A Hybrid Methodologies for Intrusion Detection Based Deep Neural Network with Support Vector Machine and Clustering Technique

Tao Ma, Yang Yu, Fen Wang, Qiang Zhang and Xiaoyun Chen

Abstract This paper proposes a novel approach called KDSVM, which utilized the k-mean techniques and advantage of feature learning with deep neural network (DNN) model and strong classifier of support vector machines (SVM) , to detection intrusion networks. KDSVM is composed of two stages. In the first step, the dataset is divided into k subset based on every sample distance by the cluster centers of k-means approach, and in the second step, testing dataset is distanced by the same cluster center and fed into the DNN model with SVM model for intrusion detection. The experimental results show that the KDSVM not only performs better than SVM, BPNN, DBN-SVM (Salama et al., Soft computing in industrial applications, 2011 [21]) and Bayes tree models in terms of detection accuracy and abnormal types of attacks found. It also provides an effective tool for the study and analysis of intrusion detection in the large network.

Keywords Intrusion detection systems · Deep neural network · Hybrid model · K-means clustering · Support vector machine

T. Ma (✉) · Y. Yu · X. Chen (✉)
School of Information Science and Engineering, Lanzhou University,
Lanzhou 730000, China
e-mail: mat13@lzu.edu.cn

X. Chen
e-mail: chen_xiaoyun@yeah.net

T. Ma · F. Wang
School of Mathematical and Computer Science, Ningxia Normal University,
Guyuan, Ningxia 756000, China

Q. Zhang
Statistics & Research Division, China Insurance Regulatory Commission Ningxia Bureau,
Yinchuan, Ningxia 750000, China

N.Y. Yen and J.C. Hung (eds.), *Frontier Computing*, Lecture Notes
in Electrical Engineering 422, DOI 10.1007/978-981-10-3187-8_13

1 Introduction

Network intrusion detection is a new network security mechanism designed to detect, prevent and repel unauthorized access to a communication or computer network. An intrusion detection system (IDS) plays a crucial role in maintaining a safe and secure network. In recent years, a huge network data is generated due to the application of new network technologies and equipment, which leads to the declining of the defect rates. The intrusion detection process is a difficult and complicated one in terms of detection accuracy, detection speed, the dynamic nature of the networks and the available processing power for processing high volumes of data from distrusted network systems [15]. Recently, many researchers proposed innovative approaches in recent years.

These methods, based on detecting in team of behavior-based and resource type of access, are divided into four categories. The first category is to detect anomalies based on statistical analysis, such as, Bayesian model [3], Decision Tree. Anomaly-based techniques build models of normal network samples and detect the samples that deviate from these models in literature [7]. It can detect new types of attacks via already known normal events. Therefore the anomaly detection approach suffers from a high rate of failure. The second category is anomaly detection approach where most methods require a set of standard normal dataset to train the classifier and check whether new sample fits the model. These principle methods are coined as outlier detection algorithm, such as k-mean, self-organizing maps and unsupervised support vector machines approaches [8]. The third category employing AI techniques to detect attack types by taking advantage of machine learning can prioritize solutions to certain problem, such as, SVM [5], RF [23], genetic algorithm (GA) and artificial neural networks etc. The last category is the hybrid and ensemble detecting methods that integrated advantages of different or same methods in order to incase accuracy of detection. These approaches include bagging, adaboost [19] technology, and the PSO-K-means ensemble approach [16]. The PSO-k-means methods could achieve optimal numbers of clusters and increase the high detection rate which utilized K-means technology to detect attack types in networks. In addition, the SVM-KNN-PSO ensemble method proposed by [1] can obtain the best results, which used advantage of nonlinear processing capability and classification capability based distance for each sample. However, their work is based on binary classification methods, which can distinguish between the two states. Alom et al. [2] combines the deep belief network (DBN) and SVM model, the proposed model utilized DBN to select the feature and SVM to capture the rules from attack process, then the reduction dimension output data by DBN regarded as the input dataset fed SVM into detection intrusion. In above methods, it is supposed that each feature of datasets is independent in all time, but in real world, the feature of intrusion dataset is complex and needed a comprehensive analysis.

