Engineering Applications of Artificial Intelligence 26 (2013) 1731–1740

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

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

GMDH-based networks for intelligent intrusion detection

Zubair A. Baig [a,*], Sadiq M. Sait [a,b], AbdulRahman Shaheen [a]

[a] Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
[b] Center for Communications and IT Research, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

ARTICLE INFO

Article history:
Received 21 April 2012
Received in revised form
12 March 2013
Accepted 18 March 2013
Available online 22 April 2013

Keywords:
Network intrusion detection
GMDH
Feature ranking
Machine learning

ABSTRACT

Network intrusion detection has been an area of rapid advancement in recent times. Similar advances in the field of intelligent computing have led to the introduction of several classification techniques for accurately identifying and differentiating network traffic into normal and anomalous. Group Method for Data Handling (GMDH) is one such supervised inductive learning approach for the synthesis of neural network models. Through this paper, we propose a GMDH-based technique for classifying network traffic into normal and anomalous. Two variants of the technique, namely, Monolithic and Ensemble-based, were tested on the KDD-99 dataset. The dataset was preprocessed and all features were ranked based on three feature ranking techniques, namely, Information Gain, Gain Ratio, and GMDH by itself. The results obtained proved that the proposed intrusion detection scheme yields high attack detection rates, nearly 98%, when compared with other intelligent classification techniques for network intrusion detection.

1. Introduction

The ever-expanding nature of Internet traffic accompanied with convenient availability of open source tools to launch malicious attacks has placed a demand for better network intrusion detection systems to detect such attacks accurately, so as to initiate subsequent countermeasures. Malicious attacks launched against an organization's computing infrastructure may cause huge financial losses. Accurate detection is an important first step towards securing a computer network. Rapid detection will allow the victim network to trigger appropriate countermeasures to reduce the effects of these attacks. A victim may range from a critical server operating to serve a client-base, to an entire infrastructure network. The attacks themselves vary in type and scopes of their abilities, from trojan horses to report back stolen information to the attacker, to distributed intensity-driven attacks such as Denial of Service (DoS). While the former tends to operate in the background and study the behavior of a machine, with the intent of stealing sensitive information, the latter attack type involves the participation of multiple attacker machines (which may be unaware of their participation in the attack), to send high volumes of traffic to the victim machine in a short interval of time. As a result, traffic will aggregate at the victim's end beyond its ability to process such inflow of high magnitude, consequently causing it to be incapacitated from providing further services.

Intrusion Detection Systems (IDS) have gained significance due to their ability to defend computer networks against the malicious attacks with constantly increasing sophistication. Network-based IDS' detect these attacks by monitoring ingress and egress traffic flow in order to identify the presence of possible outliers in network traffic patterns, where such outliers can be labeled as anomalous traffic. In general terms, intrusion detection systems can be categorized into two types:

- Anomaly detectors, which detect deviations of network traffic behavior from predefined normal traffic profiles. An initial pattern of normal network traffic behavior is learned, and deviations thereof are detected to accurately identify intrusions. Such systems use machine learning techniques and statistical information acquired from network profiles for detecting attacks. With the ability to generalize rules from learned data, anomaly based intrusion detection systems may detect attacks even in the presence of incomplete traffic data, through the use of intelligent techniques. Examples of such approaches include Support Vector Machines (SVMs), Naïve Bayesian, Decision Trees, Neural Networks, and Genetic Programming (Barbara et al., 2001; Stolfo et al., 2001).
- Misuse detectors, on the other hand, attempt to retrieve signatures of malicious patterns known to the IDS beforehand (Portnoy et al., 2001) from observed network traffic. Such detectors are also referred to as signature-based systems. The misuse detection system is trained on a database of intrusion signatures. Subsequently, network traffic is analyzed to identify the presence of these malicious signatures. Misuse detection systems are not able to detect novel intrusions, whose

* Corresponding author. Tel.: +966544017080.
E-mail addresses: baig.zubair@gmail.com, zbaig@kfupm.edu.sa (Z.A. Baig), sadiq@kfupm.edu.sa (S.M. Sait), shaheen@kfupm.edu.sa (A. Shaheen).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.03.008

signatures do not exist in the signatures database. Therefore, to maintain a high degree of accuracy in intrusion detection, an updated version of the attack signature database needs to be introduced to the misuse detector, on a frequent basis, for retraining. Misuse detectors are known for their high degree of accuracy and efficiency in attack detection (Barbara et al., 2001; Stolfo et al., 2001).

In this paper, the problem of network intrusion detection is addressed through classification of network traffic as either normal or anomalous. We introduce the use of Group Method for Data Handling (GMDH)-based networks (Ivakhnenko, 1966) for intelligent classification of network traffic. The GMDH technique has been found to hold promise in the field of intelligent computing. Such a technique for data classification based on established input–output relationships of a dataset, has been applied to diverse application domains, such as educational testing, pattern recognition, spam email classification (Abdel-Aal, 2005), and even for intrusion detecion (Onwubolu and Sharma, 2008). Unlike regression-based techniques, the GMDH technique does not require user intervention for specifying the model relationship or the architecture of the neural network a priori. In addition, it performs well even with fewer training parameters, and yields high accuracies (Agarwal, 1999; Montgomery and Drake, 1991).

The scheme proposed in this paper operates in two phases. During phase 1, selection of the most appropriate network traffic features is performed, and during phase 2, the network traffic is classified as being either normal or anomalous through the use of GMDH-based networks, tested on both ranked as well as the entire feature set. The GMDH network models are built at various levels of complexity and their attack classification performance is studied for the KDD-99 dataset (Kayacık et al., 2005).

The ranking of network traffic features, which are 41 in number for the KDD-99 dataset, is done based on three statistical ranking techniques, namely, Information Gain, Gain Ratio, and GMDH. Our proposed scheme selects the top m common features from the ranked feature lists generated by these three techniques. These features are subsequently introduced as input to the GMDH network for generation of models for network traffic classification. The KDD-99 dataset has a total of 22 attack types, and 1 normal type, as labels (y_i) for each data sample x_i. The GMDH models are generated based on a variation of the model complexity, defined through the Complexity Penalty Multiplier (CPM) parameter. The resulting models are then subject to an unlabeled segment of the dataset, to test the classification accuracy of the generated models.

The contributions of this paper can be outlined as follows:

- Introduction of three prominent statistical ranking techniques, to identify the most relevant network traffic features for the dataset,
- Proposal of a GMDH network-based approach for classification of network traffic into normal or anomalous, and
- Analysis of the simulation results obtained when the proposed scheme is tested on unlabelled network traffic data.

The remainder of this paper is organized as follows. Section 2 discusses related work found in the literature for intelligent network intrusion detection. GMDH-based networks are elaborated upon in Section 3. In Section 4, a detailed description of the proposed intrusion detection technique is provided. Section 5 provides the simulation results together with a detailed analysis and insight. Finally, the concluding remarks are stated in Section 6.

