MATEC Web of Conferences **44**, 02041 (2016)
DOI: 10.1051/matecconf/20164402041

A graphical feature generation approach for intrusion detection

Shi Chen [1,a], Zhen Zuo [1] , Zhi Ping Huang[1] and Xiao Jun Guo [1]

[1]*Department of Instrumentation Science and Technology, National University of Defense Technology, 410073 Changsha, P.R.China*

Abstract. In order to develop a novel effective and efficient intrusion detection system, a novel hybrid method based on a graphical features-based k-nearest neighbor approach, namely GFNN, is proposed in this paper. In GFNN, k-means clustering algorithm is used to extract cluster centre of each class in the given dataset. Then, the distance between a specific data sample and each cluster centre is calculated, and a radar chart is plotted based on the new data composed of distance based features. The sub-barycentre based features for each sample are extracted from the radar chart. As a result, our proposed approach transforms the original multi-dimensional feature space into 5-dimensional sub-barycentre feature space. The experimental results of 10-fold cross-validation based on the KDDcup99 dataset show that the GFNN not only performs better than or similar to several other approaches in terms of classification accuracy, precision, and recall. It also provides high computational efficiency for the time of classifier training and testing.
Key Words: Intrusion detection; Machine learning; Graphical feature; Radar chart; k-means; k-nearest neighbors

1 Introduction

With the popularity of computer network, network security problem becomes more and more important. Although many security techniques, such as user authentication, data encryption, and firewalls, have been used to improve the security of networks, there are still many unsolved problems, which has made many researchers focus on building systems called intrusion detection systems (IDSs) [1]. Generally, IDSs can be classified into two categories: misuse detection and anomaly detection. In misuse detection, the system identifies intrusions based on known intrusion techniques and triggers alarms by detecting known exploits or attacks based on their attack signatures. Misuse detection can discover attacks with a low false positive rate, but it cannot discover novel attacks without known signatures. On the other hand, in anomaly detection, the system discovers attacks by identifying deviations from normal network activities, it can discover novel attacks but with a high false positive rate.

In order to enhance detection precision and detection stability, machine learning has been widely used to improve IDSs, such as fuzzy theory [2, 3], k-nearest neighbor (k-NN)[4, 5], support vector machine (SVM) [6, 7], artificial neural networks (ANN) [8], Naïve Bayes networks [9], decision tree [10, 11], genetic algorithm (GA) [2, 12], self-organizing maps (SOM) [13], and Markov chains [14, 15], etc. Based on the number of learning techniques used, those IDSs can be categorized as two types: IDSs using single learning techniques and IDSs with hybrid learning techniques. Prior researches have demonstrated that an ensemble of several different techniques performs better than each technique individually [2, 6, 16, 17]. However, those hybrid techniques should upgrade the computational complexity and make it hard to deal with large datasets. For classification problem with high dimensional data, feature reduction techniques are always an effective method to delete the redundancy and irrelevancy features, decrease the computation time, and finally improve the IDS system performance. In this line of research, some feature reduction techniques have been applied to improve the performance of IDSs, such as GA [12], Entropy, principal component analysis (PCA) [18], etc.

In this paper, we try to develop a novel effective and efficient intrusion detection system. The main goal of this paper is aiming to achieve feature reduction, data presentation, feature generation, and to enhance IDS with hybrid method combing k-means clustering algorithm, graphical features generation approach and k-NN classifier. Firstly, k-means clustering algorithm is used to extract cluster center of each pre-defined category in the given dataset, which contains n categories. Secondly, the distance between a specific data sample and each cluster center is calculated, and then a radar chart is applied to represent each new data sample composed of distance based features. Thirdly, new graphical features (e.g., barycenter) for each sample are extracted from the radar chart. Finally, for classification, some classifiers are used to detect attacks based on the new dataset described by graphical features to obtain the best performance.

The rest of this paper is organized as follow. Some related work on IDSs, including the use of feature reduction and representation methods, is reviewed in

[a] Corresponding author: chens_sir@163.com

Section 2. Section 3 presents a detailed description of the proposed framework, including each step of the proposed approach. The dataset, evaluation strategies, and results of a performance comparison are presented in Section 4. Finally, some conclusions are provided in Section 5.

