Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 806954, 8 pages
http://dx.doi.org/10.1155/2015/806954

Research Article

Feature Selection Using Particle Swarm Optimization in Intrusion Detection

Iftikhar Ahmad

Department of Software Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia

Correspondence should be addressed to Iftikhar Ahmad; iwattoo@ksu.edu.sa

Received 15 May 2015; Accepted 6 August 2015

Academic Editor: Athanasios V. Vasilakos

The prevention of intrusion in networks is decisive and an intrusion detection system is extremely desirable with potent intrusion detection mechanism. Excessive work is done on intrusion detection systems but still these are not powerful due to high number of false alarms. One of the leading causes of false alarms is due to the usage of a raw dataset that contains redundancy. To resolve this issue, feature selection is necessary which can improve intrusion detection performance. Latterly, principal component analysis (PCA) has been used for feature reduction and subset selection in which features are primarily projected into a principal space and then features are elected based on their eigenvalues, but the features with the highest eigenvalues may not have the guaranty to provide optimal sensitivity for the classifier. To avoid this problem, an optimization method is required. Evolutionary optimization approach like genetic algorithm (GA) has been used to search the most discriminative subset of transformed features. The particle swarm optimization (PSO) is another optimization approach based on the behavioral study of animals/birds. Therefore, in this paper a feature subset selection based on PSO is proposed which provides better performance as compared to GA.

1. Introduction

Wireless sensor networks (WSNs) are vulnerable to various sorts of security threats due to the following reasons: scalable, scattered, dynamic, fault tolerant, and weak infrastructure in nature. So, these networks can easily be targeted to various security threats. So, in this paper, the focus is on proposing efficient and effective feature selection method and classification architecture for intrusion in WSNs. Classification is a core part in intrusion detection systems, which aims to classify each activity of the system into normal or intrusive. The feature space of a classification problem is a significant factor that affects the performance of an intrusive analysis engine or a classifier. Further, it is hard to determine which features are valuable because a dataset consists of several features like relevant, irrelevant, and redundant. Irrelevant and redundant features are not useful for classification because these can affect the relevant features and confuse the classifier or intrusive analysis engine. Therefore, feature selection is imperative to improve the quality of the feature space, reduce the number of features, and enhance the classifier performance [1, 2].

So the feature selection is a substantial problem in intrusion detection. In the past, a lot of work had been done on intrusion classification and feature extraction but the issue of feature selection was not addressed seriously. Therefore, optimal feature subset is deemed necessary to improve the classifier performance. To overcome this issue, a variety of search techniques have been applied to feature selection. Still, existing methods suffer several problems like complexity of classifier architecture, higher memory usage, and high computational cost. In recent approaches, PCA has been used for feature selection in which features are selected on some percentage of the top principal components. But there is possibility to miss several important features and to include irrelevant features in feature subset during this process [3]. This process selected those features which had highest eigenvalues and ignored those features which had lowest eigenvalues. This method might be not effective due to negligence of certain features which might be more sensitive and important to the classifier.

To better address this problem, the GA is applied to search the principal space for the feature subset selection. This method outperformed previous approaches but GA has

minor weaknesses like inability to discover global optimum and incapability of solving variant problems. To cope with this problem, particle swarm optimization (PSO) is proposed and implemented for optimal feature selection. PSO is an effective and efficient global search technique [4, 5]. It is an appropriate algorithm to address feature selection problems due to better representation, capability of searching large spaces, being less expensive computationally, being easier to implement, and fewer parameters being required.

The paper is organized as follows. The related work to intrusion detection in WSNs is discussed in Section 2. Section 3 describes proposed model which consists of different phases: preprocessing, feature selection, classification, training, testing, and evaluation. Section 4 concludes the paper.

2. Literature Review

The performance of intrusive analysis engine depends on accurate dataset in WSNs. The dataset consists of instances which are described by features or attributes. Generally, useful features are unknown in advance and irrelevant and redundant features may affect the classification performance in WSNs. In order to remove unnecessary features, several techniques have been used in various domains. A brief overview of those approaches is presented.

