ARTICLE IN PRESS

Knowledge-Based Systems xxx (2015) xxx-xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



27

28

29

30

31

32

33

34

35

36

37

38

39

40 41

42

43 44 45

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

82

CANN: An intrusion detection system based on combining cluster 4 01 centers and nearest neighbors

7 Q2 Wei-Chao Lin^a, Shih-Wen Ke^b, Chih-Fong Tsai^{c,*}

8 ^a Department of Computer Science and Information Engineering, Hwa Hsia University of Technology, Taiwan

9 ^b Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan

10 ^c Department of Information Management, National Central University, Taiwan

ARTICLE INFO

33

11

5 6

Article history: 14 15 Received 4 April 2014

16 Received in revised form 17 December 2014

17 Accepted 17 January 2015

- 18 Available online xxxx
- 19 Keywords: 20 Intrusion detection
- 21 Anomaly detection
- 22 Feature representation
- 23 Cluster center
- 24 Nearest neighbor
- 25

ABSTRACT

The aim of an intrusion detection systems (IDS) is to detect various types of malicious network traffic and computer usage, which cannot be detected by a conventional firewall. Many IDS have been developed based on machine learning techniques. Specifically, advanced detection approaches created by combining or integrating multiple learning techniques have shown better detection performance than general single learning techniques. The feature representation method is an important pattern classifier that facilitates correct classifications, however, there have been very few related studies focusing how to extract more representative features for normal connections and effective detection of attacks. This paper proposes a novel feature representation approach, namely the cluster center and nearest neighbor (CANN) approach. In this approach, two distances are measured and summed, the first one based on the distance between each data sample and its cluster center, and the second distance is between the data and its nearest neighbor in the same cluster. Then, this new and one-dimensional distance based feature is used to represent each data sample for intrusion detection by a k-Nearest Neighbor (k-NN) classifier. The experimental results based on the KDD-Cup 99 dataset show that the CANN classifier not only performs better than or similar to k-NN and support vector machines trained and tested by the original feature representation in terms of classification accuracy, detection rates, and false alarms. I also provides high computational efficiency for the time of classifier training and testing (i.e., detection).

© 2015 Published by Elsevier B.V.

46

1. Introduction 47

Advancements in computing and network technology have 48 made the activity of accessing the Internet an important part of 49 50 our daily life. In addition, the amount of people connected to the Internet is increasing rapidly. However, the high popularity of 51 world-wide connections has led to security problems. 52

53 Traditionally, some techniques, such as user authentication, data encryption, and firewalls, are used to protect computer secu-54 55 rity. Intrusion detection systems (IDS), which use specific analytical technique(s) to detect attacks, identify their sources, and alert 56 57 network administrators, have recently been developed to monitor 58 O3 attempts to break security [3]. In general, IDS are developed for sig-59 nature and/or anomaly detection. For signature detection, packets 60 or audit logs are scanned to look for sequences of commands or events which are previously determined as indicative of an attack. 61 On the other hand, for anomaly detection, IDS use behavior 62

> * Corresponding author. Tel.: +886 3 422 7151; fax: +886 3 425 4604. E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.knosys.2015.01.009 0950-7051/ \odot 2015 Published by Elsevier B.V.

patterns which could indicate malicious activities and analyzes past activities to recognize whether the observed behaviors are normal. As early IDS largely used signature detection to detect all the attacks captured in their signature databases, they suffer from high false alarm rates. Recent innovative approaches including behavior-based modeling have been proposed to detect anomalies include data mining, statistical analysis, and artificial intelligence techniques [21,28].

Much related work in the literature focuses on the task of anomaly detection based on various data mining and machine learning techniques. There have been many recent studies, which focus on combining or integrating different techniques in order to improve detection performance, such as accuracy, detection, and/or false alarm rates (see Table 1 in Section 2.4).

However, there are two limitations to existing studies. First, although more advanced and sophisticated detection approaches and/or systems have been developed, very few have focused on feature representation for normal connections and attacks, which 81 is an important issue in enhancing detection performance. There is a huge amount of related studies using either the KDD-Cup 99

Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009

ARTICLE IN PRESS

W.-C. Lin et al./Knowledge-Based Systems xxx (2015) xxx-xxx

83 or DARPA 1999 dataset for experiments, however there is no an 84 exact answer to the question about which features of these data-85 sets are more representative. Second, the time taken for training 86 the systems and for the detection task to further validate their sys-87 tems are not considered in many evaluation methods. Recent sys-88 tems that combine or integrate multiple techniques require much 89 greater computational effort. As a result, this can degrade the effi-90 ciency of 'on-line' detection.

Therefore, in this study, we propose a novel feature representa-91 92 tion method for effective and efficient intrusion detection that is 93 based on combining cluster centers and nearest neighbors, which 94 we call CANN. Specifically, given a dataset, the *k*-means clustering algorithm is used to extract cluster centers of each pre-defined cat-95 egory. Then, the nearest neighbor of each data sample in the same 96 97 cluster is identified. Next, the sum of the distance between a spe-98 cific data sample and the cluster centers and the distance between 99 this data sample and its nearest neighbor is calculated. This results 100 in a new distance based feature that represents the data in the 101 given dataset. Consequently, a new dataset containing only one dimension (i.e., distance = based feature representation) is used 102 103 for k-Nearest Neighbor classification, which allows for effective 104 and efficient intrusion detection.