2. Literature Review

Neural networks and Artificial intelligence (AI) techniques have been widely employed for the detection of anomalous traffic in computer communication networks. In this section we summarize some relevant work available in the literature.

An artificial neural network (ANN) consists of a collection of processing elements that are highly interconnected and provide the necessary structure for classifying inputs into expected outputs. They provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources (Cannady, 1998). The neural network performs generalization of malicious attacks for imprecise and uncertain information (Moradi and Zulkernine, 2004), which gives it additional ability to detect novel attacks. Neural network structures have been applied in building anomaly intrusion detection systems and the two most common architectures are the Self-Organizing Maps (SOMs) and its variants, and the Multilayer Perceptron (MLP) (Tavallaee et al., 2009; Mitrokotsa and Douligeris, 2005).

Self-Organizing Map (SOM) is an unsupervised learning algorithm employed to group similar data into clusters. It is a data visualization technique that produces a low dimensional topological map to help understand the original high dimensional data. Once the neural network is trained, the map converges to a stationary distribution and shows a clear separation between normal traffic and attack traffic. The output neurons are considered as the counts for normal and attack traffic points. After building the map using training data, future connections can be quickly classified as normal or anomalous based on their location in the map. Kayacik et al. (2003), Depren et al. (2005), and DeLooze (2006) used SOM in their IDS research.

Kohonen's *Emergent* Self-Organizing Maps is also popularly known as a *winner-take-all* unsupervised neural network. It is unsupervised because there is no target vector which requires the administrator to label the clusters into normal cluster and attack clusters. This approach has advantage of combining machine learning and visualization techniques. However, KSOMs has limited number of neurons in the order of tens, which is not enough for analysis of large datasets with a large number of features. Emergent Self-Organizing Maps produce topological maps that illustrate the intra-data similarity. The map will represent the network traffic in data points in clusters which help to classify it into normal or anomalous depending on the position of its best match cluster. Valleys will have the data points that belong to same class of traffic. Borders will have some points that can be classified to the nearest matched valley.

The MLP is a supervised learning algorithm which uses a feed-forward structure to solve the classification problem. MLP neural networks are trained by manipulating the weights of the neural network connections. The network weights are updated by using different functions during the training period, such as the gradient-based optimization algorithm. When the network converges to the local minima of error, the *output layer* of the network will show the expected result. Faraoun and Boukelif (2006) propose a hybrid method of the k-means algorithm and MLP. The k-means algorithm is used to group the input data into a number of clusters (22 in their case, based on the number of attacks provided in KDD99). The distances between the centers of clusters and input data points are calculated, and only the most discriminating samples that cover the maximum region of each class are selected for the learning process. The selected samples are then presented to the MLP network for division into four classes of attacks, namely, DoS, Probing, U2R, and R2L.

An MLP for misuse detection was proposed by Cannady (1998) that uses two configurations. The first is a stand alone one, and the second uses a rule-based expert system (Cannady, 1998).

The proposed scheme uses nine traffic features as input to the MLP, which means the input layer should contain nine neurons. The MLP network consists of three layers: (i) the input layer with nine neurons, (ii) the hidden layer, and (iii) the output layer with two neurons, with all layers being fully connected. The Sigmoid function is used as a transfer function between the neurons. The author uses 10,000 data points, with 90% of it being used for training, and the remainder for testing.

The Random Neural Network (RNN) model (Gelenbe, 1990, 1989) has also been used successfully for a wide range of applications. It comes in two architectures, namely, feed-forward, or a fully recurrent architecture. RNNs have strong generalization capabilities, even when the training data set is relatively small compared to the actual testing data. The model also achieves fast learning due to its computational simplicity for the weight updating process. RNN was used by Oke and Loukas (2007) for DDoS attack detection. It was used in conjunction with statistical variables like maximum likelihood, Hurst parameter, and Entropy. *Hurst parameter* gives network traffic *self-similarity* while *Entropy* shows how much data is contained in the traffic, that differentiates significantly between normal traffic and anomalous traffic (Oke and Loukas, 2007). In Flegel and Meier (2004), a novel 1-class Support Vector Machine (SVM) has been proposed, specifically designed for handling intrusion detection features, wherein a single sphere is used for representing the class of normal or anomalous connections, with all outliers labeled as being in the opposite class. A quarter-sphere approach was also defined and both the single and the quarter sphere techniques were tested on the KDD-99 dataset. The performance of both approaches under varying anomaly ratios was reported, with the highest average accuracy reported as approximately 90%, at the cost of a 10% false alarm rate, as obtained from the illustrated ROC curves.

In Onwubolu and Sharma (2008), a hybrid differential evolution-GMDH technique for network intrusion detection is proposed. The study evaluated the ability of the differential evolution technique in selecting the most appropriate parameters of the GMDH model to be generated. The resulting model was used for classification of the DARPA dataset entries into normal or anomalous. In Wasniowski et al. (2005), a framework is proposed for agent-based network intrusion detection. The authors have attempted to use the self organization ability of GMDH for pattern classification, when applied to data obtained from local network traffic by the Snort system. However, the results of tests conducted for the proposed scheme were not reported. In contrast, our proposed scheme does feature pre-processing and subsequent model generation on the resulting ranked features of the KDD-99 dataset. The following sections describe the proposed technique and its performance when simulated. Unlike conventional neural networks, GMDH generates models to depict generalization over a dataset without user intervention, and performs well even in the presence of a few independent variables. We provide an in-depth study of the GMDH technique in the following section.

3. GMDH-based networks

The original GMDH is a supervised inductive algorithm for construction of self-organizing models of optimal complexity based solely on the input–output relationships of a given dataset, without the need for user intervention. It introduces a higher-order polynomial to relate each input variable m of the dataset to a single output variable y. The procedure adopted by the GMDH technique for evolving the polynomial so as to find an optimal model to represent the input–output relationship was said to follow the way nature evolves (Farlow, 1981). In order to solve higher order polynomials using traditional techniques such as regression, it would take a substantial amount of time to solve e equations with e unknowns. On the contrary, through GMDH-based model building, the computational overhead is substantially reduced, as the independent variables (i.e. features) that do not have a high correlation with the outputs are discarded during each iteration of the procedure. Through *inductive learning*, the algebraic and finite difference types of polynomial equations, several of these derived, are used for making predictions. On the contrary, the *abductive GMDH* method repairs the original dataset through the replacement of non-essential independent variables with better estimates, obtained during each iteration, to improve the quality of the model that best generalizes the input–output relationship of the dataset.