Taking above discussions into consideration, this paper proposes the KDSVM model using the k-means algorithm to capture the feature of raw data and divide dataset into different subsets. Then each subset is fed to the improved DNN which

top layer instead of SVM model, respectively, and learning different characteristics of the sub dataset. Next, these tested datasets are divided by prior cluster center of training dataset into sub testing datasets. Finally these testing sub datasets are fed to the trained DNN for intrusion detection. Because the DNN can acquire enough information, via prior learning processing and capture more specific rules of attack types in networks based on extracting feature capability for massive and complex data [6, 13]. The DNN model based on theory of deep learning, can solve non-linear problems with complex and large-scale data, and has been successfully applied in the area of weather forecasting and stock prediction [10]. The experimental results based on the knowledge of KDD CUP99 datasets and NLS-KDD datasets [22] show that KDSVM generates better accuracy and more robust than other well-known algorithms, and well supported for parallel computing.

The rest of the paper is organized as follows. The related literature concerning of IDS is reviewed in the Sect. 2. Section 3 presents the proposed approach in detailed and describes it works. Section 4 describes the experimental datasets and illustrates the data preparation, evaluation criteria, results and discussions of experiments. Finally, the conclusions and suggestions for future work are provided in Sect. 5.

2 Literature Review

In this section, the deep learning approach of deep neural network is briefly introduced. As a matter of fact, IDS as classification method is very important for deal with feature of dataset, because the categories learner has acquired knowledge and patterns based on the characteristic of data. Additionally, the level of feature representation is determining the performance of a learner.

2.1 DNN Algorithm

The essence of the deep neural network is to learn more useful feature of machine learning and construct multiple layers in network and vast amounts of training data.

Auto-encoder: An auto-encoder is one type of unsupervised neural networks with three layers [12] and the output target of the auto-encoder is input data. The encoder network transforms the input data from a higher dimensional space to codes in a low dimensional space and the decoder network remodels the inputs from the previous works.

The encoder network is defined as an encoding function denoted by $f_{encoder}$. This function indicates the encoding process:

$$h^m = f_{encoder}(x^m) \tag{1}$$

In which x^m stands for data point from a dataset, h^m is the encoding vector obtained from x^m.

Decoder: The decoder network is defined as a reconstruction function denoted as $f_{decoder}$, this function indicates the decoding process:

$$\hat{x}^m = f_{decoder}(h^m) \tag{2}$$

In which $\hat{x}^m$ is the decoding vector obtained from h^m. There are specific algorithms for several encoding functions and reconstruction functions including:

$$\text{Logsig:} \quad f_{encoder}(x^m) = \frac{1}{1+e^{-x^m}} \tag{3}$$

$$\text{Satline:} \quad f_{encoder}(x^m) = \begin{cases} 0 & if \quad x^m \le 0 \\ z & if \quad 0 < x^m < 1 \\ 1 & if \quad x^m \le 0 \end{cases} \tag{4}$$

$$\text{Pureline:} \quad f_{encoder}(x^m) = x^m \tag{5}$$

Pre-training: The process is proceeding in the sequence until the *N*th auto-encoder is trained for initialization the final hidden layer of the DNN. In this way, all the hidden layers of DNN are stored auto-encoder by stacked structure in each training *N* times, and are regarded as pre-trained. This pre-training process is proven to be significantly better than random initialization of the DNN and conducive to achieving generalization in classification [9, 11].

Fine-tuning: Fine-tuning is the process that utilizes the supervised fashion to improve the performance of DNN. The network is retraining and labeled from training data, and the errors by difference between real and predicted values are back propagation with stochastic gradient descent (SGD) method for all multilayer network. The equation of SGD is defined as follows:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - t_i)^2 \tag{6}$$

where, the function E is loss function, y is the real label and t is the output of network. The gradient of weight parameter ω is obtained by derivative the error equation.

$$\frac{\partial \mathrm{y}}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \mu_j} \cdot \frac{\partial \mu_j}{\partial \omega_{ij}} \tag{7}$$

With the gradient of the ω_{ij} the equation of updated SGD is defined as:

$$\omega_{ij}^{new} = \omega_{ij}^{old} - \eta \cdot (y_j - t_j) \cdot y_j(1 - y_j) \cdot h_i \tag{8}$$

In which, the η is the step size and greater zero, h is the hidden layer number in the deep network [4].