105 The idea behind CANN is that the cluster centers or centroids for 106 a given dataset offer discrimination capabilities for recognition 107 both similar and dissimilar classes [9,10,35]. Therefore, the dis-108 tances between a data sample and these identified cluster centers 109 are likely to provide some further information for recognition. Sim-110 ilarly, the distance between a specific data sample and its nearest 111 data sample in the same class also has some discriminatory power.

112 The rest of this paper is organized as follows. Section 2 reviews 113 related literature including offering brief descriptions of super-114 vised and unsupervised machine learning techniques. The techniques used in this paper are also described. Moreover, the 115 techniques used, datasets and evaluation strategies considered in 116 117 related work are compared. The proposed approach for intrusion 118 detection is introduced in Section 3. Section 4 presents the exper-119 imental setup and results. Finally, some conclusions are provided 120 in Section 5.

121 2. Literature review

122 2.1. Machine learning

123 Machine learning requires a system capable of the autonomous 124 acquisition and integration of knowledge. This capacity includes 125 learning from experience, analytical observation, and so on, the 126 result being a system that can continuously self-improve and thereby offers increased efficiency and effectiveness. The main goal 127 of the study of machine learning is to design and develop algo-128 129 rithms and techniques that allow computers to learn. In general, 130 there are two types of machine learning techniques, supervised 131 and unsupervised [22] which are described in greater detail below.

2.2. Supervised learning 132

133 Supervised learning or classification is one common type of machine learning technique for creating a function from a given 134 135 set of training data. The training data are composed of pairs of 136 input objects and their corresponding outputs. The output of the 137 function can be a continuous value, and can predict a class label 138 of the input object. Particularly, the learning task is to compute a 139 classifier that approximates the mapping between the input-out-140 put training examples, which can correctly label the training data 141 with some level of accuracy.

The k-Nearest Neighbor (k-NN) algorithm is a conventional non-parametric classifier used in machine learning [22]. The purpose of this algorithm is to assign an unlabelled data sample to the class of its k nearest neighbors (where k is an integer). Fig. 1 shows an example for a *k*-NN classifier where k = 5. Consider the 5 nearest neighbors around X for the unlabelled data to be classified. There are three 'similar' patterns from class C_2 and two from class C_1 . Taking a majority vote enables the assignment of X to the C_2 class.

According to Jain et al. [13], k-NN can be conveniently used as a benchmark for all the other classifiers since it is likely to provide a reasonable classification performance in most applications. Other well-known supervised learning techniques used in intrusion detection include support vector machines, artificial neural networks, decision trees, and so on [3,33,37].

2.3. Unsupervised learning

Unsupervised learning or clustering is a method of machine learning where a model is fit to observations. It differs from supervised learning in the absence of prior output. In unsupervised learning, a data set of input objects is gathered first. The input objects are typically treated as a set of random variables. A joint density model is then built for the data set [22].

The machine simply receives the inputs x_1, x_2, \ldots, x_n , obtaining neither supervised target outputs, nor rewards from its environment. It may seem somewhat mysterious to imagine what the machine could possibly learn given that it does not get any feedback from its environment. However, it is possible to develop a formal framework for unsupervised learning based on the notion that the machine's goal is to build representations of the input that can be used for decision making, predicting future inputs, efficiently communicating the inputs to another machine, etc.

The *k*-means clustering algorithm is the simplest and most commonly used unsupervised machine learning technique [14] being a simple and easy way to classify a given dataset through a certain number of clusters. The goal of the *k*-means algorithm is to find *k* points of a dataset, which can best represent this dataset in a certain number of groups. The point, k, is the cluster center or centroid of each cluster.

In particular, *k*-means is used to cluster or group N data points into K disjoint subsets S_i containing N_i data points so as to minimize the sum-of-squares criterion,

$$J = \sum_{j=1}^{k} \sum_{n \in S_j} |x_n - \mu_j|^2$$
(1)

where x_n is a feature vector representing the *n*-th data point and μ_i is the geometric centroid of the data points in S_j.

In the literature, it can be seen that some clustering techniques are combined with specific supervised learning techniques for intrusion detection. For example, Khan et al. [16] combined self-organizing maps (SOM) and support vector machines, Xiang et al. [37] combined Bayesian clustering and decision trees, and C-means clustering and artificial neural networks are combined in Zhang et al. [38].

2.4. Comparison of related work

A number of related intrusion detection systems are compared and the results shown in Table 1. In particular, we compare the machine learning techniques used for developing the detection systems, datasets used for experiments, evaluation methods considered, baseline classifiers for comparisons, etc. in relevant studies. For a detailed review, readers can refer to García-Teodoro 201 et al. [7] and Tsai et al. [34]. 202

