Intrusion Detection System by Fuzzy Interpolation

Longzhi Yang, Jie Li, Gerhard Fehringer, Phoebe Barraclough, Graham Sexton
Department of Computer and Information Sciences
Faculty of Engineering and Environment, Northumbria University
Newcastle upon Tyne, United Kingdom NE1 8XT
Email: longzhi.yang@northumbria.ac.uk

Yi Cao
Nanjing Customes District P.R. China
Nanjing
Jiangsu, 210000
P.R. China

***Abstract*—Network intrusion detection systems identify malicious connections and thus help protect networks from attacks. Various data-driven approaches have been used in the development of network intrusion detection systems, which usually lead to either very complex systems or poor generalization ability due to the complexity of this challenge. This paper proposes a data-driven network intrusion detection system using fuzzy interpolation in an effort to address the aforementioned limitations. In particular, the developed system equipped with a sparse rule base not only guarantees the online performance of intrusion detection, but also allows the generation of security alerts from situations which are not directly covered by the existing knowledge base. The proposed system has been applied to a well-known data set for system validation and evaluation with competitive results generated.**

I. Introduction

Cyber security has become an elevated risk that is amongst the most pressing issues affecting businesses, governments, other organisations, and even individuals. This issue is expected to become more important in time, as more devices, i.e., 'the Internet of Things', are becoming connected to the Internet. Network security is the most important challenge in the field of cyber security, knowing that networks provide the essential access to others which need to be protected in cyberspace, including the computer systems and data. Serious network attacks can lead to damages on computer systems, network paralysis, data loss or leakage. Network intrusion detection systems (IDS) attempt to identify unauthorised, illicit, and anomalous behaviour based solely on network traffic to support decision making in network preventive actions by network administrators.

There are basically two types of IDSs, which inspect suspicious traffic through either signatures or anomaly. The signature-based approaches attempt to classify a network connection based on an already known signature knowledge base [1]. These approaches are only applicable to the detection of already known types of threats. Differently, anomaly detection approaches, such as [2], are introduced to also detect unknown types of attacks by identifying the behaviour of the network traffic that does not conform to any expected pattern in their knowledge bases. Common to both types of approaches is the requirement of a well-covered data set or knowledge base, which is not always readily available or obtainable. In addition, the resultant system may be very complex if sufficient data is available. This paper proposes a novel IDS using fuzzy interpolation approaches to address these limitations.

Fuzzy interpolation, initially proposed in [3], enhances conventional fuzzy inference systems, such as the Mamdani and the TSK approaches, to work with sparse rule bases which do not cover the entire problem domain [4]. In other words, when observations or system inputs do not overlap with any rule antecedents in the rule base, no rule can be fired by traditional fuzzy inference systems and thus outputs/conclusions can not be generated. However, by applying fuzzy interpolation, conclusions still can be obtained by means of interpolation. Note that, fuzzy interpolation also helps in system complexity reduction by removing those rules which can be approximated by their neighbours in the rule base. Various fuzzy interpolation methods working with Mamdani-style sparse rule bases have been developed and applied to real world-applications successfully, including [5]–[19].

TSK inference systems have also been extended to deal with sparse rule bases [20] (named as TSK-style fuzzy interpolation), given that sparse TSK-style rule bases may be resulted from a data-driven approach. This is also the case for the development of IDS in this project because the data set for network traffic is often sparse and imbalanced (i.e., some parts of the data set are dense and others are sparse). TSK-style fuzzy interpolation is built upon a modified similarity measure which calculates the similarity degrees between observations and rule antecedents. Different with the similarity measure used in the traditional TSK approach, the modified one always leads to a greater-than-0 similarity degree between any observation and rule antecedent even when they do not overlap at all. Thanks to this property, a global consequence can always be generated by integrating the results from all rules in the rule base.

The proposed IDS is built upon the TSK-style fuzzy interpolation. Having known the inference mechanism as introduced above, a 0-order TSK rule base is generated from historical traffic data. Firstly, the classical clustering algorithm K-Means is employed to group similar objects (data points) into clusters in terms of their Euclidean distances in the multi-dimensional problem domain. Then, each identified cluster of data is expressed as one TSK fuzzy rule. Finally, the generated TSK rule base comprises of rules which represent all the clusters. Thanks to the relaxation on rule base requirement, the proposed IDS is not only able to generate security alerts for known attack types, but also to detect potential unknown threats, as demonstrated in the experimentation.