In [6], a theoretical method of feature selection is proposed for intrusion detection. This method proposed PCA for feature transformation and PSO for feature selection. The SVM is suggested for the classification purpose on a standard KDD cup dataset. This method is further explored in this work using modular neural network on NSL KDD dataset. In [7], the authors provided a detailed investigation on the method of the energy-efficient sensors scheduling in WSN. So, the feature selection approaches are useful for target classification to minimize energy consumption. Ganapathy et al. [8] presented a survey of several feature selection and classification approaches in intrusion detection. Further, they proposed their own feature selection algorithm as well as classifier based on multiclass Support Vector Machine (SVM). In addition, they also addressed research challenges and highlighted potential future research directions in intrusion detection using soft computing techniques.

In [9], the problem of Gaussian-distributed WSN in intrusion detection is analysed under scenarios of single and multiple sensing detection. They discussed in detail various effects of different network parameters in intrusion detection. In addition, they formulated detection probability and provided guidelines for suitable deployment strategy and determining critical network parameters. In [10], a problem to classify the WSN parameters is that node density and sensing range were discussed in terms of a desirable detection probability. The detection probability was based on two models such as homogeneous and heterogeneous WSN. Further, they tested their experimental results for both homogeneous and heterogeneous WSNs. In [11], a detailed comparison of several intrusion detection approaches based on swarm intelligence is presented. The main focus was on exploring the efficiency of each approach in the area of intrusion detection. The swarm intelligence techniques are attracting researchers working in the field of intrusion detection due to their excellent characteristics.

In [12], a rule based feature selection algorithm is proposed to remove redundant attributes and to select sensitive feature set that is valuable for intrusive analysis engine in wireless sensor networks. The focus was only denial of service attacks. Further, multiclass SVM is extended for the improvement of classification accuracy. In [13], a monitoring technique was proposed for intrusion detection in Wireless Mesh Networks (WMN) which is based on two classes: traffic agnostic and resourceful and traffic aware and resourceful. The demonstrated results indicate optimal performance in intrusion detection rate and resource consumption in WMN.

In [14], a new framework for intrusion detection is proposed in cluster-based wireless sensor networks. SVM is used for classification because they addressed two-class problem in this work. They used the metrics detection rate, false positive rate, energy consumption, and efficiency to validate their model. In [15], Fawzy et al. addressed outlier detection problem in wireless sensor networks and they proposed outlier detection and classification mechanism in sensor network. The results showed improvement in the classification process.

In [16], a feature selection mechanism was proposed which was based on custom feature preprocessing. This method works on variance. The features with higher variance are selected and features with less variance were ignored. This process may miss many important features. In [17], a features selection algorithm is proposed based on record to record travel and Support Vector Machine is applied for the classification. They used KDD dataset for their experiments. The detection rate, accuracy, and false alarms were used to measure the performance of their mode.

In [18], De La Hoz Franco et al. used Fisher discriminant rate algorithm for feature selection and self-organizing maps used for classification. The results showed sensitivity of 97.39% and specificity of 62.73% with 17 features of the dataset. In [19], Alsharafat proposed artificial neural network and extended classifier for intrusion detection. She applied neural network for feature subset selection and extended classifier to classify normal and intrusive activities. The KDD dataset was used for experiments in this work. In [20], feature-selection method had been proposed based on the cuttlefish optimization in intrusion detection. Decision tree (DT) was applied for classification purposes. The proposed method improved performance in terms of detection rate and accuracy rate and reduces the false alarm rate.

The abovementioned discussion highlights the importance of feature selection and classification in intrusion detection in wireless sensor networks. Various methods have been proposed for feature selection but still suffer several issues as mentioned in the above discussion. Therefore, an effective and efficient method is necessary which may enhance the performance of intrusion detection system in wireless sensor networks. Thus, an optimal method of feature selection based on PSO is proposed and its performance is compared to GA which is a baseline in this work.