142

143

144

150 151

152 153

154

155 156

157

158

159 160 161

162 163

164

165 166 167

168 169

170 171 172

177 178

179 180 181

182 183

185 186

> 187 188 189

- 190 191
- 192 193 194

195

1 February 2015 KNOSYS 3053

ω

Q1		
q	Table	1

Evaluation method	Dasenne	selecti
DR ^b , FP ^c , accuracy, ROC curve ^d	N/A	Yes
DR, ROC curve	NB ^f , RF ^g , DT, AdaBoost	Yes
DR. FP. FN ⁱ	SVM	No
DR, ROC curve	DT, SVM	No
Accuracy, Recall, Precision, FP, FN, ROC	NB, ANN ^k	Yes
DR, FP, ROC	Markov chain model	Yes
DR	DT SVM	Yes
DR	N/A	Yes
Precision Recall F-measure	DT/NB ^m ANN	No
DR FP	SVM k -NN	No
DR. FP	ANN. SOMP	No
DR	N/A	No
DR FA^q Run time	N/A	No
DR Run time	DT	Ves
DR FA	EI	No
ED EN	ANN 2 laver ANN	Ves
DR	N/A	No
EP EN accuracy	SVM	No
TD ^S ED	SVM ANN k -NN	No
DR FA FD	SVM DT SOM	Vec
	5 VIN, D1, 50M	103
DR	GA	No
Accuracy	SVM, DT	No
DR, FP, FN	SVM	Yes
DR, FP, FN	ANN, k-NN, SVM	Yes
DR. FP	SOM	Yes
DR, FP	ANN	No
	7	

- ^b DR: detection rate.
- ^c FP: false positive.
- ^d ROC curve: Receiver Operating Characteristic curve.
- ^e GHSOM: Growing Hierarchical SOM.
- ^f NB: Naïve Bayes.
- ^g RF: Random Forest.
- ^h SVM: support vector machine.
- ⁱ FN: false negative.
- ^j GMDH: Group Method for Data Handling.
- ^k ANN: Artificial Neural Networks.
- ¹ SA: Simulated Annealing.
- ^m NB: Naïve Bayes.
- ⁿ FL: Fuzzy Logic.
- ° AR: Association Rules.
- ^p SOM: self-organizing maps.
- ^q FA: false alarm.
- ^r GA: Genetic Algorithm.
- ^s TP: true positive.

ARTICLE IN PRESS

W.-C. Lin et al. / Knowledge-Based Systems xxx (2015) xxx-xxx

Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl.



Fig. 1. A *k*-Nearest Neighbor rule for k = 5.

As we can see from Table 1, DARPA1998 and KDD-Cup99 are the 203 most commonly used datasets for simulating intrusion detection 204 while SVM and *k*-NN classifiers are popular baseline techniques 205 206 used in related work proposing novel techniques. In addition, it 207 is the trend that related studies consider combining or integrating 208 two different techniques in order to improve the intrusion detec-209 tion performance.

210 Related studies where the main focus is on developing more 211 advanced techniques to improve the intrusion detection performance, tend to rely only on some feature selection methods, such 212 213 as principal component analysis (PCA), to filter out unrepresenta-214 tive features in the DARPA1998 and/or KDD-Cup99 dataset (e.g., 215 [20,25,39]. However, according to Table 1, very few studies com-216 bine feature selection and classification techniques to examine 217 the effect of performing feature selection on the intrusion detec-218 tion performance.

219 However, each study uses different features over the same data-220 sets. Therefore, currently it is not known what features are more 221 representative in the two datasets.

222 The detection rate (DR), false positive (FP), false negative (FN). 223 true positive (TP), false alarm (FA), and the accuracy rate are most 224 often examined for evaluation measurements. Only Hu et al. [12] 225 and Xiang et al. [37] considered the run time during intrusion 226 detection as another performance indicator. It is known that for 227 intrusion detection systems the computational effort, i.e., run time, 228 for online detection should be as short as possible. Although this is 229 a very critical issue, very few have considered the detection time of 230 their systems.

The above discussion leads us to propose the method described below, which can not only extract representative features for improving detection performances, but also provide computational efficiency.

3. CANN: the proposed approach 235

3.1. The CANN process 236

231

232

233

234

01

The proposed approach is based on two distances which are 237 238 used to determine the new features, between a specific data point 239 and its cluster center and nearest neighbor respectively. CANN is comprised of three steps as shown in Fig. 2. 240

241 Given a training dataset T, the first step is to use a clustering technique to extract cluster centers. The number of clusters is 242 based on the number of classes to be classified. Since intrusion 243 detection is one classification problem, the chosen dataset has 244 245 already defined the number of classes to be classified. Therefore, 246 for example, if the given dataset is a three-class problem, then 247 the number of clusters is defined as three. Besides extracting clus-

Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009



(a) Extraction of cluster centers and nearest neighbors





(c) Classifier training and testing

Fig. 2. The CANN process.

ter centers, each data point of the given dataset and its nearest 248 neighbor in the same cluster is identified. This can be done by cal-249 culating the distances between one specific data point (Di) and all 250 of the other data in the same cluster. Then, the shortest distance between two data examples representing Di and its nearest neighbor can be found.

The second step is to measure and sum the distance (dis1) between all data of the given dataset and the cluster centers and the distance (dis2) between each data point and its nearest neighbor in the same cluster. This leads to a new distance based feature value to represent each data point of the given dataset, which is T. That is, the original features (i.e., the number of dimensions is usually larger than one) are replaced by one new dimension feature.

To test the new unknown data for intrusion detection, the test-261 ing set S is combined with the original training set T. Then, the pro-262 cesses of extracting cluster centers and nearest neighbors 263 (Fig. 2(a)) and new data formation (Fig. 2(b)) are executed. During 264 these processes, only the data samples in S are considered. As a 265 result, the new distance based feature dataset S' is obtained. There-266

256

257

258

259

fore, *T* and *S'* are used to train and test the *k*-NN classifier for intru-sion detection.