The rest of the paper is structured as follows. Section II introduces the theoretical underpinnings of fuzzy interpolation, with a focus on the TSK fuzzy interpolation upon which this work is built. Section III presents the details of the proposed IDS system. Section IV describes the experimentation for demonstration and validation. Section V concludes the paper and suggests possible future developments.

II. Background

TSK-style fuzzy interpolation is able to perform fuzzy inferences on a sparse TSK rule base, which is introduced in the first part of this section. This is followed by a brief review of the existing network IDSs developed using artificial intelligence approaches.

A. TSK-Style Fuzzy Interpolation

Traditional TSK system generates a crisp inference result from a given input by calculating the weighted average of the sub-consequences of all fired rules [21]. Obviously, if a given input does not overlap with any rule antecedent, the firing strength of all rules will be 0, and consequently, no consequence can be derived. TSK-interpolation has addressed such issue [20]. Assume that a sparse TSK rule base is comprised of n rules as follows:

$$\begin{aligned}
&R_1: \textbf{IF } x_1 \text{ is } A_1^1 \text{ and } x_2 \text{ is } A_2^1 \text{ and } \cdots \text{ and } x_m \text{ is } A_m^1\\
&\quad \textbf{THEN } z = \mathbb{Z}_1,\\
&\quad \cdots\ \cdots\\
&R_i: \textbf{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \cdots \text{ and } x_m \text{ is } A_m^i\\
&\quad \textbf{THEN } z = \mathbb{Z}_i,\\
&\quad \cdots\ \cdots\\
&R_n: \textbf{IF } x_1 \text{ is } A_1^n \text{ and } x_2 \text{ is } A_2^n \text{ and } \cdots \text{ and } x_m \text{ is } A_m^n\\
&\quad \textbf{THEN } z = \mathbb{Z}_n,
\end{aligned} \tag{1}$$

where n represents the size of the rule base (i.e., the number of rules included in the rule base); $A_k^i, (i \in \{1, 2, \cdots, n\}$ and $k = \{1, 2, \cdots, m\})$ represents a normal and convex fuzzy set. For simplicity, only triangular membership functions are utilised in this work, and thus A_k^i is conveniently denoted as $(a_{k1}^i, a_{k2}^i, a_{k3}^i)$, where (a_{k1}^i, a_{k3}^i) is the support of the fuzzy set and a_{k2}^i is the normal point. Given an input $I = (A_1^*, A_2^*, \cdots, A_m^*)$ which may or may not overlap with any rule antecedents, a crisp output can always be generated by following the steps below.

Step 1: Determine the matching degrees $S(A_1^*, A_1^i)$, $S(A_2^*, A_2^i)$, $\cdots$, and $S(A_m^*, A_m^i)$ between the inputs $(A_1^*, A_2^*, \cdots, A_m^*)$ and rule antecedents $(A_1^i, A_2^i, \cdots, A_m^i)$ for each rule R_i $(i \in 1, 2, \cdots, n)$ by:

$$S(A_k^i, A_k^*) = \left(1 - \frac{\sum_{j=1}^{3} |a_{kj}^i - a_{kj}^*|}{3}\right) \cdot (DF) \ , \tag{2}$$

where DF, termed as $distance\ factor$, is a function of the distance between the two concerned fuzzy sets. DF is in turn defined as:

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

where h $(h > 0)$ is a sensitivity factor, and d represents the distance between the two fuzzy sets. Smaller value of h leads to a similarity degree which is more sensitive to the distance of two fuzzy sets, and vice versa.

Step 2: Calculate the firing degree of each rule by integrating the matching degrees of its antecedents and the given input values:

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

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

Step 3: Integrate the sub-consequences from all rules to get the final output:

$$z = \frac{\sum_{i=1}^{n} \alpha_i \cdot \mathbb{Z}_n}{\sum_{i=1}^{n} \alpha_i}. \tag{5}$$

B. Network Intrusion Detection

Data are sliced into a number of units during transmission over Ethernet. Each unit of data, named a packet, is formed by adding an extra header section on the top of the transmitted unit of data. Based on the packet headers, important features regarding the corresponding data transaction can be identified, such as the source and destination IP addresses, the total length of the formatted data unit, and the source/destination ports. From this, the traffic pattern or signature of each data transmission in the network environment can be identified by analysing the information included in the IP headers. These traffic patterns and signatures are commonly utilised by IDSs to identify potential threats in a network environment, thus to generate security alerts.