2 Related work

Feature reduction is an important component in machine learning, especially when dealing with high dimensional data, because the use of correct features improves classification performance and reduces computational time. In the past decades, several IDSs, which use feature reduction as a preprocessing phase, have been developed.

Mukkamala and Sung [19] proposed a novel IDS based on the SVM, multivariate adaptive splines and linear genetic program. They used a new significant algorithm, which is independent of the modeling tools being applied, to select features. After an input feature being removed from the data, the remaining dataset is then used for training and testing the classifier. Then, the classifier's performance is compared to that of the original classifier in terms of relevant performance criteria. Finally, the features are ranked according to a set of rules based on the performance comparison. The performance of two feature reduction techniques, principal component analysis and linear discriminate analysis, is compared by Datti and Lakhina [20], and the back-propagation algorithm is used to test these techniques. Eesa, Orman and Brifcani [11] proposed a feature selection model based on the cuttlefish optimization algorithm (CFA) to produce the optimal subset of features. They used the decision tree as the classifier to improve the quality of the produced subsets of features. Tsai and Lin [21] proposed a hybrid approach combined k-means and k-NN to construct a triangle area-based nearest neighbor approach (TANN) for intrusion detection. In TANN, 10 triangle areas formed by the data and five class cluster centers are generated to replace the 41 original features of KDDcup99 dataset, and the approach achieved good performance. The dimensionality of data sample changed from 41 to 10. However, the dimensionality of the new features generated by TANN is based on the number of triangle areas, which is equal to $k \times (k-1)/2$ (k denotes the number of classes in the given dataset). If the classification problem contains a larger number of classes, the dimensionality of the new features will become very large. This may lead to the "curse of dimensionality" problem. To solve the problem in TANN, two improvement approaches are proposed. One is the distance sum-based hybrid method combining k-means and SVM, called DSSVM [22]. In DSSVM, the sum of distances between the data sample and k–1 of the k cluster centers is calculated to replace the original features. Then, the new k-dimensional dataset is obtained. Another approach is CANN [23] that proposed to reduce the dimensionality of features. In CANN, two distances are measured and summed, the first one based on the distance between each data sample and its cluster center, and the other is between the data and its nearest neighbor in the same cluster. As a result, a new one-dimensional distance based feature is used to represent each data sample for intrusion detection by a k-NN classifier.

With the development of network and storage capacity, multi-dimensional data representation and analysis tasks are becoming increasingly difficult and challenging. Under such circumstance, some methods of graphical representation of the multi-dimensional data have been developed in many fields. Wang and Hong [24] use a star plot, which is also called radar chart, to represent a multi-dimensional data for machine learning. In the star plot, each original feature of a data sample is represented by a line segment radiating from the central point of star plot. Then, new graphical features, such as sub-area features and sub-barycenter features, are extracted from the star plot. At last, the new dataset represented by graphical features is utilized for training and testing of k-NN classifier. For resolving a gene classification problem, Parkinson and Blaxter [25] developed a new visualization model, which is called SimiTri. By mapping a certain gene into a triangular phase space, whose three vertexes represent three selected distinct gene categories, the position of the gene within the phase space indicates its relationship to each of the three selected data sets and helps determine the category of the data. This SimiTri model can resolve 4-class classification problem. Enlightened by TANN [21] and SimiTri [25], Luo and Xia [26] proposed a novel four-angle-star based visualized feature generation approach, FASVFG, to evaluate the distance between samples in a five class classification problem for intrusion detection. In FASVFG, a four angle star is constructed by a square and four triangles. The distance between a data sample and the vertexes of four-angle-star helps determine the class the data belonging to.

Inspired by the applications of graphical representation approaches above, the radar chart, originally applied to analyze the financial situation of enterprise, is used to general new graphical features for intrusion detection in this paper. As a result, a hybrid approach is proposed to transform the original multi-dimensional feature space into low-dimensional graphical feature space. Then, the graphical features are regarded as the new feature space and used for the final classification decision.