269 3.2. Extraction of cluster centers and nearest neighbors

270 To extract cluster centers, a clustering technique can be applied 271 in this stage. In this study, the k-means clustering algorithm is 272 used. Fig. 3 shows an example where the chosen dataset consisting of 12 data samples (N_1 to N_{12}) is a five-class classification problem. 273 274 Thus, the number of clusters is defined as five (i.e., k = 5) for the kmeans clustering algorithm. As a result, there are five clusters, 275 which each cluster containing a cluster center (i.e., C_1 , C_2 , C_3 , C_4 , 276 and C_5). 277

On the other hand, to identify the nearest neighbor of a data point, D_i for example, the *k*-NN approach is used where the distance between D_i and each of the other data points in the same cluster can be obtained. That is, the nearest neighbor of D_i is based on the shortest distance identified by *k*-NN. Therefore, for Fig. 3, N_1 is the nearest neighbor of D_i .

284 3.3. New data formation

295

303

After the cluster center and nearest neighbor for every data 285 point of the chosen dataset are extracted and identified, two types 286 287 of distances are calculated and then summed. The first type is 288 based on the distance from each data point to the cluster centers. That is, if there are three cluster centers, then there are three dis-289 tances between a data point to the three cluster centers, respec-290 291 tively. The second type is based on the distance from each data 292 point to its nearest neighbor. Fig. 4 shows an example of five clusters, in which the two types of distances for the data point D_i are 293 obtained by 294

297
$$D_i = \overline{D_i C_1} + \overline{D_i C_2} + \overline{D_i C_3} + \overline{D_i C_4} + \overline{D_i C_5}$$
(2)

Specifically, the distance between two data points is based on the Euclidean distance. For example, given that data *A* and *B* contain 00 Q4 *n*-dimensional features, their Euclidean distance is based on

dis
$$AB = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

= $\sqrt{\sum_{i=1}^n (a_i - b_i)^2}$ (3)

Following the example shown in Fig. 3, the CANN approach transforms the original *n*-dimensional features to the one-dimensional distance feature by



Fig. 3. An example of extracting cluster centers and nearest neighbors.



Fig. 4. An example of new data formation.

$$Dis(D_i) = \sum_{j=1}^{5} dis(D_i, C_j) + \sum_{k=1}^{5} dis(D_i, N_k)$$
(4)
309

,where D_i is the *i*-th data point of a given dataset, C_j is the *j*-th cluster center identified by *k*-means, and N_k is the *k*-th data point which is the nearest neighbor of D_i . These distance values are summed to represent the feature of D_i . As a result, each data point of the given dataset has its one dimensional distance feature. Finally, for the dataset containing *m* data samples, in which each sample originally has *n* dimensions, the CANN approach produces a new dataset containing *m* data samples, in which each sample has one dimension, i.e., the distance feature.

For classifier construction, the final step of CANN, the new dataset is divided into the training and testing datasets to train and test a specific classifier. In this study, we consider the *k*-NN classifier since it is easy to implement and widely used as a baseline classifier in many applications.

It should be noted that the CANN process can be applied to any dataset with and without feature selection. For instance, the collected dataset contains D dimensional features. When dimensionality reduction is required, a chosen feature selection algorithm is used to select some representative features from the training set resulting in D' features where D' < D. Next, the reduced training dataset is used for the CANN process, which can be regarded as T shown in Fig. 2(a) and (b). For the testing set, its features are the same as the ones identified in the training set, i.e. D'. Then, the reduced testing set containing D' dimensional features, which can be regarded as S shown in Fig. 2(c), is used for intrusion detection.

4. Experiments

4.1. Experimental setup

4.1.1. The dataset

Since there is no standard dataset for intrusion detection, the 339 dataset used in this paper is based on the KDD-Cup 99 dataset¹ 340 containing 494,020 samples, which is the most popular and widely 341 used in related work (c.f., Table 1). Specifically, each data sample 342 represents a network connection represented by a 41-dimensional 343 feature vector, in which 9 features are of the intrinsic types, 13 fea-344 tures are of the content type, and the remaining 19 features are of 345 the traffic type. Each pattern of the dataset is labeled as belonging 346 to one out of five classes, which are *normal* traffic and four different 347 classes of attacks, i.e., probing, denial of service (DoS), remote to local 348

¹ http://www.sigkdd.org/kddcup/index.php?section=1999&method=data.

Q1 Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009

5

333

334

335

336

337

338

207

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

426

6

W.-C. Lin et al. / Knowledge-Based Systems xxx (2015) xxx-xxx

349 (R2L), and user to root (U2R). Therefore, this is a five-class classifica-350 tion dataset and the k value of k-means to extract cluster centers is 351 set to 5.