The *abductory induction mechanism*, is based on the self-organizing polynomial GMDH (Farlow, 1984). It uses mathematical functions for representing numerical knowledge derived from data, and uses artificial neural networks for learning functional models by subdividing complex problems into smaller and simpler ones. This variant of the original GMDH method was developed for inductively creating abductive network models (Abdel-Aal, 2005). It is a powerful supervised inductive learning approach for automatically synthesizing neural network models from input–output data relationships. It is based on the concept of abducting reasoning (Kim and Nelson, 1996), wherein, reasoning is performed from a set of general principles to specifics under uncertainty, through the use of numeric functions, measures, and abductive modeling through machine learning. The model that is formed post-training is a layered network of functional elements connected in a feed-forward manner.

The GMDH approach is a proven concept for iterated polynomial regression that can generate polynomial models for a given dataset using effective predictors. The iterative process involves using initially defined simple intra-data regression relationships, to derive more accurate representations in subsequent iterations of the technique. The number of independent variables, i.e. features that are combined for generating the appropriate models is varied in each step, and the technique is known to perform well even in the presence of a small subset of independent variables in the generated models.

3.1. Steps of execution

The algorithm selects the polynomial relationships and the input combinations that minimize the *prediction error*, during each iteration. This prevents exponential growth in the number of polynomial models generated. Iterations are stopped automatically at a point in time, when a balance between model complexity for accurate fitting of the training data, and model simplicity that allows it to generalize new data accurately, is achieved.

In the classical GMDH-based approach, abductive network models are constructed through the following steps (Farlow, 1984):

1. Data separation: The dataset is to be split into two parts, one for generation of the GMDH models and the other to test the accuracy in classification of the generated models.
2. Modeling: The independent variables (i.e. features) are considered two at a time, for calculation of the least squares polynomial. For a single GMDH node, only one independent variable is considered, and the polynomial equation is limited to the third degree, i.e.

$$y = z0 + z_1 x + z_2 x^2 + z_3 x^3 \quad (1)$$

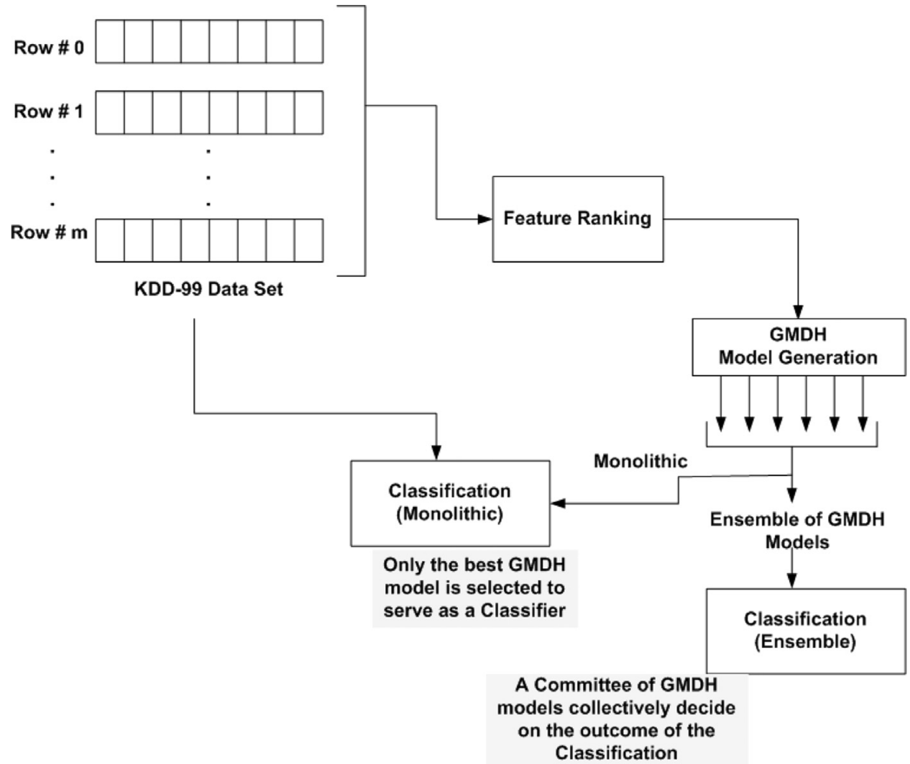

Fig. 1. The attack detection scheme through GMDH model generation using monolithic and ensemble approaches.

where x is the input to the node, y is the output of the node and z_0, z_1, z_2 and z_3 are the node coefficients.
The double node GMDH implementation takes two inputs and the third-degree polynomial equation includes a cross term so as to consider the interaction between the two inputs, i.e.

$$y = z_0 + z_1 x_i + z_2 x_j + z_3 x_i^2 + z_4 x_j^2 + z_5 x_i x_j + z_6 x_i^3 + z_7 x_j^3 \tag{2}$$

where x_i, x_j are the inputs to the node, y is the output of the node and z_0 through z_7 are the node coefficients.

3. Evaluate: The models generated in the previous step are evaluated for each data point n of the training set N, to construct a matrix Z of values generated when the obtained polynomial is used for evaluation of the data points, where, each column of Z represents the outputs generated when the polynomial of the previous step is used for classifying the N data points.
4. Replacement: The columns of the original variables (X) are replaced with those columns of Z which best predicted the output class y. Specifically, the least square error d_j is computed as follows Ivakhnenko (1966):

$$d_j^m = \sum_{i=1}^{t} (y_i - z_{ij})^2 \tag{3}$$

where, t is the number of entries in the test data set.

5. Stopping criteria: The lowest value of $d_j{}^m$ obtained from the previous step is checked to see if this value has decreased in magnitude from the previous iteration. If yes, continue with repetition of steps 2–4 for varying polynomial sizes, else stop execution.

3.2. Abductive network ensemble

Network ensemble is a learning approach where a set of network models, generated by the GMDH implementation based on varying complexities (defined through the CPM parameter), have their respective outcomes of individual data classification, merged, so as to attain higher degrees of classification accuracies. Each element of the network ensemble (or committee) is a GMDH model, trained on a mutually exclusive subset of the original training set. The resulting output of the classifier is generated through appropriate combination of the independent model outcomes of each committee member. The combination of the outputs of each ensemble member is achieved through the use of simple combination rules, such as

1. Simple majority vote: The categorical output of the classifier is a simple majority vote of the categorical output of each individual committee member. An odd number of members will ensure a clear bias towards one of two classes, as opposed to when even number of members constitute the committee.
2. Simple averaging of ensemble network members: In this method, the final output of the committee is computed based on the simple averaging of the outputs of the individual members, through the following relationship:

$$z_i = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{4}$$

Through the testing of our scheme on ensembles of GMDH models, we obtained a set of results for comparison with monolithic GMDH test scenarios. The outcomes of our simulation exercise together with the analysis, is provided in Section 5.