3 The graphical features-based k-nearest neighbor (GFNN)

The proposed GFNN consists of three stages, which are cluster centers extraction, new data formation by graphical features, and training and testing k-NN based on the new data. From methods [21, 23] we learn that the distance between the data sample and each clustering center has high discriminative power, and may lead to classification algorithm (e.g., k-NN and SVM) achieving similar or better classification accuracy than the original high-dimensional feature vectors. Moreover, the graphical feature of radar chart generalization approach is a visualization strategy that enables people to identify the class to which the data belongs [24]. The experiment

results of approach in [24] show that its proposed graphical features can help achieve high classification accuracy. Specifically, GFNN intends to provide effective and efficient detection of network attacks using k-NN algorithm with low-dimensional feature vectors.

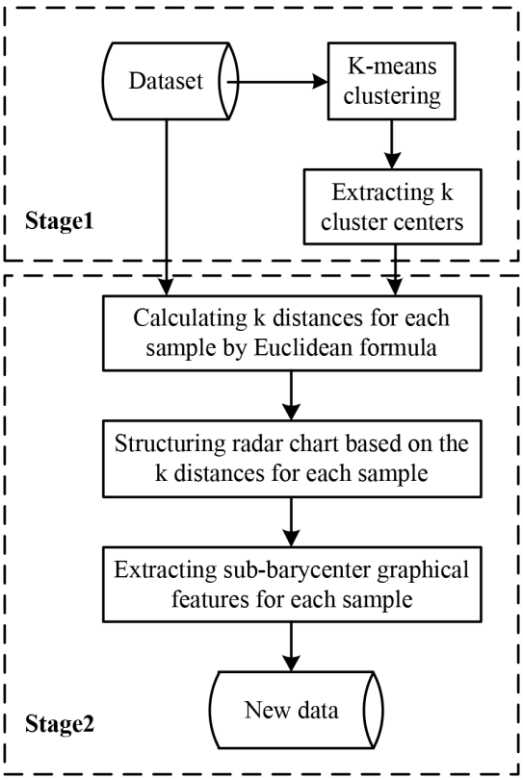

Figure 1. Process of new data formation.

3.1 Cluster centers extraction

In this paper, the KDDcup99 dataset is used, which includes five classes (four types of Internet attacks and one types of normal access). The dataset is described in detail in section 4.1. The k-means clustering algorithm is used to extract cluster centers for its simplicity and low time complexity. For five-class classification problem, the value of k is set by 5. Therefore, five cluster centers are obtained, and each cluster contain a cluster center (i.e., C_1, C_2, C_3, C_4 and C_5). The stage 1 of Figure 1 shows the process of cluster centers extraction.

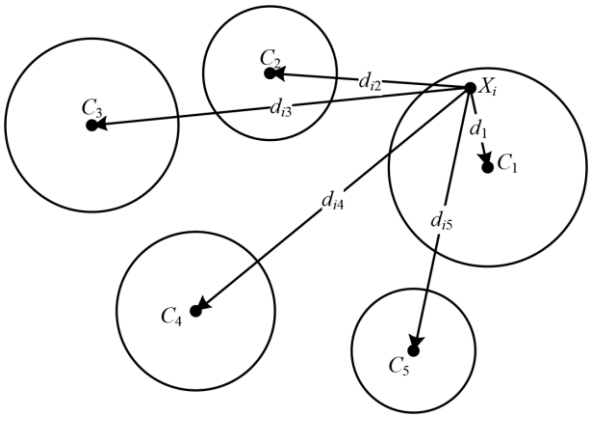

Figure 2. Distance calculation of a data sample.

3.2 New data formation by graphical features

In this stage (see stage 2 of Figure 1), first of all, the distance $d_{ij}(j=1,2,\cdots,5)$ between each n-dimensional data sample $X_i=(x_1,x_2,\cdots,x_n)$ and the five cluster centers $C_1,C_2,\cdots,C_5$ is calculated respectively based on the Euclidean distance formula. The Euclidean distance between $X_i=(x_1,x_2,\cdots,x_n)$ and each cluster center point $C_j=(c_1,c_2,\cdots,c_n)$ (see Figure 2) can be defined as

$$d_{ij}=\sqrt{\sum_{t=1}^{n}(x_t-c_t)^2} \tag{1}$$

Where $i=1,2,\cdots,m$ (m denotes the total number of the data samples), $j=1,2,\cdots,5$. Then, we can obtain five distance features for each sample, that is $D_i=(d_{i1},d_{i2},\cdots,d_{i5})$. This step transforms the n-dimensional original features vectors into the 5-dimensional distance features vectors.