352 The dataset is a large and high dimensional dataset. A dimensional reduction (or feature selection) step based on principal com-353 ponent analysis (PCA) has been considered in related work for 354 355 example, to filter out unrepresentative features (e.g., [20,25]. How-356 ever, as there is no standard answer about which features are well representative for intrusion detection, this study considers two dif-357 ferent numbers of features, 6 and 19, in order to fully assess the 358 performance of CANN. The 6-dimension KDD dataset contains 359 'land', 'urgent', 'num_failed_logins', 'num_shells', 'is_host_login', 360 and 'num_outbound_cmds' [35]. On the other hand, the KDD data-361 set containing 19 features is based on the work of Zhang et al. [39]. 362

4.1.2. Classifier design

363

388

3

The final step of CANN for classifier construction is based on the 364 k-NN algorithm. One of the baseline classifiers to compare with 365 CANN is the k-NN classifier. That is, the baseline k-NN classifier 366 is trained and tested over the original (6 and 19 dimensional) 367 368 KDD datasets without the CANN process. In particular, for the k 369 value of the k-NN algorithm used in CANN and the baseline, we 370 examine $k = 1, 3, 5, \dots, 25$ in order to obtain the best k-NN classifier 371 for comparison.

372 Another baseline classifier based on the support vector machine 373 (SVM), a popularly used baseline classifier in the literature, is considered. The polynomial kernel function is used to construct the 374 375 SVM, in which the degree of the polynomial is set from 1 to 5 to 376 obtain the SVM classifier providing the best performance for 377 comparison.

378 The 10-fold cross validation method is used to train and test 379 these classifiers, where the dataset is divided into 10 un-duplicated 380 subsets, and any nine of the ten subsets are used for training and 381 the remaining one for testing. Thus, the classifier will be trained 382 and tested10 times.

383 4.1.3. Evaluation methods

384 In this study, we consider the rates of accuracy, detection and 385 false alarms, which are widely used in literature, to evaluate the 386 performance of intrusion detection (c.f., Table 1). They can be cal-387 culated by a confusion matrix as shown in Table 2.

Then, the rates of accuracy, detection and false alarm can be obtained by:

$$\begin{array}{l} \textbf{395} \qquad \text{Detection Rate} = \frac{11}{TP + F} \\ \textbf{396} \end{array}$$

False Alarm =
$$\frac{FP}{FP + TN}$$

Table 2 Confusion matrix

C	onitusion matrix.		
	$\downarrow \textit{Actual} \setminus \textit{predicted} \rightarrow$	Normal	Intrusions (Attacks)
	Normal	TN	FP
	Intrusions (Attacks)	FN	TP

True Positives (TP): the number of malicious executables correctly classified as malicious

True Negatives (TN): the number of benign programs correctly classified as benign. False Positives (FP): the number of benign programs falsely classified as malicious. False Negative (FN): the number of malicious executables falsely classified as benign.

Moreover, in order to understand the efficiency of CANN in terms of 399 training and testing efforts, the time for executing all three steps 400 will be examined and compared with the baseline classifiers. 401

4.2. Study I: the 6-dimensional dataset

Study I is based on the 6-dimensional KDD dataset. Table 3 403 shows the performance of CANN, TANN [35], and *k*-NN. The results 404 show that CANN provides the highest accuracy rate of 99.761%, sig-405 nificantly outperforming TANN and k-NN. In addition, CANN can 406 produce the highest detection rate and lowest false alarm rate. 407

Tables 4 and 5 show the confusion matrices of *k*-NN and CANN, 408 respectively, for the 6-dimensional dataset. It is interesting that 409 with the k-NN classifier, most of the normal accesses are recog-410 nized for probing U2R and R2L as DoS attacks, which results in a 411 very high false alarm rate. However, almost all DoS attacks are 412 classified correctly. This suggests that these 6-dimensonal features 413 are not representative enough to allow k-NN to distinguish 414 between these five different classes. In other words, the results 415 obtained using k-NN for probing U2R and R2L attacks are very sim-416 ilar to the DoS attacks in the 6-dimensional feature space. 417

Although CANN performs the best, one weakness is that it 418 totally misclassifies U2R and R2L attacks into the normal traffic 419 class (see Table 5). This may be because these two attack categories 420 are similar to the pattern of normal traffic after performing the 421 CANN process. The proposed feature representation extracted from 422 the 6-dimensional dataset allows the classifier to distinguish 423 between the normal traffic, probing, and DoS classes quite well 424 (see Table 6). Q5 425

4.3. Study II: the 19-dimensional dataset

The second study is based on a 19-dimensional KDD dataset. 427 Table 7 shows the performance of CANN, *k*-NN, and SVM. As can 428

Table 3

T:

C

(6)

(7)

Performances of CANN, k-NN, and SVM over the 6-dimensional dataset.

	Accuracy	Detection Rate	False Alarm
CANN (k = 1)	99.76% (1)	99.99% (1)	0.003% (1)
TANN (k = 21)	93.87% (2)	93.39% (2)	28.69% (2)
k-NN (k = 5)	80.65% (3)	80.32% (3)	99.92% (3)

able 4 onfusion matrix of <i>k</i> -NN over th	ne 6
Predicted	

Predicted						Accuracy (%)	
	Normal	Probe	DoS	U2R	R2L		
Actual							
Normal	73		97,204	0	0	0.075	
Probe	0	0	4107	0	0	100	
Dos	12	0	391,446	0	0	99.99	
U2R	0	0	52	0	0	0	
R2L	0	0	1126	0	0	0	

-dimensonal dataset.

Table 5

Confusion matrix of CANN over the 6-dimensonal dataset.

Predicted						Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Actual						
Normal	97,275	0	2	0	0	99.99
Probe	0	4106	1	0	0	99.98
Dos	2	0	391,456	0	0	99.99
U2R	52	0	0	0	0	0
R2L	1126	0	0	0	0	0

Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

W.-C. Lin et al. / Knowledge-Based Systems xxx (2015) xxx-xxx

Table 6

Performances of CANN, k-NN, and SVM over the 19-dimensonal dataset.

