to form the final rule base.

Once the rule bases are generated for all three input signals, the TSK+ inference approach as introduced in Section 2.1 is applied, which generates the signal inputs for the DCA as illustrated in Figure 3.

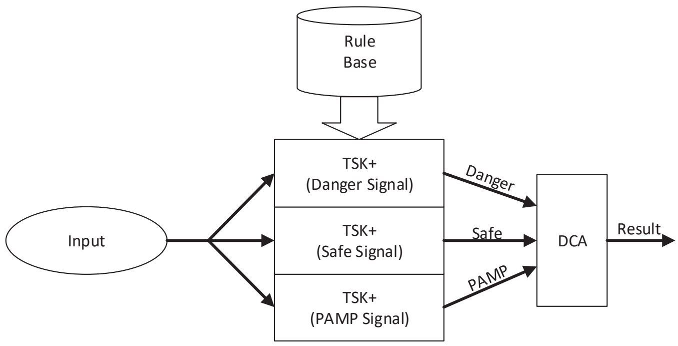

Fig. 3. Inputs generation for DCA

4 Experimentation

The proposed TSK-DCA system was validated and evaluated using the KDD-99 cup dataset [25]. The KDD-99 dataset was published in 1999 in the context of the 1998 DARPA initiative for IDS within the realm of computer networks [25]. This dataset has been intensively employed for building network intrusion detectors to distinguish normal and abnormal (i.e., intrusions or attacks) network connections. 10% (494,021) and 2.5% (125,973) data instances of the original KDD-99 dataset were respectively used for training and testing in this work.

4.1 Model Generation

In order to reduce the system complexity, the information gain method is employed for feature selection. Ten features were typically selected from 41 ones [26], and this work also follows this tradition, and the selected features for each signal category are listed in Table 1. From this, the dataset was normalised using the min-max (MM) normalisation approach [27].

The rule base was generated in three step based on the training dataset:

Step 1: Training Dataset Partition. Divided the entire training dataset T into three sub-training dataset T_1, T_2 and T_3 based on the results of the information gain method.

Table 1. Selected features for each DCA input signal

Signal	Features
DS	count and srv count
SS	logged in, srv different host rate and dst host count
PAMP	serror rate, srv serror rate, same srv rate, dst host serror and dst host rerror rate

Step 2: Optimal Number of Clusters Determination for Each Sub-Training Dataset. The K-Means clustering algorithm was adopted for each sub-training dataset in which the optimal number of clusters was determined by the Elbow method. The identified cluster numbers for the three sub-datasets are listed in Table 2.

Table 2. The number of clusters for each sub-dataset

	DS	SS	PAMP
Number of clusters	7	7	7

Step 3: TSK Rule Extraction. Based on the results of the Elbow method, there were 21 TSK fuzzy rules in total in the final rule base. For instance, the rule antecedents of one fuzzy rule in DS sub-rule base can be expressed as:

$$x_1 = (0, 0.39, 3.52) \text{ and } x_2 = (0, 0.39, 31.51). \tag{10}$$

Step 4: Fine-Tune Polynomial Coefficients for TSK Consequence. GA was applied to find the optimised constant parameters of polynomial functions of TSK consequent, and the results are listed in Table 3.

Table 3. The employed GA parameters

Number of Individuals	50
Number of Iterations	250
Crossover Rate	0.95
Mutation Rate	0.1

Taking Equation 10 as an example, the optimised fuzzy rule is:

$$\begin{aligned} R_1 : \textbf{IF } & x_1 = (0, 0.39, 3.52) \text{ and } x_2 = (0, 0.39, 31.51) \\ \textbf{THEN } & f_1(x_1, x_2) = 18.2x_1 - 4.05x_2 - 5.5 \ . \end{aligned} \tag{11}$$

Once the TSK fuzzy rule base has been generated, three TSK+ fuzzy inference systems were applied to generate the three input signals for the DCA model. In this work, the DCA model reported in [26] was employed and the corresponding parameters for Equation 6 were configured as shown in Table 4. Note that, antigen multiplier was not used in this work.

Table 4. DCA parameters

	smDC			*mDC*		
	W_{PAMP}	W_{SS}	W_{DS}	W_{PAMP}	W_{SS}	W_{DS}
Weights	5	11	11	8	7	14

4.2 Results and Discussion

Based on the parameter values shown in Tables 3 and 4, the best performance for the training dataset is 98.36%, and for the testing dataset is 92.07%. The accuracies for the 250 iterations of GA optimisation is shown in Figure 4. The proposed TSK-DCA approach was compared with the basic DCA without using antigen multiplier as presented by [26] and Fuzzy-based DCA proposed by [28]. The proposed TSK-DCA approach overall outperforms the two benchmark approaches as demonstrated in Table 5.