This process is tuned and optimized by the weight and threshold based on the real label data in the DNNs, in this way, the deep networks can learn important knowledge for final output and direct the parameter of whole network to detect correct classification [20].

3 Proposed Approach of KDSVM

This section, the proposed approach is used based on clustering methods and deep learning with SVM model to solve above problems. In the first place, the sub training datasets divided the training process into different subsets and calculate center points by each train points. Second, the sub train datasets are trained by *k*th DNNs, the number *k* is the value of clusters, this take DNNs that have learned different characteristic of each cluster centers. Third, the sub testing datasets are divided from the test datasets by k-means algorithm that uses the previous cluster centers in the first step, and these sub testing datasets are applied to detect intrusion attack type by completely trained per DNN which top layer used SVM classifier. Finally, the outputs of every DNN are aggregated for the final results of intrusion detection classifiers.

3.1 The KDSVM Algorithm

The approach in detail is showed the algorithm of KDSVM. The point center and training sets are generated by output of k-means function in line 1, the sub testing sets are obtained by calculating the distance with Huffman function in line 2–6, the *k*th DNNs is trained by training set in line 7–12, the sub testing sets are index and the final results are predicted by the aggregation in line 13–19.

Algorithm 1: KDSVM algorithm

Input: TR-Train Dataset, TE-Test Dataset, K- the number of the cluster, HLN- the number of the hidden layer node, HL- the number of the hidden layer, TSVM-top layer classifier of architecture in each DNN.

Output: classification result- for KDSVM model

/* get the center points and sub-train dataset */

C, subTR ← kmeans (TR, K)

/* calculate distance from each data points in TE to center points */

For $i = 1$ to N

  For $j = 1$ to K

    distance (i, j) = huffman (TE_i, C_j)

  End

End /* Train the every deep neural network by the each sub-train dataset */

Switch p do

  case p do

    Train the DNN_p =DNN(subTRp, HLN,HL);

    Train the $TSVM_p$=DNN_p

  End

End /* get sub-test dataset by the per cluster center Cj*/

For m=1 to N

  index ← find (min($distance\ (i)$)) ; $subTE_{index} \leftarrow TE_i$

End /* train DNN model for each subTR and test DNN model for subTE */

For m=1 to K

  $prediction_m = model_m(subTE_m)$

End /* aggregate each prediction result */

Return classification result = aggregate (prediction)

4 Experiments

The experiments will be examined and compared with other detection engineer models, for instance, SVM, BPNN, DBN-SVM and naive Bayes. The six datasets from the KDDCUP99 and NSL-KDD are used to evaluate the performance of all models. Then, the parameters of the number of clusters and the weights of DNN are discussed and analyzed.

4.1 The Dataset

In this research, six datasets are randomly generated from two datasets, KDD CUP'99 and NSL-KDD, which reduce the amount of data, and called Dataset1 to Dataset6, respectively [18] and show in Table 1.

Table 1 The distribution of training set and testing set are shown in six dataset from the KDD'99 and NSL-KDD

Data set	Training dataset					Testing dataset				
	Normal	Dos	Prob.	U2R	R2L	Normal	Dos	Prob.	U2R	R2L
Dataset1	9727	39145	4107	52	1126	60593	229853	4166	228	16189
Dataset2	48639	195729	4107	52	1126	60593	229853	4166	228	16189
Dataset3	97278	391458	4107	52	1126	60593	229853	4166	228	16189
Dataset4	13449	9234	2289	11	209	9711	7458	2421	200	2754
Dataset5	33671	22963	11656	52	995	9711	7458	2421	200	2754
Dataset6	13449	9234	2289	11	209	2152	4342	2402	200	2754

The six new datasets are used to evaluate the performance of KDSVM algorithm, and execute to compare the other detection engineering methods, such as SVM, BPNN, DNB-SVM, and Naive Bayes [14].

4.2 Evaluation Methods

In this study, the Accuracy, Recall, and Error Rate (ER) are used to evaluate the performance of the detection models. The formulas of above criteria are calculated as follows [17]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

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

$$\text{Error Rate} = \frac{FP + FN}{FP + TP + TN + FN} \tag{11}$$

In which, True Positives (TP) indicates the number of network attack types distinguishing correct cases, the True Negatives (TN) shows the number of normal network type classifying the correct normal type, the False Negative (FN) is denotes the number of classified attack type detection as normal type, the False Positive (FP) means that the number of classified normal type as attack cases. The step of Accuracy shows the degree of whole correct detection accuracy of dataset and the ER refers to robust of classifier, the Recall indicates the degree of correctly detection attack type in whole attack recodes. In above team, higher accuracy and recall and lower ER is represented good performance.