To construct a radar chart, all distance data are normalized into $[a,1]$, by using the following formula:

$$d'_{ij}=a+(1-a)\frac{d_{ij}-\min_{1\le i\le m}(d_{ij})}{\max_{1\le i\le m}(d_{ij})-\min_{1\le i\le m}(d_{ij})}, \quad i=1,2,3,\cdots,m;\ j=1,2,\cdots,5. \tag{2}$$

Where a is the desired length of the smallest ray relative to the largest, m is the total number of the data samples, $\min_{1\le i\le m}(d_{ij})$ and $\max_{1\le i\le m}(d_{ij})$ are the minimum and maximum value of the j-th feature, respectively.

After obtaining five normalize distance features, radar chart (see Figure 3) can be plotted to represent each sample data $D'_i=(d'_{i1},d'_{i2},\cdots,d'_{i5})$.

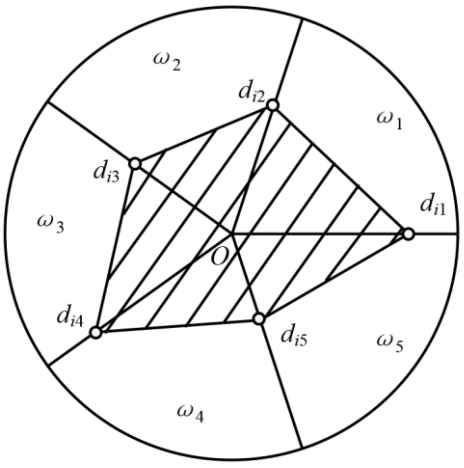

Figure 3. Radar chart of a data sample.

As shown in Figure 3, the radius of the circle is 1. Five rays, that divide the perimeter of the circle into five equal parts, represent five distance features, respectively. We can obtain a point on each ray, and the distance between the point and the center of the circle is equal to each distance data d'_{ij}. After connecting each two adjacent

points with lines, a pentagon is obtained, which is composed of five triangles. The angle formed by two adjacent rays can be calculated by $\omega_j = 2\pi / k, k = 5$ for KDDcup99 dataset.

There are several kinds of graphical features that can be extracted from the radar chart, such as area, barycenter, and perimeter of each triangle or pentagon. The work in [24] show that the performance of k-NN with sub-barycenter (the barycenter of triangle) features of radar chart is superior to that with sub-area features. Therefore, in this paper, the sub-barycenter features of radar chart are selected to form the training and testing dataset.

The sub-barycenter features of radar chart are extracted as the following. Each triangle (see Figure 4) in the pentagon has a barycenter $G_{ij} = (l_{ij}, \alpha_{ij})$, where l_{ij} is the distance between the barycenter of triangle and the center of radar chart, α_{ij} is the angle formed by $G_{ij}O$ (O donates the center of radar char) line and horizontal line.

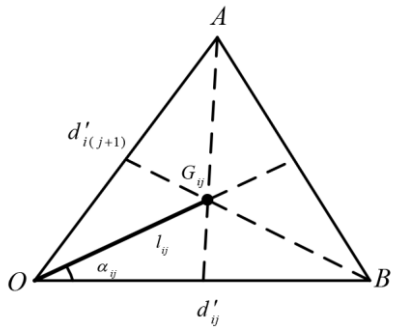

Figure 4. Barycenter of a triangle.