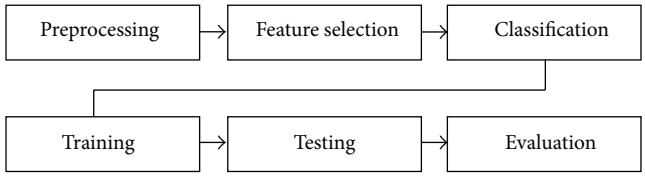

Figure 1: Proposed model for feature selection and evaluation.

3. Proposed Model

The proposed model consists of six phases: preprocessing, feature selection, classification, training, testing, and evaluation of results which is shown in Figure 1. This model uses NSL KDD dataset for the experiments in this work. This dataset is a standard, which is considered as a benchmark for evaluating security detection mechanisms. Another reason is that it is difficult to get another dataset which contains such richness and variety of attacks as NSL KDD includes. Further, it is a refined form of KDD cup dataset. The details of proposed model with each of its phases are described in the following subsections.

3.1. Preprocessing. Preprocessing of raw features is necessary because the raw features confuse the classifier which results in false alarms. In addition, few symbolic features increase computational and memory resources and are unexploited for the classification techniques. The raw feature set from the NSL KDD dataset is expressed by

$$rf = \{rx_1, rx_2, rx_3, rx_4, \ldots, rx_l\}, \quad (1)$$

where $l = 41$, which indicates that there are forty-one features in raw dataset. The symbolic features are discarded from the raw feature set because these features increase overheads without any benefits in learning process. The resultant raw feature set is expressed by

$$rf' = \{rx_1, rx_2, rx_3, rx_4, \ldots, rx_m\}, \quad (2)$$

where $m = 38$, which indicates number of raw features. Additional preprocessing is required on this raw feature set so that prominent features are selected based on their sensitivity. So, for this purpose, feature transformation and prominent features selection are deemed necessary which are discussed in detail in Feature Selection.

3.2. Feature Selection. The purpose of feature selection is to determine the minimum number of feature subsets that is essential and appropriate for the classifier to classify the connection or activity into normal or intrusive. The feature subset p is always less than original feature set m. The flow of feature subset selection is shown in Figure 2. The details of each step in Figure 2 are discussed in the proceeding sections.

3.2.1. Feature Transformation Using PCA. PCA is a statistical method normally used for data analysis and is a very useful method of feature selection. The PCA is applied to transform raw features into principal features so that the features are more clearly visible and their importance is visualized. This technique has been used from last few years in different domains [21]. In this technique, the features are selected on the basis of eigenvalues, the features with higher eigenvalues are selected and the features with lower eigenvalues are ignored. This method of feature selection is not an optimal mode due the probability of losing some important features. Which feature is important and which is not important? Which feature is selected and which is not in principal space? This is an optimization problem and PSO and GA are the best methods; those offer proven capability of solving optimization problems. So, in this work, the PCA was used for feature transformation only and the feature selection was done through PSO and GA. The flow of PCA for feature transformation is shown in Figure 3. The applied PCA algorithm is described.

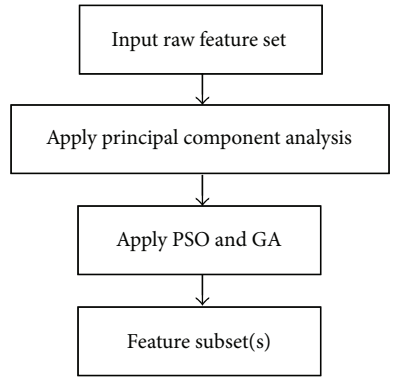

Figure 2: The flow of feature subset selection.

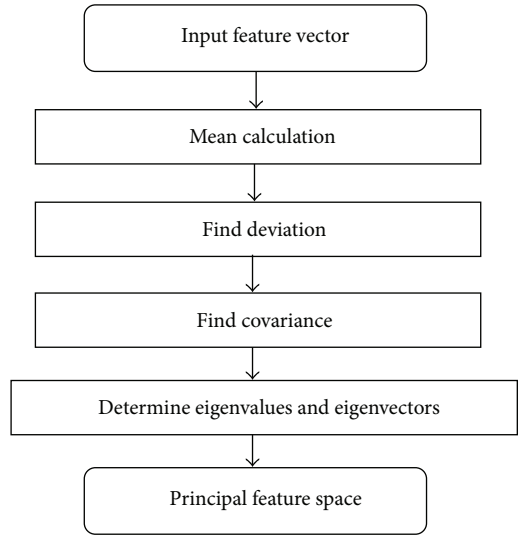

Figure 3: The flow of PCA for feature transformation.