3.3. Feature ranking

GMDH can also be used for ranking features of a given dataset. The feature ranking process is executed through the identification of the predictive quality of the data. The abductive learning algorithm is repeatedly forced to select a small subset of optimum predictors with reduced complexity settings. The process is repeatedly executed, with selected features removed from the dataset during each iteration, with the quality of features in iteration i+1 always being less than that of features for each iteration $t < i+1$. As a result, features are ranked in groups based on predictive quality, with those selected earlier being better predictors.

The ranked features can be selected for model synthesis based on one of two approaches. In the first approach, the top m ranked features can be selected for introduction to the classifier for training. It may be noted that based on system needs, the classifier can be trained on either the final ranked feature subset, or on each feature subset generated during the feature ranking iterations. In the second approach, the top ranked features are determined by repeatedly forming subsets of the m ranked features, with increasing values of m, starting from 1 and reaching the total number of available features in the dataset. For the latter approach, it is postulated that increasing m will lead to nondecreasing accuracy in data classification. The model that yields the lowest classification error rate is thus selected. If two models with different values of m have the same classification error rates, the one generated based on the lesser value of m is selected (Abdel-Aal, 2005).

4. The attack detection scheme

In order to construct a model to accurately represent network traffic as being either normal or anomalous, the abductive network-based intrusion detection scheme proposed in this section operates in two phases. Phase 1 of the scheme is where the network traffic features of the dataset are ranked based on three common techniques, namely, Information Gain, Gain Ratio, and GMDH. The top m ranked features appearing as an intersection set of the outcomes of these three techniques, are selected for abductive modeling. An abductive model is defined as an interconnection of a set of input network traffic features based on specific criteria, as defined below. The model is evolved based on the selected complexity penalty multiplier (CPM) of the algorithm, at time of initialization. The final model of the network traffic (from the dataset) will help classify the data into one of two classes accurately, with the postulation that the model shall best fit any observed network traffic with close similarity to the attack model. The accuracy in detection of the attacks is defined as the ability of a model to correctly distinguish between normal and anomalous traffic data. An illustration of the proposed intrusion detection scheme is provided in Fig. 1. It may be noted that we address the issue of intrusion detection as a two-class problem, wherein, the network traffic is either tagged as normal or anomalous. Intra-class differentiation between the various attack types of the dataset is beyond the scope of our proposed scheme.

Prior to the application of any training algorithm on a given data set, it is essential to convert all features (attributes) to a format that is intelligible by the classification algorithm. Subsequent to preprocessing of data, the features of the data set are identified as either being significant to the intrusion detection process, or redundant. This process is known as feature selection. Redundant features are generally found to be closely correlated to one or more other features. As a result, omitting them from the intrusion detection process does not degrade classification accuracy. In fact, the accuracy may improve due to the resulting data reduction, and removal of noise and measurement errors associated with the omitted features. Therefore, choosing a good subset of features proves to be significant in improving the performance of the system.

The features are filtered to create the most prominent feature subset before actual GMDH-based model generation is performed. The three feature ranking techniques constituting the proposed technique are summarized as follows:

1. *Information Gain*: is used to individually rank attributes based on class separation in the dataset rows. Attribute ranks can be calculated using Information Gain with respect to class based on the following formula:

$$\textit{Information Gain} = (D_x) - (D_{-x}) \tag{5}$$

where D_x is the information which includes attribute x, and D_{-x} is information which excludes attribute x. The value of D_{-x} is calculated as the average of each value that this particular attribute can take. The information itself is calculated using the entropy equation:

$$\textit{entropy} = D_x = -\sum_{k=1}^{n} p_k \log p_k \tag{6}$$

where p_k is the probability of occurrence of value k for feature x, with the total number of distinct values of feature x being equal to n.

2. *Gain Ratio* is an improvement of the Information Gain technique that resolves the bias towards features which have a larger diversity of values. For example, if a dataset contains a diverse range of serial numbers of customers of a grocery store, then the Information Gain of the customer serial number will be high, and it will be used at the high level in decision trees. This bias degrades the ability of learning algorithms, such as decision trees, of generalization of new customers because the serial number will be considered on the top of the decision tree, as a result causing a skew in the accuracy in the recognition process. Information Gain Ratio corrects this shortcoming by taking the intrinsic information in terms of entropy of distribution of instance values, for a given attribute i.e. feature. The Gain Ratio is large when the data is evenly spread and is small when the data has a single value. It is calculated as following:

$$\textit{Gain Ratio}(\textit{Feature}) = \frac{\textit{Information Gain}}{\textit{Intrinsic Value}} \tag{7}$$

where,

$$\textit{Intrinsic Value} = -\sum_{v \in values(a)} \frac{|x \in S, value(x,a) = v|}{|S|} \cdot \log_2 \frac{|x \in S, value(x,a) = v|}{|S|} \tag{8}$$

where S is the set of all samples of the dataset, x is a dataset sample, a is a feature of the dataset, $values(a)$ is the set of all possible values of feature a of the dataset, and $value(x,a)$ is defined as the value of feature a in the dataset sample x. *Information Gain* is defined through Eq. (7). The numerator is the information we learn about the class. The denominator however, represents the information we learn about the attribute (feature), or in other words, the information necessary to specify the feature value of a particular attribute.

3. *GMDH* synthesizes optimized polynomial network structures through continuous iterations. Instead of using the two techniques for feature ranking mentioned above, a straightforward approach towards classifying the data is to use GMDH for feature ranking as well, prior to actual classification of the data. Feature ranking using abductive networks through the wrapper approach (Witten et al., 2011; Bello et al., 2008; Guyon, 2009) is done based on the predictive quality of the data, and consists of the following steps:
   (a) Model synthesis to select three inputs (features) to the abductive network at any given time.

(b) Removal of selected features to force the model to select from the less-predictive remaining features.
(c) Repetition of the process until all features are selected or no further features can be selected.
(d) Change model complexity in steps from small to large, if needed, to force the modeler to select the remaining features.

After feature ranking is completed, the top ranked features are considered for attack detection, one at a time as long as the accuracy of the selected model is non-decreasing. We stop when the accuracy drops, as an indication of model overfitting. In Section 5, we follow this procedure at different levels of model complexity, numbers of layers, and numbers of inputs, and study the corresponding effect on the attack detection process.
A comparison of the impact of all three techniques on the attack detection accuracy is provided in Section 5.

Subsequent to filtering and selection of the highest ranked features for the intrusion detection process, the reduced data set is used for training and evaluating the detection scheme.

5. Simulation results and analysis

This section describes the simulation performed for feature ranking based on the three statistical techniques defined in Section 3, and to build models of GMDH networks for network traffic classification. The simulator as such provides for simultaneous feature selection and model building on the dataset. The dataset itself was partitioned with 75% of it being used for training and the remainder 25% of unlabeled data used for testing the accuracy of the proposed technique. The GMDH networks were modeled at various levels of complexity, defined through the CPM (Complexity Penalty Multiplier) parameter. The value of CPM has an inverse effect on the complexity of the model generated (i.e. the number of levels of the model and the interconnections between the levels). Therefore, smaller CPM values will lead to more complex models as opposed to larger ones.