Soft computing algorithms have been widely employed for the development of IDSs [22], to enable an intelligent agent in the system that is capable of disclosing the latent patterns in abnormal and normal connection audit records, and to generalise the patterns to new connection instances of the same class. For instance, the use of artificial immune systems in intrusion detection aims to take the challenge of defending increasingly complex networks in a dynamic environment. IDSs have also been developed using artificial neural networks, each of which is composed of a number of neurones that are interconnected with each other with different weights. Genetic algorithm based IDSs utilise biological concepts of natural selection, that is, survival of the fittest. Fuzzy logic has been employed in either fuzzy inference based IDSs or fuzzified other soft computing approaches to handle uncertainty.

Amongst the existing fuzzy logic based IDSs, D-FRI-Snort [23] was developed using a dynamic fuzzy rule interpolation approach. This system provides an additional level of intelligence to Snort alongside with the liability of creating a dynamic rule base, which enhances the prediction capability of the original Snort. In particular, D-FRI-Snort starts with network traffic data collection and analysis using three features: the average packet time, the number of packets sent and received. Threat alerts are generated to indicate an additional threat level if advised by the existing rule base. The system also stores interpolated results and dynamically promotes new rules based on the collected interpolated rules to enhance the original Snort according to the current network conditions.

III. Intrusion Detection System

The proposed IDS is essentially a fuzzy inference system which comprises of an inference engine and a rule base. The inference engine has been discussed in Section II-A. The construction of the rule base and the adaptation of the inference engine for network intrusion detection are detailed below. For simplicity, in this work, the modelling is supposed to take place under a noise free network environment. Also, only normal and convex triangle fuzzy sets are employed in fuzzy modelling.

A. Feature Selection and Data Collection

A number of general features can be readily monitored by networking tools for networking analysis during data packet transmission over the network, but most of them are irrelevant to intrusion detection and some are redundant. Therefore, a well-thought feature selection process by experts often takes place for the task of network attack detection [24]. This widespread practice is also employed in this work. In particular, four important features are identified by experts as an IDS signature for the proposed system, which are listed in Table I.

TABLE I. Features for IDS

Feature	Description
Source bytes	The number of data bytes sent by source IP host
Destination bytes	The number of data bytes sent by destination IP host
Count	The number of connections to the same host as the current connection in the past 2 seconds
Dst_Host_Diff_Rate	The percentage of connections that ports are different within the past 100 connections with the sane destination IP.

Once the features are determined, data regarding a given network in certain environment need to be collected for model training. This is typically implemented in stages based on firstly an attack free network and then different types of attacks. That is, data regarding normal network traffic is collected first from a threat free condition network environment. Then, a number of attacks simulating the first type of attack are artificially launched such that this type of attack is sufficiently covered by the data set. This process is repeated for every other type of attacks such that all the classes that need to be considered are fully covered by the data set. Note that, as the classification of all attacks is a very challenging, if not impossible, task, only certain types of attacks are considered during the training data set collection process. However, the proposed IDS is able to deal with unknown or unconsidered types of attack with good success, as indicated in experimentation.

B. Rule Base Generation

Given a labelled training data set regarding a particular network and its environment, a TSK rule base can be generated to enable intrusion detection using a fuzzy inference system. The overall rule base generation process is illustrated in Fig. 1. Without losing generality, suppose that $l(l \geqslant 1, l \in \mathbb{N})$ classes are labelled in the given training data set, which covers $l-1$ types of attacks and the normal situation. The proposed system firstly divides the training data set T into l sub data sets $T_1, T_2, \cdots, T_l$, each representing a type of attack or the normal traffic (i.e., $T = \cup_{s=1}^{l} T_s$). Then, the K-Means, one of the most widely used clustering algorithm, is employed to each sub data set to group its data points into clusters based on their feature values. From this, each determined cluster is expressed as a fuzzy rule contributing to the constitution of the TSK rule base. Important sub-procedures in rule base generation are detailed below.