For simplification and dimensionality reduction, we only choose l_{ij} as the graphical features that can be obtained by

$$l_{ij} = \begin{cases} \sqrt{(\frac{d'_{ij}}{3}\sin\omega_j)^2 + [(d'_{ij}\cos\omega_j - \frac{d'_{i(j+1)}}{2})/3 + \frac{d'_{i(j+1)}}{2}]^2}, & j = 1,2,3,4 \\ \sqrt{(\frac{d'_{ij}}{3}\sin\omega_j)^2 + [(d'_{ij}\cos\omega_j - \frac{d'_{i1}}{2})/3 + \frac{d'_{i1}}{2}]^2}, & j = 5 \end{cases} \quad (3)$$

As a result, the n-dimensional original feature vectors are transformed into new five-dimensional sub-barycenter feature vectors. Finally, the new dataset D_L is used for training and testing the k-NN classifier.

3.3 Training and testing classifiers

In this stage, the new dataset $D_L = \{L_1, L_2, \cdots, L_m\}$, represented by five-dimensional feature vectors, is used to train and test k-NN classifier. Several classifiers, such as Naïve Bayes (NB), Decision Tree (J48), Random Tree (RT), and SVM, are utilized to take performance comparison with k-NN. Figure 5 shows the process of training and testing a classifier. In this stage, 10-fold cross validation is used to avoid the variability of the samples that affect the performance of model training and testing. In 10-fold cross validation, the whole dataset is divided into 10 unduplicated subsets. Nine of the 10 subsets are used as training subset and the remainder is used for testing. Then 10 classification results are obtained by 10-fold cross validation. Finally, classification accuracy is the average value of the 10 classification results.

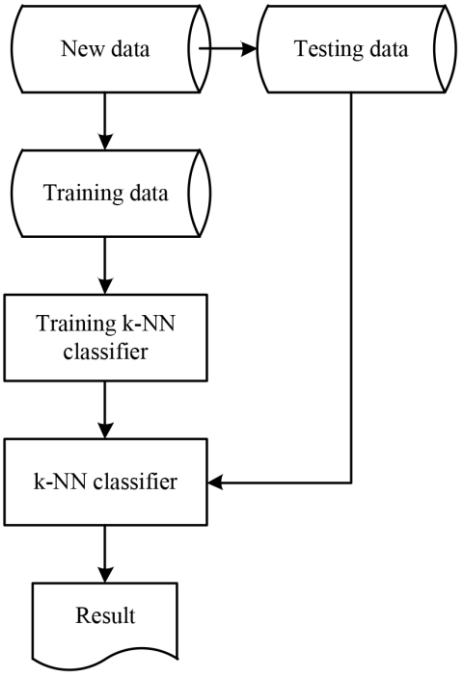

Figure 5. Training and testing k-NN with new data.

4 Experiments

4.1 The dataset

In this section, we discuss the experimental dataset: KDDcup99 dataset. KDDcup99 dataset was originally derived from the raw DARPA network traffic, which was prepared and managed by MIT Lincoln Labs in 1998. KDDcup99 dataset has become a benchmark dataset for intrusion detection. The complete dataset has almost 5 million input patterns and each record represents a TCP/IP connection. Considering the sake of simplicity, the dataset used in our study is the 10% subset, namely kddcup.data_10_percent, which contains 494,021 instances.Except normal class, the recordings in KDDcup99 dataset can be divided into four attack classes as follows:

- Denial of service (DOS): an attacker results by preventing legitimate requests to a network resource by consuming the bandwidth or by overloading computational resources, e.g., smurf;
- Probe: an attacker scans a network of computers and collect information of target system, e.g., port scanning;
- User to Root (U2R): unauthorized access to local super user privileges by a local unprivileged user, e.g., various “buffer overflow” attacks;

- Remote to Local (R2L): unauthorized access from a remote machine to a local machine, e.g., guessing password.

4.2 Dataset preprocessing

The 10% KDDcup99 dataset has a huge number of redundant records, that almost 70.5% of the records are duplicated, which cause the learning algorithm to be biased towards the most frequent records. For these duplicates, we just delete them. Then the size of the dataset changes to 145,585.

After the deletion, the dataset still have 87,832 records of class Normal and 54,572 records of class Dos. In order to reduce the experimental time, we randomly select 10% of class Normal and Dos as the final experimental Normal and Dos records, while the other three classes remain unchanged. As a result, the size of the final experimental dataset is reduced from 145,585 to 17,421.The final size of each class is shown in Table 1.