5.1. The dataset

For testing the accuracy of the GMDH models in distinguishing normal from anomalous traffic, the KDD-99 dataset (Tavallaee et al., 2009) was used. This dataset was originally derived from the raw DARPA network traffic. In the dataset, the network connection details that were obtained from the raw data were parsed into a vector with 41 distinct features. The processing of raw network connection data was carried out through the use of data mining and expert systems, so as to emulate a misuse-based network intrusion detection system. In addition, each attack type of the dataset was categorized into one of four categories, namely, Denial of Service, U2R, R2L, and Probing. Several intrusion detection schemes have been proposed in the past, with their corresponding performances being tested on the KDD-99 dataset (Yu, 2008; Sabhnani, 2003; Ahmad et al., 2008; Middlemiss and Dick, 2003; Zhang et al., 2011). The 41 features of the dataset constituting the feature vector are constituted of

- Thirteen content-based features derived from network traffic payload. These features were constructed to identify U2R and R2L attacks.
- Ten host-based header features constructed over a 100 s time window to detect slow probes (Denial of Service) attacks.
- Ten temporal header features constructed over a 2 s time window, and
- Nine basic and header features to depict the state of each connection.

Table 1
Top 30 features identified by the three techniques.

Rank	GMDH	Information Gain	Gain Ratio
1	4	38	36
2	36	39	32
3	62	63	45
4	45	62	59
5	67	66	58
6	69	36	71
7	66	67	72
8	72	68	39
9	73	71	62
10	59	45	14
11	71	72	63
12	74	58	38
13	58	59	19
14	60	56	11
15	70	32	66
16	14	70	70
17	32	14	67
18	61	65	68
19	2	69	41
20	19	19	20
21	56	64	56
22	57	57	23
23	63	74	24
24	64	11	16
25	3	73	6
26	65	60	17
27	68	61	10
28	11	20	25
29	20	3	5
30	28	4	21

We utilized the NominalToBinary *Weka* WEKA Data Mining Software filter for obtaining a binary feature set for the dataset, to facilitate GMDH model creation. As a result, each k-valued feature of the dataset was transformed to k binary features. For instance, if the feature $\{protocol_{ID}\}$ can possess one of three different nominal values, namely, $\{TCP, UDP, RTP\}$, this particular feature will be transformed into three distinct features, labeled as *TCP*, *UDP*, and *RTP*, respectively. Each of these three newly generated features will be able to hold a binary value to represent either the presence or absence of the particular feature in a sample of the original dataset. As a result, for a dataset with k nominal features, with each feature capable of possessing one of l_k distinct values (3 in our case for the example above), the total number of transformed features that will be obtained is equal to: $\sum_{i=1}^{k} l_k$. The number of features that were obtained post-transformation for the KDD-99 dataset is equal to 123. In order to reduce the total number of features to be used for intrusion detection, the three feature selection techniques elaborated upon earlier were applied to this transformed dataset.

5.2. Performance measures

The performance of intelligent classifiers may be measured using several metrics. The confusion matrix is one such visualization tool used for tabulating the overall performance of the classifier. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. The following measures are derived from the confusion matrix, and will be used for evaluating the proposed scheme:

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

$$Recall = true\ positive\ rate = \frac{TP}{TP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Specificity = \frac{TN}{TN + FP} \tag{12}$$

$$Detectionrate = \frac{Attacks\ detected}{Total\ Number\ of\ attacks} * 100\% \tag{13}$$

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

The Receiver Operating Characteristic (ROC) area is used for weighing the performance of the GMDH classifier on the input feature set, through the following defined levels:

- 1.0: perfect prediction
- 0.9: excellent prediction
- 0.8: good prediction
- 0.7: mediocre prediction
- 0.6: poor prediction
- 0.5: random prediction
- <0.5: poor prediction.

Precision–Recall (PR) curves, often used in Information Retrieval (Manning and Schlutze, 2000; Raghavan et al., 1989), have been cited as an alternative to ROC curves for tasks with *a large skew in the class distribution* (Bunescu et al., 2005; Goadrich et al., 2004). An important difference between ROC space and PR space is the visual representation of the curves. Looking at PR curves can expose differences between algorithms that are not apparent in ROC space.

5.3. Feature ranking results and analysis

Simulations performed to test the proposed scheme can be divided into two phases. During Phase 1, the three feature ranking techniques defined in Section 3, were implemented to rank the features of the dataset. For running simulations based on selected features, the commonly occurring features in the three lists of ranked features, are selected (see Table 1). These selected features are then introduced to the abductive network during Phase 2 of the scheme, for building generic models to represent the dataset (i.e. training), and for subsequent classification of unlabeled data, i.e. testing of the dataset to quantify the accuracy in attack detection.

5.4. Monolithic abductive models

For monolithic abductive models, we ran the simulation using all features from the training set, and different CPM values, i.e. $CPM = 0.1, 0.5, 1, 2, 5$. It may be noted that all features, regardless of their ranking, were introduced to the simulator, incrementally. A total of 65 abductive network models were built, with each model consisting of four layers and varying CPM values. In Table 2, the attack detection and the false alarm rates are illustrated for five synthesized GMDH models with varying CPM values. It is evident from the findings that the accuracy of the synthesized models remains consistent around 97.6%, unaffected by the

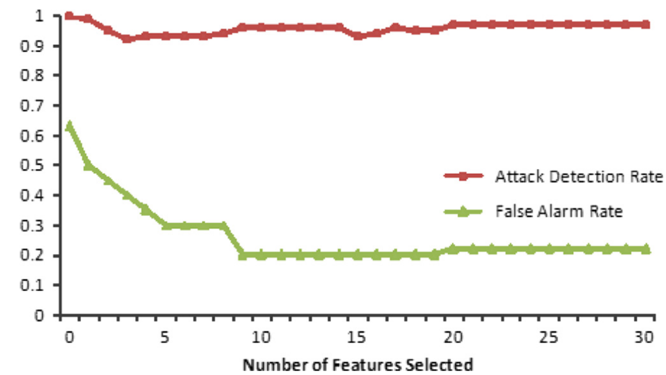

Fig. 2. Attack detection rate vs. number of features selected, with five layer abductive networks.