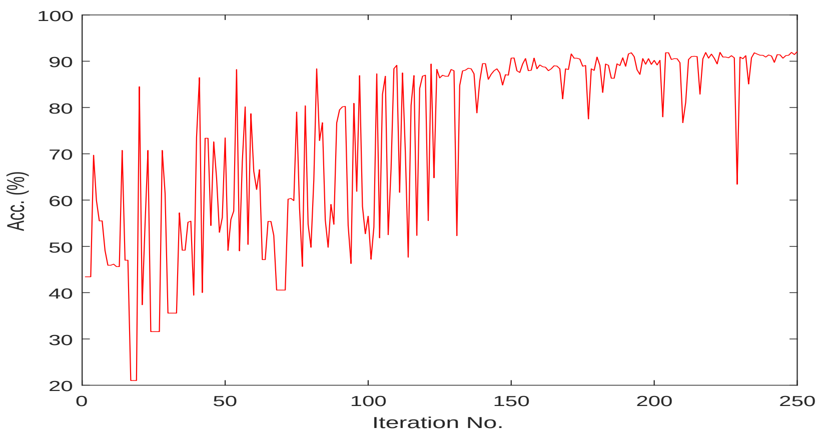

Fig. 4. The processing of fine-tuning the testing accuracies

Table 5. Performance comparison with existing DCA approaches

Approach	Acc. (%)	
	Training	Testing
Basic DCA without antigen multiplier [26] (2008)	78.92	-
Fuzzy-based DCA [28] (2015)	94.00	-
TSK-DCA	**98.36**	**92.07**

5 Conclusion

This work proposed the TSK-DCA classifier which generates the DCA input signal values from the assigned attributes using the TSK+ fuzzy inference system. The TSK-DCA is applicable to either linear or nonlinear related data instances. The experimental results using the KDD99 dataset demonstrate that the TSK-DCA achieves better classification accuracy in reference to its conventional DCA counterparts. Although promising, the work can be further improved by better fine-tune the rule base parameters as the present work only trains the parameters for the rule consequences. In addition, it is interesting to combine the proposed approach with other extensions and modifications of DCA to further boost the performance.

References

1. Uwe Aickelin and Steve Cayzer. The danger theory and its application to artificial immune systems. *arXiv preprint arXiv:0801.3549*, 2008.
2. Frank M Burnet. Immunological recognition of self. *Science*, 133(3449):307–311, 1961.
3. Polly Matzinger. The danger model: a renewed sense of self. *Science*, 296(5566):301–305, 2002.
4. Julie Greensmith, Uwe Aickelin, and Steve Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *ICARIS*, volume 3627, pages 153–167. Springer, 2005.
5. J. Li, L. Yang, X. Fu, F. Chao, and Y. Qu. Dynamic qos solution for enterprise networks using tsk fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, July 2017.
6. L. Yang, J. Li, G. Fehringer, P. Barraclough, G. Sexton, and Y. Cao. Intrusion detection system by fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, July 2017.
7. L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1):169–201, 1993.
8. LászlóT Kóczy and Kaoru Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.
9. Z. Huang and Q. Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.
10. L. Yang and Q. Shen. Adaptive fuzzy interpolation with prioritized component candidates. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 428–435, June 2011.
11. L.Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. Theme: Fuzzy Systems.
12. L. Yang and Q. Shen. Adaptive fuzzy interpolation with uncertain observations and rule base. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 471–478, 2011.
13. N. Naik, R. Diao, and Q. Shen. Genetic algorithm-aided dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2198–2205, 2014.

14. L. Yang, C. Chen, N. Jin, X. Fu, and Q. Shen. Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2184–2191, 2014.
15. J. Li, L. Yang, H. P. H. Shum, G. Sexton, and Y. Tan. Intelligent home heating controller using fuzzy rule interpolation. In *UK Workshop on Computational Intelligence*, 2015.
16. L. Yang, F. Chao, and Q. Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, (DOI: 10.1109/TFUZZ.2016.2582526), 2016.
17. J. Li, L. Yang, X. Fu, F. Chao, and Y. Qu. Dynamic QoS solution for enterprise networks using TSK fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017.
18. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(1):116–132, 1985.
19. J. Li, Y. Qu, H. P. H. Shum, and L. Yang. *TSK Inference with Sparse Rule Bases*, pages 107–123. Springer International Publishing, Cham, 2017.
20. Zeineb Chelly and Zied Elouedi. A survey of the dendritic cell algorithm. *Knowledge and Information Systems*, 48(3):505–535, 2016.
21. H Günes Kayacik, A Nur Zincir-Heywood, and Malcolm I Heywood. Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets. In *Proceedings of the third annual conference on privacy, security and trust*, 2005.
22. Jie Li, Longzhi Yang, Yanpeng Qu, and Graham Sexton. An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation. *Soft Computing*, 22(10):3155–3170, May 2018.
23. A. Kumar, D. P. Agrawal, and S. D. Joshi. A ga-based method for constructing tsk fuzzy rules from numerical data. In *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, volume 1, pages 131–136 vol.1, May 2003.
24. Y. Tan, J. Li, M. Wonders, F. Chao, H. P. H. Shum, and L. Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 110–117, July 2016.
25. KDD Cup 1999 Data. "http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/. Accessed: 2018-04-16.
26. Feng Gu, Julie Greensmith, and Uwe Aickelin. Further exploration of the dendritic cell algorithm: Antigen multiplier and time windows. In *International Conference on Artificial Immune Systems*, pages 142–153. Springer, 2008.
27. Zheming Zuo, Jie Li, Philip Anderson, Longzhi Yang, and Nitin Naik. Grooming detection using fuzzy-rough feature selection and text classification. In *Fuzzy Systems (FUZZ-IEEE), 2018 IEEE International Conference on*. IEEE, 2018.
28. Zeineb Chelly and Zied Elouedi. Hybridization schemes of the fuzzy dendritic cell immune binary classifier based on different fuzzy clustering techniques. *New Generation Computing*, 33(1):1–31, 2015.