4.3 Experiments with KDSVM

In this section, cluster number of k is evaluation of KDSVM based on the six dataset, because the area of value of k are different in each dataset and this are serious impact precision of results for KDSVM method. Next, the testing datasets are used to compare the performance of the five models.

Results and Comparisons

In this section, the fusion matrix and the evaluated criterion are calculated with the KDSVM and other four traditional detection engineers in the six datasets respectively. The experiment results of above algorithm in six datasets are shown in Table 2 and Fig. 1.

Table 2 The comparing the results for the intrusion network for six datasets (\%)

Dataset	k	model	Normal	Dos	Probe	U2R	R2L	Acc	Recall	ER
Dataset1	–	SVM	98.21	83	66.01	0.88	3.14	81.52	77.72	18.48
	–	BPNN	96.51	89.49	46.18	9.21	1.93	85.66	83.48	14.34
	–	DBN-SVM	93.65	96.62	59.27	0	0	90.44	91.08	9.56
	–	Bayes	91.51	95.59	61.35	4.39	3.56	89.48	**92.57**	10.52
	2	KDSVM	97.21	96.87	80.32	11.4	6.88	**91.97**	91.68	**8.03**
Dataset2	–	SVM	96.22	97.1	65.84	0	0.05	91.39	90.52	8.61
	–	BPNN	91.44	97.42	62.69	7.02	5.41	90.93	92.88	9.07
	–	DBN-SVM	98.23	96.48	38.26	0	0	90.95	89.51	9.05
	–	Bayes	95.92	95.98	62.55	4.82	4.38	90.69	91.07	9.31
	4	KDSVM	98.42	97.2	70.64	3.51	1.57	**92.03**	**91.35**	**7.97**
Dataset3	–	SVM	95.87	97.23	64.86	0	0.06	91.41	90.59	8.59
	–	BPNN	81.53	96.95	8.81	6.14	7.26	88.03	90.05	11.97
	–	DBN-SVM	99.57	96.57	0	0	0	90.76	89.37	9.24
	–	Bayes	96.38	96.29	59.15	7.02	7.46	91.12	90.95	8.88
	5	KDSVM	97.61	97.23	65.96	4.39	6.59	**92.1**	**92.23**	**7.9**
Dataset4	–	SVM	95.54	70.18	57.37	0	1.63	70.73	53.26	29.27
	–	BPNN	96.35	71.17	65.55	0	0.58	72.16	**57.79**	27.84
	–	DBN-SVM	99.63	63.11	7.23	0	0	64.57	40.45	35.43
	–	Bayes	93.9	72.18	41.02	0	0	68.73	52.78	31.27
	3	KDSVM	96.17	75.84	53.37	3	3.01	**72.64**	57.48	**27.36**
Dataset5	–	SVM	98.57	18.93	49.89	0	0.11	54.1	20.45	45.9
	–	BPNN	91.79	7.63	66.58	1.5	2.43	49.53	27.56	50.47
	–	DBN-SVM	99.69	62.64	48.99	0	0	68.93	46.43	31.07
	–	Bayes	99.06	61.65	35.4	0	0	66.87	44.28	33.13
	3	KDSVM	97.19	74.51	48.37	5	0.62	**71.83**	**55.08**	**28.17**
Dataset6	–	SVM	95.81	41.5	43.67	0	0	41.46	30.6	58.54
	–	BPNN	74.72	4.61	88.67	0	1.53	33.59	30.6	66.41
	–	DBN-SVM	99.72	36.15	6.74	0	0	32.73	18.9	67.27
	–	Bayes	82.16	48.25	28.52	0	0	38.37	30.08	61.63
	5	KDSVM	84.2	50.02	52.66	1.5	0.98	**44.55**	**37.85**	**55.45**