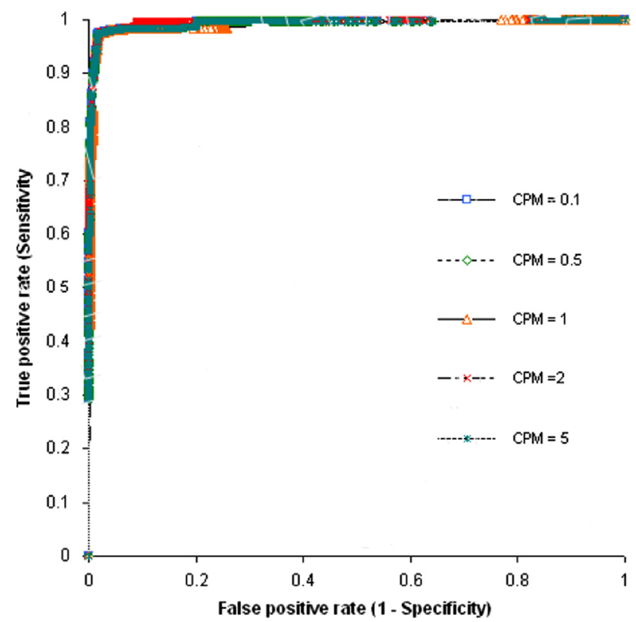

Fig. 3. Comparison of the ROC curve for five abductive network classifiers: the optimum monolithic model when $CPM = \{0.1, 0.5, 1, 2, 5\}$.

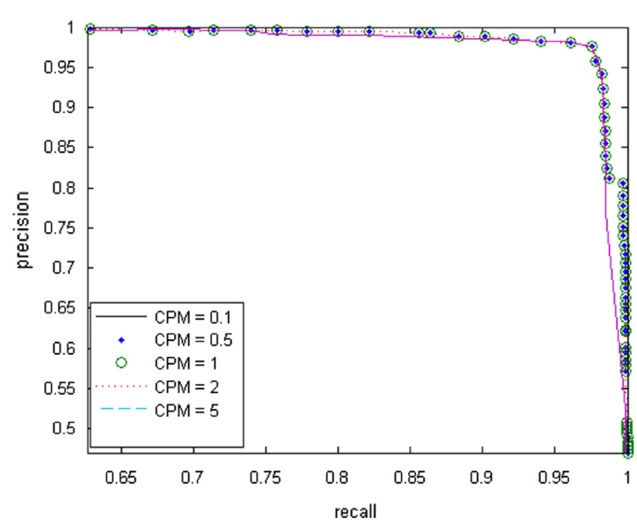

Fig. 4. Comparison of the Precision–Recall curve for five abductive network classifiers: the optimum monolithic model when $CPM = \{0.1, 0.5, 1, 2, 5\}$.

variation of the CPM value. The false alarms associated with the scheme can be seen to remain constant at 2.2%, for varying values of CPM. The simulation was run a second time with the number of

Table 2
Performance results of five abductive network models synthesized using five different CPM values.

CPM	FN	TN	FP	TP	FAR	DR
0.1	57	2769	65	2358	0.022	0.976
0.5	57	2769	65	2358	0.022	0.976
1	60	2777	57	2355	0.020	0.975
2	56	2773	61	2359	0.021	0.976
5	57	2769	65	2358	0.022	0.976

Table 3
Outcomes of simulation done to study the effect of synthesizing models based on the top-ranked 14 and 20 features on the attack detection process.

No. of features	FN	TN	FP	TP	FAR	DR
14	112	2753	81	2303	0.028	0.953
20	74	2775	59	2341	0.020	0.969

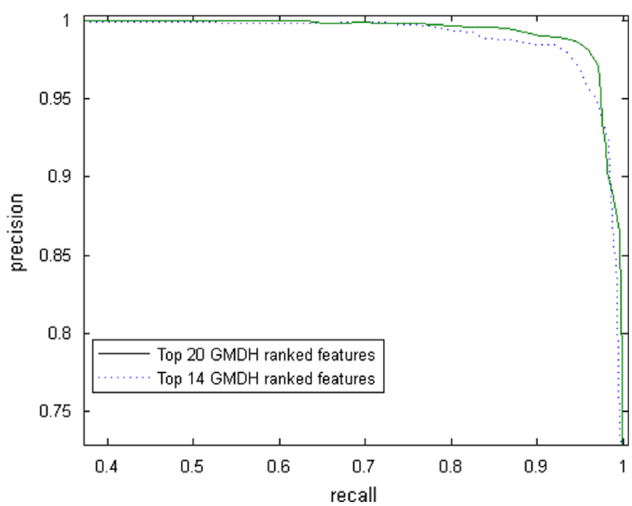

Fig. 5. Comparison of the Precision–Recall curve for two abductive network classifiers synthesized using 14 and 20 top ranked GMDH features.

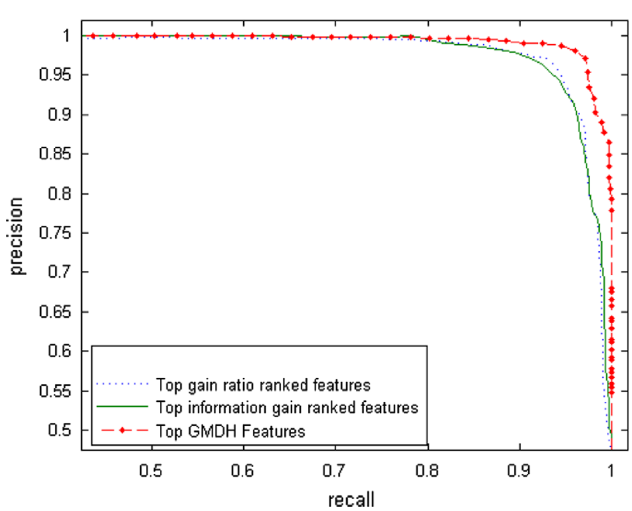

Fig. 6. Comparison of the Precision–Recall curve for three abductive network classifiers: network models synthesized using top 14 GMDH, Gain Ratio, and Information Gain-ranked features.

network layers set to five and with $CPM = 1$. Fig. 2 shows that the model stabilizes both in terms of the detection rate as well as the false alarm rates, beyond $k = 20$, when a total of five layers of an

Table 4
Performance results of different network models synthesized using top 14 features selected using different feature selection algorithms.

Model	FN	TN	FP	TP	FAR	DR
GMDH	112	2753	81	2303	0.028	0.953
Information gain	152	2712	122	2263	0.043	0.937
Gain Ratio	59	2602	232	2356	0.082	0.975

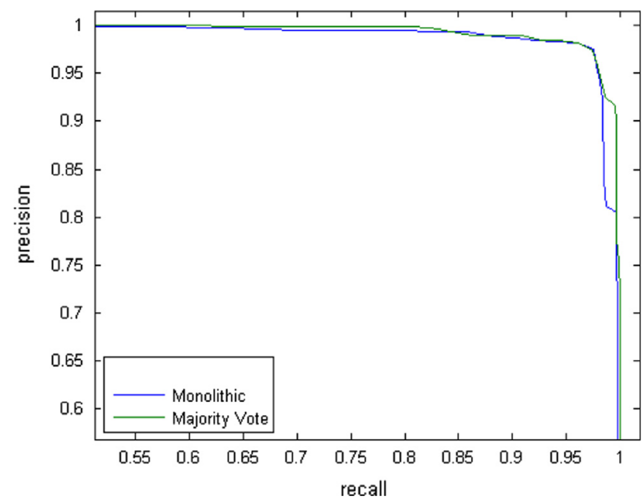

Fig. 7. Comparison of the Precision–Recall curve for two abductive network classifiers: the optimum monolithic model when CPM = 1, and a three member network ensemble based on majority voting.