In the KDDcup99 dataset, there are several nominal features. Before extracting cluster centers based on k-means, the nominal features are mapped into binary numeric features. Note that "service type" has 67 different values, which would heavily increase the dimensionality if mapped into binary features. Specially, for comparing with DSSVM [22], the "service type" feature is not used in this paper. After mapping nominal features into binary values, 52 numeric features are constructed.

Table 1. Final size of each class.

Class	Normal	Dos	Probe	U2R	R2L
The final dataset	8,783	5,457	2,130	52	999

Because the scales of some numerical features in the KDDcup99 dataset are different, for instance, "logged in" has binary value, whereas "source bytes" has a range from 0 to 693,375,640, normalization is required that it can avoid attributes with greater values dominating these attributes with smaller values. In this paper, the numerical attributes are scaled into the interval [0, 1] by dividing every attribute value by its own maximum value.

4.3 Evaluation criteria

We choose the confusion matrix as the performance measures to estimate the efficiency of classifier. The following measures are derived from the confusion matrix, and will be used for evaluating the proposed scheme:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

$$Pricision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where *TP*, true positive is the number of attack test samples correctly classified as attack. *TN*, true negative is the number of normal test samples correctly classified as normal. *FP*, false positive is the number of normal test samples falsely classified as attack. *FN*, false negative is the number of attack test samples falsely classified as normal.

Moreover, we use the area under the Receiver Operating Characteristic curve (AUC) to measure the performance of the proposed method for IDS. Usually, the AUC value ranges from 0.5 to 1.0, with larger AUC values representing better performance of the classifier.

4.4 Experimental results and analysis

The experiments are carried on an Intel Core 2.90 GHz computer with 4 GB RAM running Windows 7. We use a data mining software Weka to build classifiers in the experiment. Weka is a collection of machine learning programs for data mining tasks written in Java. The version used is Weka 3.7.

4.4.1 Classification results

The performances comparison of intrusion detection model based on NB, J48, RT, SVM and k-NN with the new dataset is given in Table 2. The comparative results allow us to see the performance of each classifier and identify the best one for intrusion detection.

Table 2. Performances of NB, J48, RT, SVM, and k-NN with the new graphical features.

	Normal (%)	Dos (%)	Probe (%)	U2R (%)	R2L (%)	$Accuracy_{total}$ (%)	For the normal and attack classes		AUC_{avg}
							Precision (%)	Recall (%)	
NB	88.45	77.66	81.69	0.00	32.63	80.78	88.39	89.40	0.965
J48	**98.67**	98.55	96.38	30.77	93.59	97.86	98.60	97.90	**0.991**
RT	98.43	98.81	96.62	36.54	93.29	97.85	98.40	98.19	0.984
SVM	97.59	95.58	91.83	0.00	92.79	95.69	97.49	95.15	0.965
k-NN	98.63	**99.38**	**98.26**	**57.69**	**95.80**	**98.54**	**98.62**	**98.96**	0.990

Table 3. Performances of TANN, DSSVM and GFNN over the 52-dimentional pre-processing dataset.

	Normal (%)	Dos (%)	Probe (%)	U2R (%)	R2L (%)	$Accuracy_{total}$ (%)	For the normal and attack classes		AUC_{avg}
							Precision (%)	Recall (%)	
TANN(k=1)	98.75	99.24	**98.31**	55.77	95.60	**98.542**	98.73	98.85	0.990
DSSVM	**99.50**	98.09	96.34	25.00	92.29	98.037	**99.48**	96.96	0.982
GFNN(k=1)	98.63	**99.38**	98.26	**57.69**	**95.80**	98.536	98.62	**98.96**	**0.990**

The result in Table 2 indicate that, with the new graphical features, the accuracy of k-NN (k=1) is 98.54% which outperforms that of NB (80.78%), J48 (97.86%), RT (98.40%), and SVM (97.49%). In addition, for the normal and attack classes, the graphical features-based k-NN (GFNN) also provides the higher precision rate (98.62%) and recall rate (98.96%) than NB, J48, RT, and SVM. Moreover, the accuracy for class normal and the average AUC value of GFNN is only slightly lower than that of J48. As shown in Table 2, the class U2r is not well detected by any of the classifiers, since the KDD dataset is rather unbalanced. In fact, fifty-two samples of U2R represent only 0.003% of the training samples.