PCA Algorithm. Suppose $y = \{rx_1, rx_2, rx_3, rx_4, \ldots, rx_m\}$ are n vectors, where $m = 38$.

Step 1. Calculate mean:

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{3}$$

Step 2. Find deviation.

Subtract the mean: $D_i = (y_i - \overline{y})$, where $i = 1, 2, \ldots, n$.

Step 3. Find covariance matrix Q.

From the matrix $B = [D_1, D_2, D_3, \ldots, D_N]$ ($N * M$ Matrix), compute Q:

$$Q = \frac{1}{M}\sum_{N=1}^{M} D_N, \quad D_N = BB^T. \tag{4}$$

Step 4. Compute the eigenvalues of Q: $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_N$:

$$Q = \sum_{i=1}^{N} \lambda_i. \tag{5}$$

Step 5. Compute the eigenvectors of Q: $\mu_1, \mu_2, \mu_3, \ldots, \mu_N$.

Vector y can be written as a linear combination of the eigenvectors as

$$y = \sum_{i=1}^{N} b_i \mu_i. \tag{6}$$

3.2.2. Feature Subset Selecting Using PSO. The PSO is a population based technique developed by Eberhat and Kennedy [22]. PSO is a successful and valued global search technique [1]. It is a suitable algorithm to address feature selection problems due the following reasons: easy encoding of feature, global search facility, being reasonable computationally, less parameters, and easier implementation [2]. The PSO is applied for feature selection due to the aforementioned reasons. The PSO flow for feature selection is shown in Figure 4. The principal space is the search space through which a subset of principal components or principal features were explored and selected via PSO. In PSO, the particles represent candidate solutions in the search space particles and form a population which is also known as a swarm. The swarm of particle is generated by distributing 1 s and 0 s randomly. For every particle, if the principal component is 1, it is selected and the principal component with 0 is ignored. Thus, every particle indicates a different subset of principal components. The particles swarm is initialized randomly and then it moved in the search space or principal space to search the optimal subset of features by updating its position and velocity. The current position of particle i and its velocity are expressed in (7) and (8):

$$x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}, \tag{7}$$

where D is the dimension of the principal search space,

$$v_i = \{v_{i1}, v_{i2}, \ldots, v_{iD}\}. \tag{8}$$

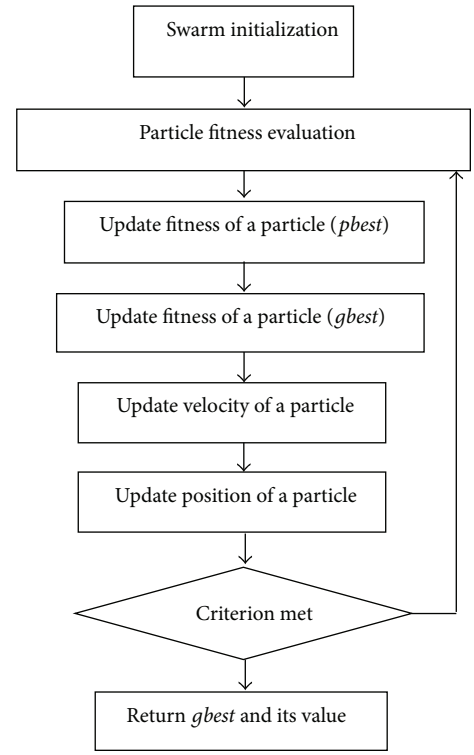

Figure 4: The flow of PSO for feature selection.