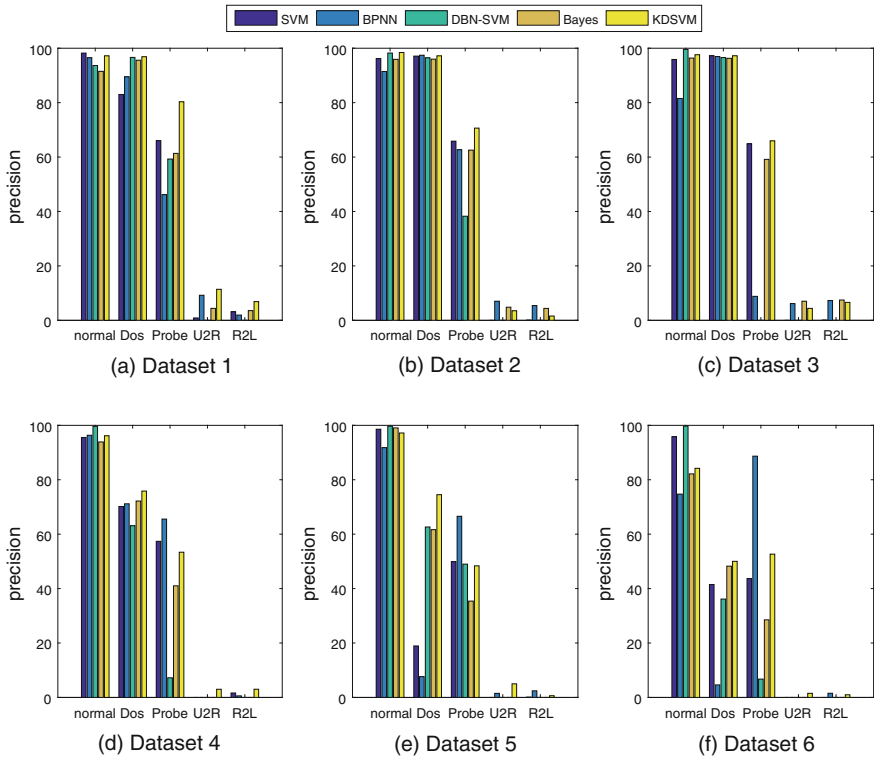

Fig. 1 The prediction accuracies histogram of five types for models of SVM, BPNN, DBN-SVM, Bayes and KDSVM are compared in six datasets in different colors

In which, the columns symbol of ACC in table heads mean the average accuracy for each models. The records are unbalance in six dataset, the types of Normal and Dos have major compositions, the U2R and R2L have sparse distribution, because the last two cases have especially intrusion actions which have obtained advanced user right, it is more covert intrusion for difficultly detection.

From Table 2 and Fig. 1, consideration accuracy, the KDSVM has better accuracy than other four methods, and has the lowest error rates in the datasets.

4.4 Discussion

The overall accuracy is used to generate the histogram and compare distinguished results in six datasets and shown in Fig. 2. This is more detailed to evaluate the classification performance with five types (one normal and four types).

From the above, the results show that KDSVM algorithm is good at detection cases of Normal, Dos and Probe in the six dataset. Therefore, for sparse and difficult

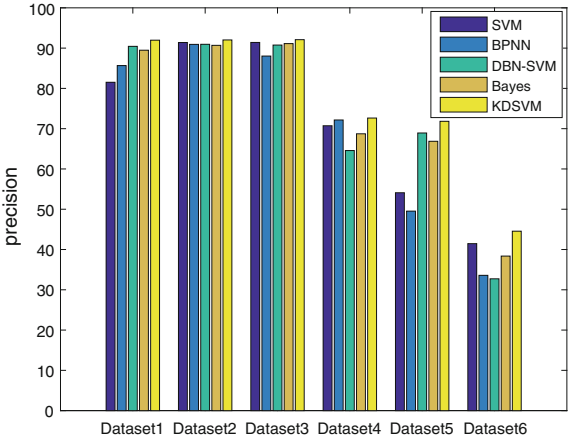

Fig. 2 The histograms of the average precision by the five models are compared with the six datasets

cases of U2R and R2L in six datasets, the KDSVM model also obtains higher accuracy.