For evaluating the performance of GFNN, two recently proposed approaches are reproduced to compare with GFNN. These approaches are trained and tested over the 52-dimentional preprocessing dataset. One approach is distance sum-based support vector machine (DSSVM) [22]. We examine several kernel function (such as Polynomial kernel, RBF kernel) and different degree in order to obtain the best performance for comparison. Another is triangle area-based nearest neighbors (TANN) [21], in which different k values are examined for getting the best k-NN classifier for comparison.

Table 3 shows the performances of these approaches over the 52-dimentional preprocessing dataset. As illustrated in Table 3, the total classification accuracy of GFNN is 98.536%, which is slightly lower than that of TANN (98.542%) and higher than that of DSSVM (98.037%). In addition, the GFNN also works the best for class DOS, U2R and R2L among the three classifiers with an accuracy of 99.38%, 57.69% and 95.80%. Although GFNN does not perform the best for class normal and probe, accuracy and detection rates of GFNN is very similar to the ones of TANN. This indicates that there is no significant difference in their performances.

4.4.2 Efficiency evaluation

Besides the quality of the classification results obtained by GFNN, it is important to consider the efficiency of the method. We evaluate the CPU time of GFNN by comparing against the baseline k-NN, TANN, and DSSVM. We divide CPU time into feature generating, training and testing. The feature generating for TANN, DSSVM and GFNN contain three parts: (i) dataset preprocessing; (ii) cluster centers extraction with k-means; (iii) dataset transformation.

Table 4 shows a comparison of the CPU time taken by each approach. Looking at Table 4, although k-NN, with 41 original features, doesn't need to take CPU time to do feature generating, it requires greatest training and testing time. Meanwhile, the training and testing time and the total CPU time taken by GFNN is the least among these approaches. Overall, GFNN's new low-dimensional feature vectors composed of the sub-barycenter can provide relatively high efficiency for intrusion detection.

Table 4. CPU time of k-NN, TANN, DSSVM, and GFNN.

Method	Feature generating (s)	Training and testing (s)	Total (s)
k-NN (41 features)	-	51	51
TANN (10 features)	9	17	26
DSSVM (5 features)	8	46	54
GFNN (5 features)	10	11	21

5. Conclusion

In this paper, we propose a novel hybrid method based on a graphical features-based k-nearest neighbors approach, namely GFNN, for intrusion detection. The proposed approach tests intrusion detection over the KDDcup99 dataset. Firstly, the k-means clustering algorithm and radar chart are used as feature generation tools. Then, this new and 5-dimensional sub-barycenter based feature is used to represent each data sample for intrusion detection by a k-NN classifier. From simulation results, the performance of the proposed algorithm outperforms other existing approaches for class DOS, U2R and R2L. It also provides high computational efficiency for the time of classifier training and testing.

In future work, some issues will be considered. Firstly, we plan to extract other graphical features from radar chart to construct low-dimensional feature space. Secondly, different clustering algorithms and classification techniques can be applied during the cluster center extraction and classifier training stages, respectively. Finally, our proposed feature generation methods will be used as a preprocessing step over other

datasets which contain different numbers of classes and also do a thorough comparison of our proposed feature generation methods with other ones.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61374008).