	Accuracy	Detection rate	False alarm
CANN (k = 1)	99.46% (2)	99.28% (2)	2.95% (2)
SVM (degree = 2)	95.37% (3)	98.97% (3)	4% (3)
k-NN (k = 1)	99.89% (1)	99.92% (1)	0.32% (1)

Table 7

Confusion matrix of k-NN over the 19-dimensional dataset.

Predicted						Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Actual						
Normal	96,964	42	118	17	126	99.68
Probe	48	4045	11	0	3	98.49
Dos	52	6	391,394	3	3	99.98
U2R	29	1	3	9	10	17.31
R2L	80	6	4	3	1033	91.74

be seen in this table, the *k*-NN classifier performs the best in terms
of accuracy and detection rates. The SVM classifier performs the
worst. Although CANN does not outperform *k*-NN, the accuracy
and detection rates of *k*-NN and CANN are very similar, being less
than 1%. This indicates that there is no significant difference in
their performances.

Tables 7–9 show the confusion matrices of *k*-NN, CANN, and
SVM respectively, for the 19-dimensional dataset. It is worth noting here that *k*-NN can accurately recognize most normal accesses
over the 19-dimensional dataset, which is different from the result
obtained using the 6-dimensional dataset. However, these 19dimensional features are not discriminative enough for *k*-NN to
detect the U2R attacks.

On the other hand, although CANN can correctly classify some 442 U2R and R2L attacks into the right attack groups, the accuracy rates 443 are not satisfactory, i.e., 3.846% and 57.016%, respectively. This 444 means that the distance based feature of CANN extracted from 445 446 the 19-dimensional features only allows the classifier to better dis-447 tinguish between the normal accesses, probing, and DoS classes. 448 This result is similar to the distance based feature extracted from 449 the 6-dimensional features which shows that CANN is not good at detecting U2R and R2L attacks. In summary, CANN is good at 450

Table 8

Confusion matrix of CANN over the 19-dimensonal dataset.

Predicted						Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Actual						
Normal	94,398	221	2130	35	493	97.04
Probe	201	3598	306	1	1	87.61
Dos	1076	177	390,190	8	7	99.68
U2R	36	1	11	2	2	3.85
R2L	471	1	10	2	642	57.02

Table 9

Confusion matrix of SVM over the 19-dimensonal dataset.

Predicted						Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Actual						
Normal	93,367	3780	130	0	0	95.98
Probe	120	3967	20	0	0	96.59
Dos	20,760	44,639	324,329	1390	340	82.85
U2R	0	0	20	32	0	61.54
R2L	0	69	9	159	889	78.95

detecting the normal traffic, probing, and DoS classes, whereas SVM can be used for detecting the U2R and R2L classes.

These results indicate that the 19-dimensional features are more representative for intrusion detection, making SVM and *k*-NN perform very well. However, it is important to examine the run time of these classifiers. CANN only uses the one-dimensional feature for intrusion detection, but SVM and *k*-NN are trained and tested by the original features, which can result in high computational effort.

4.4. Efficiency evaluation

Table 10 shows the results obtained by comparing the run time of these classifiers.² Note that the data preparation time for k-NN includes data pre-processing and loading. For CANN, it includes the time for the processes of extracting cluster centers and nearest neighbors and new data formation. This comparison does not consider SVM because so much time is needed for training, e.g., over 100 h for the 19-dimensional dataset.

As we can see, a much longer run time is needed for the dataset containing higher numbers of dimensions needs. Thus *k*-NN needs about 25 more hours to deal with the 19-dimensional dataset compared with the 6-dimensional one. However, with the CANN approach, increasing the number of dimensions for the dataset does not greatly affect the run time. In this case, we only need an additional 3 h.

If we consider the results from Study II, CANN does not outperform *k*-NN. However, CANN is still a good candidate for intrusion detection since it saves over two times the run time over the 6and 19-dimensional dataset compared with *k*-NN, while still providing very similar performance to *k*-NN and SVM. In addition, with the 6-dimensional dataset, CANN provides an accuracy, detection rate, and false alarm of 99.76%, 99.99%, and 0.003% respectively, results which are better or similar to the best classifier for the 19-dimensional dataset, which for 99.89%, 99.92%, and 0.0289%, respectively.

Compare the run time with the most recent related works (i.e., testing times). Kim et al. [17] and Nadiammai and Hemalatha [23] obtained times of 11.2 and 8 s, respectively, whereas CANN requires about 13 s. However, it should be noted that it is very difficult to make a direct comparison between these works since the computing environments and relevant settings are different. The run time could certainly be enhanced by using more efficient computing equipment, but not the detection performance. Thus, although the methods discussed in these two works may be more efficient than CANN, they only provide about 99% accuracy and 0.15% false alarm rates, which are all lower than the ones produced by CANN. Moreover, CANN performs better than the winner of the KDD'99 contest, which provides 97.12% accuracy and 0.03 false alarm rates.

4.5. Discussion

Regarding the previous results, if we consider the average accuracy, detection rate, and false alarm rate, for the 6-dimensional dataset, CANN performs the best in terms of the detection rate and false alarm rate, whereas for the 19-dimensional dataset *k*-NN performs the best in terms of accuracy.