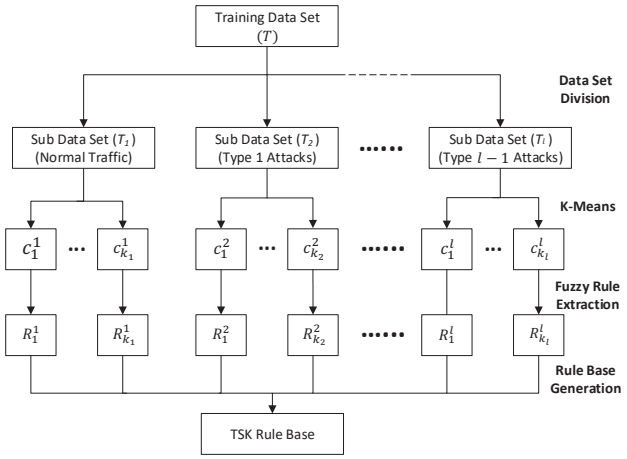

Fig. 1. Rule base generation

1) Clustering: The K-Means clustering algorithm is employed to each sub data set to group similar objects (data points in the sub data set) into clusters based on the Euclidean distance. Suppose that k_s clusters are needed for sub-data set T_s. The algorithm starts with the initialisation of k^s cluster centroids, which may be generated randomly or based on some strategies. Then each data object is allocated to the cluster whose centroid is the closest from the data object. After that, the algorithm updates the cluster centroids and reassign each data object to its closest cluster again. This process is reiterated until the centroids stabilised or the sum of squared error (SSE) is minimised. In particular, SSE is defined as follows:

$$SSE = \sum_{j=1}^{c_i^s} \sum_{i=1}^{k^s} (\parallel x_{ij}^s - v_i^s \parallel)^2 \ , \tag{6}$$

where x_{ij}^s is the j^{th} data point in i^{th} cluster in the sub data set T_s; v_i^s is the centre of the i^{th} cluster in the sub dataset T_s; c_i^s is the number of data points in i^{th} cluster of the subset T_s; and $\parallel x_{ij}^s - v_i^s \parallel$ is the Euclidean distance between x_{ij}^s and v_i^s.

Note that the value of k^s has to be pre-defined to enable the application of the K-Means algorithm. A number of approaches have been proposed to determine the value of k_s in the literature, such as the Elbow method [25]. This approach has been employed in this work, which determines the number of clusters based on the criteria that adding another cluster does not give much better modelling of the data set. It is noteworthy that different extensions, such as fuzzy C-means and K-medians, have been proposed to extend the K-Means algorithm. The application of these extensions may be of great importance for particular types of practical problems, but this is out of the scope of this paper.

2) Fuzzy Rule Extraction: TSK model takes each determined cluster of data to form one TSK fuzzy rule. Follow the last subsection and takes the sub data set T_s as an example. As four input attributes are considered in the proposed

IDS, the i^{th} cluster in sub dataset T_s can be represented as $(A_1^{si}, A_2^{si}, A_3^{si}, A_4^{si})$. The generation of the four fuzzy sets from c_i^s 4-dimensional data points is detailed below.

Given that only triangular membership functions are utilised in this work, the core of fuzzy set A_k^{si}, $k = \{1, 2, 3, 4\}$ is set as the cluster centre v_{ik}^s regarding variable x_k, that is $a_{k2}^{si} = (x_{ijk}^s + x_{ijk}^s + \cdots + x_{ijk}^s)/c_i^s$. The support of the fuzzy set $(a_{k1}^{si}, a_{k3}^{si})$ is expressed as the span of the cluster along this input dimension, where a_{k1}^{si} is the minimum value in this input dimension (i.e., $a_{ks}^{si} = \min(x_{ijk}^s, x_{ijk}^s, ...x_{ijk}^s)$), and similarly, a_{k3}^{si} is the maximum value in this input dimension (i.e., $a_{ks}^{si} = \max(x_{ijk}^s, x_{ijk}^s, ...x_{ijk}^s)$). To facilitate the discussion, the generation of (A_1^{si}, A_2^{si}) on two dimensions (x_1, x_2) for the i^{th} cluster of the s^{th} sub data set is illustrated in Figure 2 (as two dimension problems can be expressed in a plane for visualisation and better understanding).