The velocity and position of the particle i are calculated by (9)

$$\begin{aligned} v_{id}^{t+1} &= \omega * v_{id}^{t} + c_1 * r_{1i} * \left(p_{id} - x_{id}^{t}\right) + c_2 * r_{2i} \\ &\quad * \left(p_{gd} - x_{id}^{t}\right), \\ x_{id}^{t+1} &= x_{id}^{t} + v_{id}^{t+1}, \end{aligned} \tag{9}$$

where t denotes the tth iteration in the process and d denotes the dth dimension in the search space. w is inertia weight and c_1 and c_2 are acceleration constants. r_{1i} and r_{2i} are random values uniformly distributed in $[0, 1]$. p_{id} and p_{gd} represent the elements of *pbest* and *gbest* in the dth dimension.

The position and velocity values of each particle are continuously updated to search for the best set of features until stopping criterion is met which can be a maximum number of iterations or a satisfactory fitness value. The applied PSO algorithm is described.

PSO Algorithm

Step 1 (swarm initialization). Randomly initialize the *position* and *velocity* of each particle.

Step 2 (particle fitness evaluation)

if fitness of $x_i > pbest_i$

$pbest_i = x_i$

if fitness of $pbest_i > gbest_i$

$gbest_i = pbest_i$

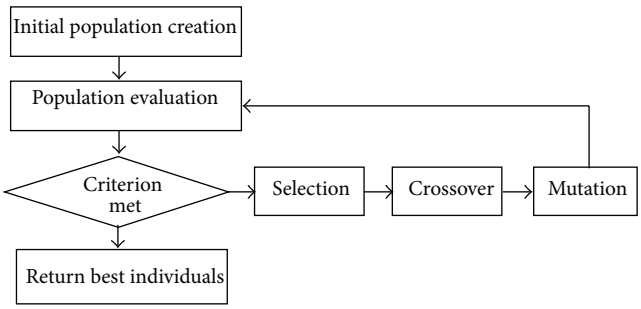

Figure 5: The flow of GA for feature selection.

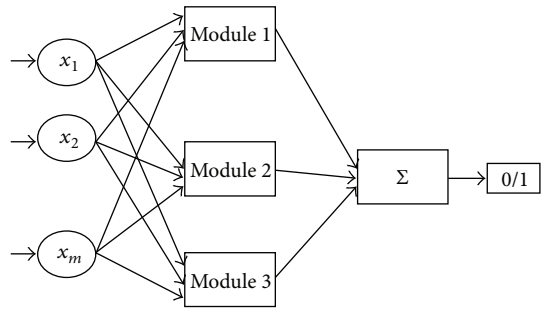

Figure 6: The general architecture of MNN.

Step 3. Update the velocity of particle i

$$v_{id}^{t+1} = \omega * v_{id}^{t} + c_1 * r_{1i} * \left(p_{id} - x_{id}^{t}\right) + c_2 * r_{2i} * \left(p_{gd} - x_{id}^{t}\right). \quad (10)$$

Update the position of particle i

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}. \quad (11)$$

Step 4. If stopping criterion is not met, continue Steps 2 and 3.

Step 5. Return *gbest* and its *fitness values*.

3.2.3. Feature Subset Selecting Using GA. Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Genetic algorithms (GAs) are search algorithms based on the principles of natural selection and genetics [23]. In GAs, a population of chromosomes indicates candidate solutions of the problem. Each chromosome is represented with fixed-length bits. The primary population of chromosomes is created by distributing 1 s and 0 s arbitrarily. This distribution was based on average assignment of 1 s and 0 s. In this encoding scheme, every chromosome is a bit of strings (1 s and 0 s) whose length is calculated by the number of principal components in the principal space. The bit with 1 is selected and bit with 0 is not selected in this encoding scheme. Each chromosome indicates a candidate solution or a subset of principal components. The population grows by searching the optimal solution using genetic operators. The GAs have two major problems of local optima and being expensive computationally. The flow of GA for feature selection is shown in Figure 5. The algorithm applied is described.