We further examine these two approaches over the five classes (including the four attack classes) in advance. Fig. 5 shows the difference in performance. To detect the normal, probe, and DoS classes, the CANN approach performs slightly better than *k*-NN.

 $^{2}\,$ The software is based on Matlab 7 and carried out on an Intel Pentium 4, with 3.4 GHz CPU, and 1.5 GB RAM.

Q1 Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009

ARTICLE IN PRESS

8

W.-C. Lin et al. / Knowledge-Based Systems xxx (2015) xxx-xxx

Table 10

Run time of CANN and *k*-NN.

		Data preparation	Training and testing	Total
6-dimension dataset	CANN	40 min (0.7 h)	1570 min (26 h)	1648 min (27 h)
	k-NN	20 min (0.3 h)	2765 min (46 h)	2785 min (46 h)
19-dimension dataset	CANN	180 min (3 h)	1608 min (27 h)	1750 min (30 h)
	k-NN	20 min (0.3 h)	4210 min (70 h)	4230 min (71 h)



Fig. 5. The performances of CANN and k-NN over 5 classes.

However, *k*-NN can correctly detect some U2R and R2L cases, but
CANN cannot.

These results show that with the 19-dimensional dataset U2R 511 512 and R2L attacks are very hard to detect by all methods, except 513 for *k*-NN. According to the definition of KDD-Cup 99, U2R is either 514 unauthorized access to a local super user or administrator privileges by a local unprivileged user. These attacks are opportunities 515 516 for exploitation where the hacker starts off with a normal user 517 account but attempts to abuse vulnerabilities in the system in 518 order to gain super user privileges e.g., perl, xterm. On the other 519 hand, R2L refers to a remote user obtaining unauthorized user priv-520 ileges on a local host. In the attack the user sends packets to a 521 machine over the internet, to which she/he does not have access 522 to in order to expose the machines vulnerabilities and exploit priv-523 ileges which a local user would have, e.g., xlock, guest, xnsnoop, 524 phf, sendmail dictionary, etc.

525 These results indicate that with 6 and 19 features the classifier 526 cannot detect representative U2R attacks whereas with 19 features 527 it can somewhat detect representative R2L attacks. Similarly, 528 transforming the 6 and 19 features by CANN still cannot lead to 529 effective detection of U2R and R2L attacks. According to Jeya 530 et al. [15], the four attack classes have different representative features among the 41. Therefore, CANN is generally suitable for 531 detection of normal or attack cases in the binary classification 532 533 problem, with a false alarm rate of 0.003%. However, for U2R and R2L attacks, for which there are fewer cases among all accesses, 534 existing approaches still have room for improvement. 535

536 5. Conclusion

and efficient intrusion detection, namely CANN. The CANN539approach first transforms the original feature representation of a
given dataset into a one-dimensional distance based feature. Then,
this new dataset is used to train and test a *k*-NN classifier for
classification.541
542

The experimental results show that CANN performs better than 544 the *k*-NN and SVM classifiers over the original 6-dimension data-545 set, providing higher accuracy and detection rates and a lower false 546 alarm rate. On the other hand, CANN performs similar to the k-NN 547 and SVM classifiers over the original 19-dimension dataset. How-548 ever, the advantage of CANN is that it requires less computational 549 effort than the k-NN or SVM classifiers trained and tested by the 550 two original datasets. In other words, although CANN requires 551 additional computation to extract the distance based features, 552 the training and testing (i.e., detection) time is greatly reduced 553 since the new dataset only contains one dimension. 554

As to the limitations of this research CANN cannot effectively 555 detect U2L and R2L attacks, which means that this one-dimen-556 sional distance based feature representation is not able to well rep-557 resent the pattern of these two types of attacks. This is an issue 558 that future work can look into. One possibility is to consider the 559 weight for the distances between the data to each of the cluster 560 centers and its nearest neighbor. Alternatively, before performing 561 CANN, outlier detection and removal can be employed in order 562 to first filter out noisy or bad data from the given dataset. Finally, 563 as CANN is applicable to the 5-class intrusion detection problem, 564 other domain datasets including different numbers of dimensions 565 and classes can be used to examine its effectiveness. 566

567



Q6 568

Q1 Please cite this article in press as: W.-C. Lin et al., CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.01.009