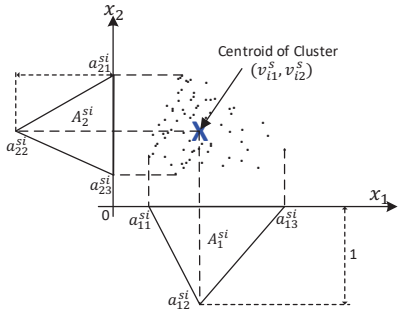

Fig. 2. Fuzzy representation of a 2-dimensional cluster

In this work, 0-order TSK rules are adopted. All data points in each cluster share the same class label of an integer number, which is utilised as the consequent of the corresponding TSK rule. The final TSK fuzzy rule base is generated by combining all the extracted rules from all l sub data sets (i.e., T_1, $T_2, \cdots, T_l$) as follows:

$$
\begin{aligned}
R_1^1 &: \textbf{IF } x_1 \text{ is } A_1^{11} \text{ and } \cdots \text{ and } x_4 \text{ is } A_4^{11} \\
&\quad \textbf{THEN } z = Z_1 \\
&\quad \textbf{...} \\
R_{k_1}^1 &: \textbf{IF } x_1 \text{ is } A_1^{1k_1} \text{ and } \cdots \text{ and } x_4 \text{ is } A_4^{1k_1} \\
&\quad \textbf{THEN } z = Z_n \ . \\
&\quad \textbf{...} \\
R_1^l &: \textbf{IF } x_1 \text{ is } A_1^{l1} \text{ and } \cdots \text{ and } x_4 \text{ is } A_4^{lk_l} \\
&\quad \textbf{THEN } z = Z_1 \\
&\quad \textbf{...} \\
R_{k_l}^l &: \textbf{IF } x_1 \text{ is } A_1^{lk_l} \text{ and } \cdots \text{ and } x_4 \text{ is } A_4^{lk_l} \\
&\quad \textbf{THEN } z = Z_n \ .
\end{aligned}
\tag{7}
$$

The number of rules in this rule base equals to the sum of the numbers of clusters for all the sub data sets (i.e., $k_1 + k_2 + ... + k_l$).

C. Intrusion Detection by TSK-Interpolation

The generated TSK rule bases are usually sparse thanks to the characteristics of network systems and its environment. Naturally, TSK-interpolation, as briefed in Section II-A, can be readily used to perform inferences in attack detection. In order to generate network intrusion alerts in real time, the system keeps capturing traffic data upon its deployment in a real network environment. The captured data in real time are then fed into the proposed system for system updating.

Assumes that the four features of current traffic for a given network are observed as $O = (x_1^*, x_2^*, x_3^*, x_4^*)$. From this input, the TSK-interpolation system firstly calculates the matching degrees between the given inputs (O) and the antecedents of individual rules utilising Equation 4. Then, the inference result of the TSK IDS model is produced from Equation 5. Finally, the numerical consequence values of the TSK-interpolation system is rounded to a whole number (symbolic value) which represents the category of the observed traffic.

Note that fuzzy rule interpolation has been successfully applied to the development of IDS in a real network environment [23], as discussed in Section II-B. This system generates accurate alerts and promotes new fuzzy rules to enhance the existing sparse rule base. Compared with the D-FRI-Snort system, which requires the involvement of experts for fuzzy variable determination and fuzzy rule base initialisation, the proposed system only requires network experts in the generation of training data, which implies better applicability.

IV. Experimentation

In order to validate and evaluate the proposed system, it is applied to a well-known data set, the KDD Cup 99 data set, and thereby to demonstrate its efficacy.

A. The Data Set

The KDD Cup 99 data set is a popular benchmark in the research field of Intrusion Detection, which includes legitimate connections and a wide variety of intrusions simulated in a military network environment [26]. This data set contains almost 5 millions of data instances with 42 attributes, including the 'class' attribute which indicates whether a given instance is a normal connection instance or one of the four types of attacks to be identified (i.e., Normal, Denial of Service Attacks, User to Root Attacks, Remote to User Attacks, and Probes). Knowing the inherent issues associated with the data set, such as the high duplication rate of 78%, the KDD Cup 99 data set has been further processed to NSL-KDD-99 [26]. This processed data set includes 125,937 data samples with all the features of the original data set kept. Thanks to its convenience, this data set has been utilised in a number pieces of recent research, such as [27], which is also employed for system validation and evaluation.

B. Model Construction

In this work, 80% of the data samples randomly chosen from the NSL-KDD-99 data set are used for system training, and the rest 20% for testing. Note that there are five possible labels in the training data set, including Normal, Denial of

Service Attacks (DoS), User to Root Attacks (R2R), Remote to User Attacks (R2U), and Probes. Five integer numbers are utilised as the 0-order consequences of TSK rules to represent the five possible consequences. The five categories and their corresponding rule consequences, the number of data instances for each category in the training and testing data sets are listed in the Table II.