References

1. H.J. Liao, C.H.R. Lin, Y.C. Lin, K.Y. Tung, Intrusion detection system: A comprehensive review, JOURNAL OF NETWORK AND COMPUTER APPLICATIONS, **36**,1:16-24 (2013)
2. M.S. Abadeh, J. Habibi, C. Lucas, Intrusion detection using a fuzzy genetics-based learning algorithm, Journal of Network and Computer Applications, **30**,1:414-428 (2007)
3. A. Tajbakhsh, M. Rahmati, A. Mirzaei, Intrusion detection using fuzzy association rules, Applied Soft Computing Journal, **9**,2:462-469 (2009)
4. Y.H. Liao, V.R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, COMPUTERS & SECURITY, **21**,5:439-448 (2002)
5. M.-Y. Su, Real-time anomaly detection systems for denial-of-service attacks by weighted k-nearest-neighbor classifiers, Expert Systems With Applications, **38**,4:3492-3498 (2011)
6. S. Mukkamala, A.H. Sung, A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, Journal of Network and Computer Applications, **28**,2:167-182 (2005)
7. S. Mukkamala, A.H. Sung, Feature ranking and selection for intrusion detection systems using support vector machines. *In International conference on information and knowledge engineering (ICIKE)*. 503–509 (2002)
8. D. Fisch, A. Hofmann, B. Sick, On the versatility of radial basis function neural networks: A case study in the field of intrusion detection, Information Sciences, **180**,12:2421-2439 (2010)
9. Z.A. Baig, S.M. Sait, A. Shaheen, Gmdh-based networks for intelligent intrusion detection, ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE, **26**,7:1731-1740 (2013)
10. S.S.S. Sindhu, S. Geetha, A. Kannan, Decision tree based light weight intrusion detection using a wrapper approach, EXPERT SYSTEMS WITH APPLICATIONS, **39**,1:129-141 (2012)
11. A.S. Eesa, Z. Orman, A.M.A. Brifcani, A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems, EXPERT SYSTEMS WITH APPLICATIONS, **42**,5:2670-2679 (2015)
12. T. Shon, X. Kovah, J. Moon, Applying genetic algorithm for classifying anomalous tcp/ip packets, Neurocomputing, **69**,16:2429-2433 (2006)
13. D. Ippoliti, X.B. Zhou, A-ghsom: An adaptive growing hierarchical self organizing map for network anomaly detection, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, **72**,12:1576-1590 (2012)
14. N. Ye, Y. Zhang, C.M. Borror, Robustness of the markov-chain model for cyber-attack detection, IEEE Transactions on Reliability, **53**,1:116-123 (2004)
15. S. Shin, S. Lee, H. Kim, S. Kim, Advanced probabilistic approach for network intrusion forecasting and detection, EXPERT SYSTEMS WITH APPLICATIONS, **40**,1:315-322 (2013)
16. T. Shon, J. Moon, A hybrid machine learning approach to network anomaly detection, Information Sciences, **177**,18:3799-3821 (2007)
17. R.M. Elbasiony, E.A. Sallam, T.E. Eltobely, M.M. Fahmy, A hybrid network intrusion detection framework based on random forests and weighted k-means, Ain Shams Engineering Journal, **4**,4:753–762 (2013)
18. G. Liu, Z. Yi, S. Yang, A hierarchical intrusion detection model based on the pca neural networks, Neurocomputing, **70**,7:1561-1568 (2007)
19. Significant feature selection using computational intelligent techniques for intrusion detection. London: Springer London. 285-306 (2005)
20. R. Datti, S. Lakhina, Performance comparison of features reduction techniques for intrusion detection system, International Journal of Compute Science and Technology, **3**,1:332–335 (2012)
21. C.-F. Tsai, C.-Y. Lin, A triangle area based nearest neighbors approach to intrusion detection, Pattern Recognition, **43**,1:222-229 (2010)
22. C. Guo, Y. Zhou, Y. Ping, Z. Zhang, G. Liu, Y. Yang, A distance sum-based hybrid method for intrusion detection, Applied Intelligence, **40**,1:178-188 (2014)
23. W.C. Lin, S.W. Ke, C.F. Tsai, Cann: An intrusion detection system based on combining cluster centers and nearest neighbors, KNOWLEDGE-BASED SYSTEMS, **78**:13-21 (2015)
24. W. Hong, X. Li, The new graphical features of star plot for k nearest neighbor classifier.4682, Berlin, Heidelberg: Springer Berlin Heidelberg. 926-933 (2007)
25. J. Parkinson, M. Blaxter, Simitri - visualizing similarity relationships for groups of sequences, BIOINFORMATICS, **19**,3:390-395 (2003)
26. B. Luo, J.B. Xia, A novel intrusion detection system based on feature generation with visualization strategy, EXPERT SYSTEMS WITH APPLICATIONS, **41**,9:4139-4147 (2014)