Table 5
Performance results of ensemble network individual models synthesized using CPM = 1.

Model	FN	TN	FP	TP	FAR	DR
Monolithic	60	2777	57	2355	0.020	0.975
Majority vote ensemble	63	2773	61	2352	0.021	0.973

Table 6
Performance comparison of various intelligent techniques for network intrusion detection.

Intrusion detection scheme	False alarm rate (%)	Attack detection rate (%)
PCC (Shyu et al., 2003)	2	96.07
GMDH	2.25	97.72
AODE (Baig et al., 2011)	00.1	99.54
NB	1.08	88.55
MLP (Sabhnani and Serpen, 2003)	3.5	93

abductive network (for CPM = 1) are used for classification. It may be noted that with $k = 1$, the detection rate was found to be 100%, albeit with a very high false alarm rate. It is therefore impractical to have a scheme implementation wherein only a single feature is used for classification.

In Fig. 3, the ROC curve is illustrated as a performance measure for the abductive models, for varying CPM values. From these figures, it is evident that the CPM value does not have an effect on the attack detection rate. The area under the curve is nearly 99% for all cases. Fig. 4 shows the Precision–Recall curves for different abductive networks synthesized using varying model complexity values. The observed results are again unaffected by varying CPM values. However, higher CPM values proved to slightly improve the performance as compared to lower CPM values, in terms of recall.

The simulation was also run with varying numbers of top-ranked features. When the top 14 and 20 ranked features were selected, with CPM = 1, and the number of GMDH layers set as 4, the results were not as good as those obtained when using the full feature set, but were comparable to a certain extent. However, it was noticed that the training time for the reduced feature set simulation run was much less than the time required for running on the full feature set. The time required for training, i.e. abductive network model building, was found to improve with decreasing numbers of features. When all 123 features were used for training, the simulator took an estimated 1805 s for model building, whereas, with 14 features selected, the training time reduced to 589 s (Table 2).

Table 3 shows the results of using the top 14 and 20 commonly ranked features by all three techniques from Section 3. For the same simulation, Fig. 5 illustrates the precision–recall curves. As may be observed from both Table 3 and Fig. 5, the detection rate showed a slight degradation in performance, reaching a maximum of only 96.9% when 20 features are selected, as opposed to 97.9% when all features are selected (from the previous subsection results). For the 14-feature case, the false alarm rate was found to be 2.8%, whereas if 20-features are selected, the false alarm rate drops down to 2.0%.

5.5. Abductive networks for top-ranked features

After performing simulation runs with all features selected, we synthesized abductive networks using the top 14 selected features, ranked by the three feature selection algorithms outlined in Section 3. The resulting networks are compared based on the precision–recall curve, as shown in Fig. 6. The area under the curve for the GMDH selected features, is 0.993, whereas the area under the curve for the abductive network model synthesized using Gain Ratio-selected features is 0.990, and the area under the curve based on features selected through Information Gain is 0.983. It can be observed here that abductive networks synthesized using the GMDH top-ranked features outperform abductive networks synthesized based on features ranked through Gain Ratio and Information Gain. The abductive networks synthesized using the Gain Ratio approach outperform abductive networks synthesized using Information Gain. Abductive networks synthesized using Information Gain were found to yield better Recall values as opposed to Precision values.

In Table 4, we provide an illustration of the attack detection and the false alarm rates obtained when the top 14 features of each of the three feature selection techniques, are used for abductive network model synthesis. As may be observed, in terms of attack detection rates, features ranked highest by Gain Ratio proved to yield a detection rate of 97.5% when used for GMDH model generation. On the contrary, GMDH-ranked features yielded a detection rate of 95.3%. Gain Ratio was found to generate the most number of false alarms, at 4.3%, followed by a 4.3% false alarm rate of the Information Gain technique. Features ranked through GMDH were found to best model the dataset, and yielded the lowest false alarm rate of 2.8% (at par with results from the previous sub-section).

5.6. Ensemble abductive models

The best abductive network model was synthesized for $CPM = 1$ (see Section 5.4). Therefore, an ensemble network comprising of three separate abductive network models was built, with the value of *CPM* fixed at 1, and all 123 features selected. The GMDH models generated by the committee were combined based on a majority vote, wherein the output classification was performed based on the majority voting of the binary classification outputs of the three classifiers. The results of the ensemble network were compared through precision–recall curves, thus obtained. From Fig. 7, it may be observed that the ensemble classifier improves on the performance results of the monolithic abductive network.

The area under the curve was found to be 0.9963, whereas for the monolithic model, wherein the closest model to a given input was selected without having the need for a committee of classifiers for deciding the outcome of the attack detection process, was found to be 0.993.

Table 5 provides a comparison of the results obtained through network ensembles against monolithic networks. The attack detection rate for the monolithic network was found to be 97.5% as compared to a 97.3% rate for the ensemble network. In addition, the false alarm rates for both approaches were comparable, at 2.0% and 2.1%, respectively. Therefore, it may be conclusively stated that the effect of an ensemble networks on improving the performance of the proposed approach for intrusion detection, is insignificant.

5.7. Performance comparison

Table 6 shows that the performance of our proposed scheme falls second only to Averaged One-Dependence Estimator (AODE) in terms of attack detection rates and fourth in terms of false alarm rates. Although the false alarms generated by AODE, Naive Bayes and Principal Component Analysis (PCC) are less than those generated by our proposed scheme, the attack detection rate is only second to AODE. We can therefore infer from the findings that the scheme proposed in this paper is closely comparable to the best known schemes for network intrusion detection, through intelligent classification.

6. Conclusions

Abductive learning methods have been found to hold promise in the field of intelligent computing. Through this paper, a two-phased approach towards classifying network traffic into normal and anomalous, was proposed. The technique operates through the identification of the most significant features of the KDD-99 dataset during phase 1. The feature ranking process is performed based on three techniques, namely, Information Gain, Gain Ratio, and GMDH. Subsequently, these top-ranked features are introduced to the simulator for modeling of abductive networks. These models help classify the traffic data of the KDD-99 dataset into either normal or anomalous. Simulation results of the monolithic abductive network models with and without feature selection, were analyzed. In addition, the effect of varying GMDH model complexities (defined through the CPM paramter) on the performance of the scheme was analyzed. It was found that ranking and subsequent selection of ranked features improved the performance of the scheme in terms of improved attack detection rates and reduced false alarm rates, as opposed to when all features of the dataset were used. In addition, the training time significantly reduced with decreasing numbers of features. A similar set of simulations were performed for ensemble abductive networks. It was observed that ensemble networks based on majority voting yielded insignificant improvements in performance over monolithic abductive networks.