TABLE II. INFORMATION FOR CATEGORIES

Category	No. of Training Instances	No. of Testing Instances	TSK rule Consequence
Normal Traffic	67,343	9711	1
DoS	45,927	5741	2
U2R	52	37	3
R2U	995	2199	4
Probes	11,656	1106	5

A TSK-style rule base was generated from the NSL-KDD training data set using the approach proposed in Section III-B. The key steps are summarised below.

[Step 1] *Training data set preprocessing:* Four important features have been identified in Section III-A for network intrusion detection. Therefore, only these four features (which corresponds to the 5^{th}, 6^{th}, 23^{rd}, and 35^{th} features in the NSL-KDD-99 data set) were kept and all other feature values were removed from the training data set.

[Step 2] *Data set division:* The training data set was divided into five sub data sets such that all the data instances in each sub data set were of the same 'class' attribute value.

[Step 3] *Clustering:* The standard K-Means clustering algorithm was employed to partition the given training data samples into multiple clusters, and the Elbow method was used to determine the number of clusters in this work. Taking the sub data set for DoS as an example, the SSE values for different cluster numbers were calculated as shown in Figure 3. It is clear from this figure that the elbow point is 4, which was taken as the value of k when applying K-Means. The cluster numbers for other sub data sets were identified in the same way, with the results listed in Table III.

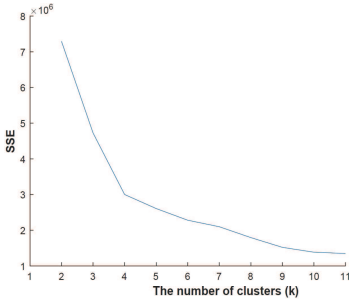

Fig. 3. Identification of the number of clusters using the Elbow method

TABLE III. CLUSTER NUMBERS FOR EACH SUB DATA SET

Type of Data Set	Corresponding Consequent	Determined Number of K
Normal Traffic	1	21
DoS	2	4
U2R	3	3
R2U	4	6
Probes	5	6

[Step 4] *Rule base generation:* Each cluster was represented as a fuzzy rule. Therefore, 40 rules in total were generated forming the final rule base in the format of Equation 7. For instance, the third extracted rule from sub data set of DoS can be expressed as:

$$R_3: \textbf{IF } x_1 \text{ is } (520, 1032, 1480) \text{ and } x_2 \text{ is } (0,0,0) \text{ and } x_3 \text{ is } (158, 486, 511) \text{ and } x_4 \text{ is } (0, 0.02, 0.2) \textbf{ THEN } z = 2\ . \quad (8)$$

C. Experimental Results

Once the rule base is generated, the model is then applied to the testing data set with the result shown in Table IV. This table also includes the results generated from several existing well-known methods [28] for comparative study, including the decision tree, Naive Bayes, and back-propagation neural network (BPNN). The proposed approach, in general, outperforms other approaches. In particular, the approach performs the best for the classification of three classes, including Normal Traffic, R2U and Probes. The accuracy rate for DoS from the proposed approach is worse than those from all other approaches by a marginal difference. The classified result for the class of U2R from the proposed approach is better than that from Naive Bayes, but worse than those from decision tree and BPNN.

TABLE IV. RESULTS COMPARISON

	Decision Tree (%)	Naive Bayes (%)	BPNN (%)	Proposed System (%)
Normal Traffic	91.22	89.22	89.75	96.81
DoS	99.84	99.69	99.79	98.15
U2R	50.00	25.00	50.00	45.71
R2U	33.33	46.15	57.14	75.99
Probes	50.00	52.61	60.94	74.11

The proposed system has a good generalisation ability. In order to evaluate the ability of the proposed system in handling uncovered types of threats in the training data set, another pre-processed sub data set is adopted for testing purposes. In particular, this data set contains 3,220 data instances extracted from original KDD Cup 99 data set. This testing data set is also provided by [26], and the types of threats in this testing data set are not covered by NSL-KDD (which has been used to train the TSK model). The accuracy of detection of unknown types of treats is shown in Table V with a comparison with the result led by naive Bayes. This comparative study demonstrates that the proposed system is able to generate a much better result.