GA Algorithm

Step 1. Initial population creation (n chromosomes)

Step 2. Population evaluation (fitness evaluation of each chromosomes)

Step 3

if(Criterion is not met)

{
Selection;
Crossover;
Mutation;
}

Step 4. Return best individuals.

3.3. Classification. A modular neural network (MNN) is a series of independent modules/components which form a single neural network. Each component works on input data without collaborating with others to achieve some subtask of the whole task of the network. The integration unit integrates the outputs from each module and generates its output which indicates the decision of the modular neural network. This network outperforms a single network. This network architecture is applied on selected feature set from PSO and GA. The architecture of MNN is shown in Figure 6.

This architecture consists of three layers: input layer, modular/hidden layer, and the output layer. The input layer takes input from the selected subset of features. The hidden layer takes input from the input layer and does the processing in modular fashion. The output of the hidden layer is fed to the output layer which finally decides about the input activity whether it is normal or intrusive. Output "1" indicates intrusive activity while output "0" is considered as normal.

3.4. Training. The objective of training is to reduce the variance between the output generated by the MNN and the desired output. To accomplish this objective, weights are changed and updated through some specific steps and this process is known as training. First of all, thirty thousand samples of network records are selected randomly from NLS KDD cup dataset [24]. The nominated dataset consists of 19,200 (64%) normal and 10800 (36%) intrusive ones. After that, the features of this dataset are projected into principal space to analyse their importance for the classifier. Then, GA and PSO are applied for the selection of optimal features subset. The resultant dataset is further divided into two parts: training and production datasets.

The training dataset is used to train the system. 50% of the dataset (7500), that is, 3750, is used to train the system. The training of MNN should be stopped when the system

Table 1: NSL KDD dataset statistics.

Dataset(s)	Number of records/connections network connections
Dataset	30,000 records/connections are selected randomly, in which 19200 are normal and 10800 are intrusive connections
Training	7500
Cross-validation	1,000 (20% of 7500)
Testing	1,500 (30% of 7500)
Production	22,500

has learned the task. There are no direct indicators that indicate when to stop the training process. However, there are some ways on the basis of which the training process can be stopped. These methods are explained.

The testing dataset is used to test the performance of the neural network. Once the neural network is trained the weights are then frozen, the testing dataset is fed into the neural network, and the neural network output is compared with the desired output. In this work, 30% of dataset (7500), that is, 2250, is used to test the performance of trained MNN.

Cross-validation is another method for training process. This technique controls the error on an independent set of data and stops training when this error starts to increase. The size of dataset for cross-validation is recommended: 20% for normal generalization and 40% for high generalization. This work used one thousand and five hundred (1500) datasets for cross-validation.

3.5. Testing. Once the training process is finalized then weights of the MNN are frozen and its performance is assessed. Testing the MNN involves two steps: (i) verification step and (ii) generalization step.

3.5.1. Verification Step. In verification step, network is tested against the data which are used in training process. Purpose of the verification step is to test network learning capability on the training dataset. If a network was trained effectively, outputs generated by the network will be analogous to the real outputs. This work used 30% of the training dataset (7500), that is, 1500.

3.5.2. Generalization Step. In generalization step, testing is done with dataset which is not used in training process. Goal of the generalization step is to determine generalization capability of the trained network. After training, the network only involves computation of the feed-forward phase. For this, a production dataset is used that has only input data without their labels. This research work uses a dataset of 22,500 as a production dataset. Moreover, this method is also tested on total dataset (30,000) that consists of both training dataset and production dataset. Table 1 shows statistics of the dataset used for experiments.