569 References

570

571

572

573

574

575

576

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

606

607

608

609

- [1] M.S. Abadeh, J. Habibi, Z. Barzegar, M. Sergi, A parallel genetic local search algorithm for intrusion detection in computer networks, Eng. Appl. Artif. Intell. 20 (8) (2007) 1058-1069.
- [2] Z.A. Baig, S.M. Sait, A. Shaheen, GMDH-based networks for intelligent intrusion detection, Eng. Appl. Artif. Intell. 26 (7) (2013) 1731-1740.
- [3] Y. Chen, A. Abraham, B. Yang, Hybrid flexible neural-tree-based intrusion detection systems. Int. J. Intell. Syst. 22 (2007) 337-352.
- 577 [4] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, A. Martinez-Alvarez, Feature 578 selection by multi-objective optimisation: application to network anomaly 579 detection by hierarchical self-organising maps, Knowl.-Based Syst. 71 (2014) 580 322-338.
 - [5] W. Feng, Q. Zhang, G. Hu, J.X. Huang, Mining network data for intrusion detection through combining SVMs with ant colony networks, Future Gener. Comput. Syst. 37 (2014) 127–140.
 - [6] 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 Syst. Appl. 42 (5) (2015) 2670-2679.
 - [7] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomalybased network intrusion detection: techniques, systems and challenges, Comput Secur 28 (2009) 18-28
 - [8] G. Giacinto, R. Perdisci, M. Del Rio, F. Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, Inf. Fusion 9 (2008) 69-82
 - [9] H. Guan, J. Zhou, M. Guo, A class-feature-centroid classifier for text categorization, in: Proceedings of the International Conference on World Wide Web, 2009, pp. 201–209.
 - [10] E.-H. Han, G. Karypis, Centroid-based document classification: analysis and experimental results, in: Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, 2000, pp. 424-431.
 - [11] J.V. Hansen, P.B. Lowry, R.D. Meservy, D.M. McDonald, Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection, Decis. Support Syst. 43 (2007) 1362-1374.
- 602 W. Hu, W. Hu, S. Maybank, AdaBoost-based algorithm for network intrusion [12] 603 detection, IEEE Trans. Syst. Man Cybernet. - Part B: Cybernet. 38 (2) (2008) 604 577-583. 605
 - [13] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4-37.
 - [14] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Survey 31 (3) (1999) 264-323.
- P.G. Jeya, M. Ravichandran, C.S. Ravichandran, Efficient classifier for R2L and [15] 610 U2R attacks, Int. J. Comput. Appl. 45 (21) (2012) 28-32.
- 611 L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using [16] 612 support vector machines and hierarchical clustering, VLDB J. 16 (2007) 507-613 521
- 614 [17] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating 615 anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) (2014) 616 1690-1700.

- [18] Y. Li, L. Guo, An active learning based TCM-KNN algorithm for supervised network intrusion detection, Comput. Secur. 26 (2007) 459-467.
- [19] S.-W. Lin, K.-C. Ying, C.-Y. Lee, Z.-J. Lee, An intelligent algorithm with feature selection and decision rules applied to anomaly instruction detection, Appl. Soft Comput. 12 (12) (2012) 3285-3290.
- [20] G. Liu, Z. Yi, S. Yang, A hierarchical intrusion detection model based on the PCA neural networks, Neurocomputing 70 (2007) 1561-1568.
- G.A. Marin, Network security basics, IEEE Secur. Privacy 3 (6) (2005) 68-72. [21
- T. Mitchell, Machine Learning, McGraw Hill, New York, 1997. [22]
- G.V. Nadiammai, M. Hemalatha, Effective approach toward Intrusion Detection [23] System using data mining techniques, Egypt. Infor. J. 15 (1) (2014) 37–50.
- [24] N.J. Nisson, Introduction to Machine Learning, MIT Press, USA, 1996.
- [25] T. Ozyer, R. Alhajj, K. Barker, Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening, J. Network Comput. Appl. 30 (2007) 99-113.
- [26] S. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, J. Network Comput. Appl. 30 (2007) 114-132.
- P. Sangkatsanee, N. Wattanapongsakorn, C. Charnsripinyo, Practical real-time [27] intrusion detection using machine learning approaches, Comput. Commun. 34 (18) (2011) 2227-2235.
- [28] S.T. Sarasamma, Q.A. Zhu, J. Huff, Hierarchical Kohonenen net for anomaly detection in network security, IEEE Trans. Syst. Man Cybernet. – Part B: Cybernet. 35 (2) (2005) 302-312.
- S. Shin, S. Lee, H. Kim, S. Kim, Advanced probabilistic approach for network intrusion forecasting and detection, Exp. Syst. Appl. 40 (1) (2013) 315-322.
- [30] T. Shon, J. Moon, A hybrid machine learning approach to network anomaly detection, Inform. Sci. 177 (2007) 3799-3821.
- [31] T. Shon, X. Kovah, J. Moon, Applying genetic algorithm for classifying anomalous TCP/IP packets, Neurocomputing 69 (2006) 2429-2433.
- [32] A. Tajbakhsh, M. Rahmati, A. Mirzaei, Intrusion detection using fuzzy association rules, Appl. Soft Comput. 9 (2009) 462-469.
- X. Tong, Z. Wang, H. Yu, A research using hybrid RBF/Elman neural networks for intrusion detection system secure model, Comput. Phys. Commun. 180 (10) (2009) 1795-1801.
- [34] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, Intrusion detection by machine learning: a review, Expert Syst. Appl. 36 (10) (2009) 11994-12000.
- C.-F. Tsai, C.-Y. Lin, A triangle area based nearest neighbor approach to intrusion detection, Pattern Recogn. 43 (1) (2010) 222-229.
- [36] G. Wang, J. Hao, J. Ma, L. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, Exp. Syst. Appl. 37 (9) (2010) 6225-6232
- [37] C. Xiang, P.C. Yong, L.S. Meng, Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees, Pattern Recogn. Lett. 29 (2008) 918–924.
- C. Zhang, J. Jiang, M. Kamel, Intrusion detection using hierarchical neural network, Pattern Recogn. Lett. 26 (2005) 779-791.
- [39] X.-Q. Zhang, C.H. Gu, J.J. Lin, Intrusion detection system based on feature selection and support vector machine, in: International Conference on Communications and Networking in China, 2006, pp. 1-5.

666 667

9

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664