TABLE V. RESULT OF UNKNOWN INTRUSIONS

Accuracy by Proposed System (%)	Accuracy by Naive Bayes (%)
74.41	55.77

D. Analysis and Discussion

The experimental results show the potential of the proposed approach, in that it generally outperforms other approaches regarding the classification of the three classes. It is noteworthy that the poor performance on the prediction of class U2R is led by the very small sub data set, given that there are only 52 data instances are associated with this class. The imbalance of data set significantly limits the performance of the inference

model. Better data capture strategies are therefore appealing in the real world situation to avoid this situation.

Although many fuzzy inference systems have been employed to detect attacks form unlimited network connections, they all require a dense fuzzy rule base to enable the performance of fuzzy inferences. However, a dense rule base covering the unlimited network connections is practically hard, if not impossible, to be obtained, which limits the effectiveness of these approaches. Also, IDSs developed from conventional fuzzy inference systems with sparse knowledge bases may result in unexpected system outputs. Thanks to the relaxation on the requirements of rule bases from TSK-interpolation, reasonable intrusion alerts can still be generated when a given observation is not overlapped with any rule antecedents, which significantly improves the applicability of fuzzy inference systems in the field of network intrusion detection.

V. Conclusions

This paper presented a data-driven network intrusion detection system, by employing the recently proposed TSK-interpolation approach. The experiment results using the benchmark data set KDD-99 demonstrated that the proposed system is not only able to successfully generate security alerts for known attack types, but also to detect the unknown types of treats with good success thanks to its good generalisation ability. This work can be enhanced by employing the recently proposed rule base generation approach [29] to generate a sparse rule base directly from very complex training data sets, and rule base adaptation approach [30] to allow the rule base to be adapted and enhanced along with the operation of the IDS system. Also, the proposed work is developed using TSK-interpolation, it is worthwhile to investigate how the proposed system may be developed by employing other fuzzy interpolation approaches with Mamdani-style rules bases. In addition, as the DSS Cup 99 is an old data set, real-world applications are desired for more system evaluation and validation.

References

[1] K. Hwang, M. Cai, Y. Chen, and M. Qin. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. *IEEE Transactions on Dependable and Secure Computing*, 4(1):41–55, 2007.

[2] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):516–524, Sept 2010.

[3] L. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.

[4] L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1):169–201, 1993.

[5] Z. Huang and Q. Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.

[6] L. Yang and Q. Shen. Adaptive fuzzy interpolation and extrapolation with multiple-antecedent rules. In *International Conference on Fuzzy Systems*, pages 1–8, 2010.

[7] 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.

[8] L.Yang and Q. Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, Dec 2011.

[9] Q. Shen and L. Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298, 2011.

[10] L.Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. Theme: Fuzzy Systems.

[11] 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.

[12] 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.

[13] 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.

[14] S. Cheng, S. Chen, and C. Chen. Fuzzy interpolative reasoning based on ranking values of polygonal fuzzy sets and automatically generated weights of fuzzy rules. *Information Sciences*, 325:521 – 540, 2015.

[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] S. Chen and W. Hsin. Weighted fuzzy interpolative reasoning based on the slopes of fuzzy sets and particle swarm optimization techniques. *Cybernetics, IEEE Transactions on*, 45(7):1250–1261, 2015.

[17] S. Chen and Z. Chen. Weighted fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on piecewise fuzzy entropies of fuzzy sets. *Information Sciences*, 329:503 – 523, 2016.

[18] L. Yang, F. Chao, and Q. Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, (DOI: 10.1109/TFUZZ.2016.2582526), 2016.

[19] 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.

[20] 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.

[21] 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.

[22] M. Patel, A.and Taghavi, K. Bakhtiyari, and J. C. JúNior. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of network and computer applications*, 36(1):25–41, 2013.

[23] N. Naik, R. Diao, and Q. Shen. Application of dynamic fuzzy rule interpolation for intrusion detection: D-fri-snort. In *2016 IEEE International Conference on Fuzzy Systems*, pages 78–85, 2016.

[24] S. Guha, S. S. Yau, and A. B. Buduru. Attack detection in cloud infrastructures using artificial neural network with genetic feature selection. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing*, pages 414–419, Aug 2016.

[25] R. L. Thorndike. Who belongs in the family. *Psychometrika*, pages 267–276, 1953.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *The Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009.

[27] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303–336, 2014.

[28] G. Wang, J. Hao, J. Ma, and L. Huang. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, 37(9):6225 – 6232, 2010.

[29] 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, 2016.

[30] J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 102–109, July 2016.