3.6. Evaluation. This section discusses the results of proposed method of feature selection (PSO) with its baseline method of feature selection (GA) by exploring and searching the feature space of the PCA. The MNN is applied to measure the performance of each selection method. The overall performance of each selection method is presented in Table 2. A number of experiments were conducted with datasets: raw dataset, PCA based dataset, which was obtained through conventional method, PCA + GA dataset, and PCA + PSO dataset. Firstly, the performance of MNN was tested on full featured (38) raw dataset. The MNN completes its training process in two hours and thirty-five minutes and ten seconds. The detection rate was 94.50% with 5.5% false alarms. Secondly, the MNN was trained and tested on PCA based dataset (20). Here, the network consumed two hours and twelve minutes and eighteen seconds to train itself. The detection rate was 96.60% with 3.4% false alarms. Thirdly, the network is evaluated on PCA + GA dataset (10) and its training process was completed in twenty-one minutes and fourteen seconds. In this case, MNN showed detection rate of 98.20% with 1.8% false alarms. Fourthly, PCA + PSO dataset (8) was fed to the network to check the performance of PSO based feature selection method. This time MNN trained itself in twelve minutes and twenty seconds and demonstrated its performance maximum which was 99.40% with 0.6% false alarms.

Table 2: Performance comparison of feature selection methods.

FS method	Selected features	Training time	Detection rate (%)	False alarms (%)
Raw	38	2:35:10	94.50	5.5
PCA	20	2:12:18	96.60	3.4
PCA + GA	10	0:21:14	98.20	1.8
PCA + PSO	8	0:12:20	99.40	0.6

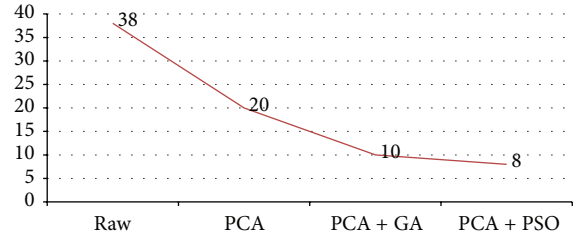

Figure 7: Comparison of feature selection techniques based on number of features.

Figure 7 shows comparison of feature selection techniques based on number of features. The raw datasets have 38 features, PCA conventional based method has 20 features in its dataset, the numbers of features are 10 in the dataset obtained through PCA + GA based selection method, and PCA + PSO based method selected eight features from the raw dataset. Thus, the last method provided a smaller subset of features which improve the performance of the classifier and minimize its architecture's complexity.

Figure 8 demonstrates the comparison of feature selection techniques based on detection rate. The obtained feature subsets outperform a raw feature set. Further, PSO based

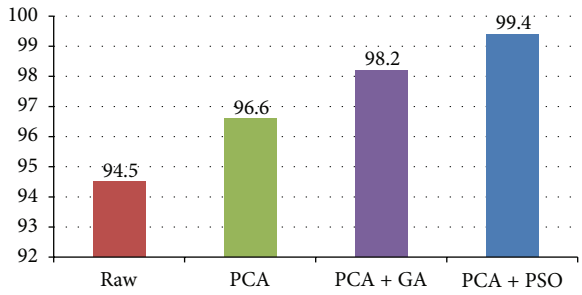

Figure 8: Comparison of feature selection techniques based on detection rate.

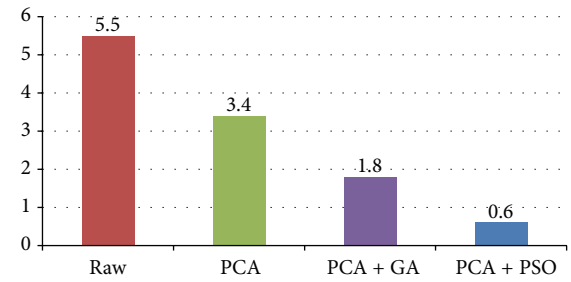

Figure 9: Comparison of feature selection techniques based on false alarm rate.

feature selection indicates best performance with minimum number of features.

Figure 9 illustrates comparison of false alarm rate among the feature selection techniques. The PSO based method of feature selection shows less false alarm rate as compared to other methods.

4. Conclusion

Wireless sensor networks (WSNs) are unprotected due to their nature and are more exposed to various security threats. Therefore, a powerful security mechanism is needed to overcome such threats. Several intrusion detection techniques are available but the problem is their performance. The performance can be improved by proposing suitable method of feature selection and classification. Thus, in this paper, a method of feature selection in intrusion detection for wireless sensor network is proposed which is based on PSO and selects optimal subset of features from the principal space or the PCA space. The performance of the proposed method is tested on NSL KDD dataset which is considered standard dataset for the evaluation of intrusion detection methods. The selected feature subset (PSO based) is validated on modular neural network and compared with feature subset (GA based) and other methods (PCA based, raw). The results indicate that PSO based feature selection method outperforms the existing methods.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The author extends his appreciation to the Research Center at CCIS and Deanship of Scientific Research at King Saud University for funding the work through Project no. RC131022.