Acknowledgments

The authors wish to acknowledge the continued support for research provided by King Fahd University of Petroleum & Minerals.

This research work was conducted as part of research project no. NSTIP-11-INF1658-04.

References

Abdel-Aal, R., 2005. GMDH-based feature ranking and selection for improved classification of medical data. J. Biomed. Inf. 38 (6), 456–468.

Agarwal, A., 1999. Abductive networks for two-group classification: a comparison with neural networks. J. Appl. Bus. Res. 15, 1–12.

Ahmad, I., Ansari, M., Mohsin, S., 2008. Performance comparison between backpropagation algorithms applied to intrusion detection in computer network systems. In: Proceedings of the 7th WSEAS International Conference on Applied Computer and Applied Computational Science, pp. 47–52.

Baig, Z.A., Shaheen, A., Abdel-aal, R., 2011. An AODE-based network intrusion detection system. In: Proceedings of the IEEE World Congress on Information Security.

Barbara, D., Wu, N., Jajodia, S., 2001. Detecting novel network intrusions using Bayes estimators. In: Proceedings of the First SIAM Conference on Data Mining.

Bello, J.P., Chew, E., Turnbull, D. (Eds.), 2008. In: Proceedings of the Ninth International Conference on Music Information Retrieval.

Bunescu, R., Ge, R., Kate, R., Marcotte, E., Mooney, R., 2005. Comparative experiments on learning information extractors for proteins and their interactions. Artif. Intell. Med. 33 (2), 139–155.

Cannady, J., 1998. Artificial neural networks for misuse detection. In: Proceedings of the National Information Systems Security Conference, pp. 443–456.

DeLooze, L., 2006. Attack characterization and intrusion detection using an ensemble of self-organizing maps. In: Proceedings of the IEEE Information Assurance Workshop.

Depren, O., Topallar, M., Anarim, E., Ciliz, M., 2005. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst. Appl. 29 (4), 713–722.

Faraoun, K., Boukelif, A., 2006. Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions. Int. J. Comput. Intell. 3 (2), 161–168.

Farlow, S.J., 1981. The GMDH algorithm of Ivakhnenko. Am. Stat. 35, 210–215.

Farlow, S., 1984. Self-Organizing Methods in Modeling: GMDH-Type Algorithm. CRC Press.

Flegel, U., Meier, M., 2004. Detection of intrusions and malware and vulnerability assessment. In: Proceedings of the GI Special Interest Group SIDAR Workshop (DIMVA).

Gelenbe, E., 1989. Random neural networks with negative and positive signals and product form solution. Neural Comput. 1 (4), 502–510.

Gelenbe, E., 1990. Stability of the random neural network model. Neural Comput. 2 (2), 239–247.

Goadrich, M., Oliphant, L., Shavlik, J., 2004. Learning ensembles of first-order clauses for recall-precision curves: a case study in biomedical information extraction. In: Proceedings of the 14th International Conference on Inductive Logic Programming ILP, pp. 98–115.

Guyon, I., 2009. A practical guide to model selection. In: Proceedings of the Machine Learning Summer School Text in Statistics, Springer.

Ivakhnenko, A., 1966. Group method of data handling—a rival of the method of stochastic approximation. Sov. Autom. Control 13, 43–71.

Kayacik, G.H., Zincir-Heywood, A.N., Heywood, M.I., 2003. On the capability of SOM based intrusion detection systems. In: Proceedings of the 2003 IEEE International Joint Conference on Neural Networks (IJCNN-2003).

Kayacık, H., Zincir-Heywood, A., Heywood, M., 2005. Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets. In: Proceedings of the Third Annual Conference on Privacy, Security and Trust.

Kim, K.S., Nelson, W.A., 1996. Assessing the rental value of residential properties: an abductive learning networks approach. J. R. Estate Res. 12 (1), 63–77.

Manning, C., Schlutze, H., 2000. Foundations of Statistical Natural Language Processing. MIT Press.

Middlemiss, M., Dick, G., 2003. Design and application of hybrid intelligent systems. Feature selection of intrusion detection data using a hybrid genetic algorithm/KNN approach, pp. 519–527.

Mitrokotsa, A., Douligeris, C., 2005. Detecting denial of service attacks using emergent self-organizing maps. In: Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).

Montgomery, G., Drake, K., 1991. Abductive reasoning networks. Neurocomputing 2, 97–104.

Moradi, M., Zulkernine, M., 2004. A neural network based system for intrusion detection and classification of attacks. In: Proceedings of the IEEE International Conference on Advances in Intelligent Systems—Theory and Applications.

Oke, G., Loukas, G., 2007. A denial of service detector based on maximum likelihood detection and the random neural network. Comput. J. 50 (6), 717.

Onwubolu, G., Sharma, A., 2008. Intrusion detection system using hybrid differential evolution and group method of data handling approach. In: Proceedings of the International Conference on Information Management.

Portnoy, L., Eskin, E., Stolfo, S.J., 2001. Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security.

Raghavan, V., Bollmann, P., Jung, G., 1989. A critical investigation of recall and precision as measures of retrieval system performance. ACM Trans. Inf. Syst. (TOIS) 7 (3), 205–229.

Sabhnani, M., 2003. Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In: Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications, pp. 209–215.

Sabhnani, M., Serpen, G., 2003. Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In: Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications.

Shyu, M., Chen, S., Sarinnapakorn, K., Chang, L., 2003. A novel anomaly detection scheme based on principal component classifier. In: Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop.

Stolfo, S.J., Lee, W., Chan, P.K., Fan, W., Eskin, E., 2001. Data mining-based intrusion detectors: an overview of the columbia IDS project. ACM SIGMOD Record 30 (4), 5–14.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A., 2009. A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, IEEE Press, pp. 53–58.

Wasniowski, R.A., 2005. Using self-organizing networks for intrusion detection. In: Proceedings of the Sixth WSEAS International Conference on Neural Networks, pp. 90–94.

WEKA Data Mining Software, Machine Learning Group, University of Waikato Available at: ⟨http://www.cs.waikato.ac.nz/ml/weka/index.html⟩.

Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers.

Yu, Z., Tsai, J., Weigert, T., 2008. An adaptive automatically tuning intrusion detection system. ACM Trans. Auton. Adapt. Syst. 3, 10:1–10:25.

Zhang, Y., Wang, L., Sun, W., II, R.C.G., Alam, M., 2011. Distributed intrusion detection system in a multi-layer network architecture of smart grids. IEEE Trans. Smart Grid 2, 796–808.