5 Conclusion

The attacking events of low frequent are usually difficult to predict and it can cause severe threats to networks. This paper puts forward the innovative approach which takes the advantage of k-means and hybrid deep neural network with top layer used SVM classifier, to detect attack types. In the first stage, the features of the network dataset are clustering and divided into k sub datasets in a bid to find more knowledge and patterns from the similar clusters. Then in the second stage, the highly abstract information is obtained by deep learning networks from the subsets during the clustering process. Finally, the DNNs which used SVM classifier to instead of softmax layer, are used to detect the attack cases with testing subsets. This is an efficient way to improve the accuracy of the detection rates. The results of experiment show that the KDSVM performs better than the SVM, BPNN, DBN-SVM and Bayes with best accuracy over the six datasets. On the other hand, the proposed algorithm is more capable of classifying term of sparse attack cases and effectively improves detection accuracy in real security system. However, limitations of the KDSVM include the DNN parameters of weights and threshold of the every layer, and the SVM parameters that need to be optimized by heuristic algorithms, and it will be study works in the further.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 11361046) and the Key Research Fund of Ningxia Normal University (Grant No. NXSFZD1517 NXSFZD1603 and NXSFZD1608), the Natural Science Fund of Ningxia Province (Grant NZ16260) and the Fundamental Research Fund for Senior School of Ningxia Province (Grant No. NGY2015124).

References

1. Aburomman, A.A., Reaz, M.B.I.: A novel svm-knn-pso ensemble method for intrusion detection system. Applied Soft Computing 38, 360–372 (2016)
2. Alom, M.Z., Bontupalli, V., Taha, T.M.: Intrusion detection using deep belief networks. In: 2015 National Aerospace and Electronics Conference (NAECON). pp. 339–344. IEEE (2015)
3. Barbara, D., Wu, N., Jajodia, S.: Detecting novel network intrusions using bayes estimators. In: SDM. pp. 1–17. SIAM (2011)
4. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. Neural Networks, IEEE Transactions on 5(2), 157–166 (1994)
5. Chen, W.H., Hsu, S.H., Shen, H.P.: Application of svm and ann for intrusion detection. Computers & Operations Research 32(10), 2617–2634 (2005)
6. Chilimbi, T., Suzue, Y., Apacible, J., Kalyanaraman, K.: Project adam: Building an efficient and scalable deep learning training system. In: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14). pp. 571–582 (2014)
7. Denning, D.E.: An intrusion-detection model. Software Engineering, IEEE Transactions on SE-13(2), 222–232 (1987)
8. Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.N.: Data mining for network intrusion detection. In: Proc. NSF Workshop on Next Generation Data Mining. pp. 21–30 (2002)
9. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? The Journal of Machine Learning Research 11, 625–660 (2010)
10. Grover, A., Kapoor, A., Horvitz, E.: A deep hybrid model for weather forecasting. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 379–386. ACM (2015)
11. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation 18(7), 1527–1554 (2006)
12. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and helmholtz free energy. Advances in neural information processing systems pp. 3–3 (1994)
13. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using click through data. In: Proceedings of the 22nd ACM international Conference on information & knowledge management. pp. 2333–2338. ACM (2013)
14. Japkowicz, N., Shah, M.: Evaluating learning algorithms: a classification perspective. Cambridge University Press (2011)
15. Kabiri, P., Ghorbani, A.A.: Research on intrusion detection and response: A survey. IJ Network Security 1(2), 84–102 (2005)
16. Karami, A., Guerrero-Zapata, M.: A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks. Neurocomputing 149, 1253–1269 (2015)
17. Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I.: A hierarchical som-based intrusion detection system. Engineering Applications of Artificial Intelligence 20(4), 439–451 (2007)
18. Koc, L., Mazzuchi, T.A., Sarkani, S.: A network intrusion detection system based on a hidden naive bayes multiclass classifier. Expert Systems with Applications 39(18), 13492–13500 (2012)
19. Marin, G.: Network security basics. Security & Privacy, IEEE 3(6), 68–72 (2005)
20. Palm, R.B.: Prediction as a candidate for learning deep hierarchical models of data. Technical University of Denmark (2012)
21. Salama, M.A., Eid, H.F., Ramadan, R.A., Darwish, A., Hassanien, A.E.: Hybrid intelligent intrusion detection scheme. In: Soft computing in industrial applications, pp. 293–303. Springer (2011)

22. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009 (2009)
23. Zhang, J., Zulkernine, M., Haque, A.: Random-forests-based network intrusion detection systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 38(5), 649–659 (2008)