References

[1] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[2] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms," *Applied Soft Computing Journal*, vol. 18, pp. 261–276, 2014.

[3] I. Ahmad, "Enhancing MLP performance in intrusion detection using optimal feature subset selection based on genetic principal components," *Applied Mathematics & Information Sciences*, vol. 8, no. 2, pp. 639–649, 2014.

[4] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, Wiley, 2nd edition, 2007.

[5] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Evolutionary Computation Series, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 2001.

[6] I. Ahmad and F. Amin, "Towards feature subset selection in intrusion detection," in *Proceedings of the IEEE 7th Joint International Information Technology and Artificial Intelligence Conference (ITAIC '14)*, pp. 68–73, Chongqing, China, December 2014.

[7] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Networks and Applications*, vol. 11, no. 5, pp. 723–740, 2006.

[8] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, L. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey," *Eurasip Journal on Wireless Communications and Networking*, vol. 2013, article 271, 2013.

[9] Y. Wang, W. Fu, and D. P. Agrawal, "Gaussian versus uniform distribution for intrusion detection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 342–355, 2013.

[10] Y. Wang, X. Wang, B. Xie, D. Wang, and D. P. Agrawal, "Intrusion detection in homogeneous and heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 698–710, 2008.

[11] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: a survey," *Computers & Security*, vol. 30, no. 8, pp. 625–642, 2011.

[12] K. Anand, S. Ganapathy, K. Kulothungan, P. Yogesh, and A. Kannan, "A rule based approach for attribute selection and intrusion detection in wireless sensor networks," *Procedia Engineering*, vol. 38, pp. 1658–1664, 2012.

[13] A. Hassanzadeh, A. Altaweel, and R. Stoleru, "Traffic-and-resource-aware intrusion detection in wireless mesh networks," *Ad Hoc Networks*, vol. 21, pp. 18–41, 2014.

[14] H. Sedjelmaci, S. M. Senouci, and M. Feham, "An efficient intrusion detection framework in cluster-based wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 10, pp. 1211–1224, 2013.

[15] A. Fawzy, H. M. O. Mokhtar, and O. Hegazy, "Outliers detection and classification in wireless sensor networks," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 157–164, 2013.

[16] R. R. Staudemeyer and C. C. Omlin, "Extracting salient features for network intrusion detection using machine learning methods," *South African Computer Journal*, pp. 5282–5296, 2014.

[17] Z. Othman, Z. Muda, L. M. Theng, and M. Rafique Othman, "Record to record feature selection algorithm for network intrusion detection," *International Journal of Advancements in Computing Technology*, vol. 6, no. 2, pp. 163–175, 2014.

[18] E. De La Hoz Franco, A. O. Garcia, J. O. Lopera, E. De La Hoz Correa, and F. M. Palechor, "Implementation of an intrusion detection system based on self organizing map," *Journal of Theoretical & Applied Information Technology*, vol. 71, no. 3, pp. 324–334, 2015.

[19] W. Alsharafat, "Applying artificial neural network and extended classifier system for network intrusion detection," *International Arab Journal of Information Technology*, vol. 10, no. 3, pp. 230–238, 2013.

[20] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670–2679, 2015.

[21] M. A. Livani and M. Abadi, "A PCA-based distributed approach for intrusion detection in wireless sensor networks," in *Proceedings of the International Symposium on Computer Networks and Distributed Systems (CNDS '11)*, pp. 55–60, IEEE, Tehran, Iran, February 2011.

[22] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, October 1995.

[23] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[24] The NSL-KDD Dataset, http://nsl.cs.unb.ca/NSL-KDD.