Scientific Research and Essays Vol. 6(34), pp. 6804-6810, 30 December, 2011
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.142
ISSN 1992-2248 ©2011 Academic Journals

Full Length Research Paper

Intrusion detection using feature subset selection based on MLP

Iftikhar Ahmad[1,2]*, Azween Abdullah[1], Abdullah Alghamdi[2], Khalid Alnfajan[2] and Muhammad Hussain[3]

[1]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia.
[2]Department of Software Engineering, College of Computer and information Sciences, P.O. Box 51178, Riyadh 11543, King Saud University, Saudi Arabia.
[3]Department of Computer Science, College of Computer and information Sciences, P.O. Box 51178, Riyadh 11543, King Saud University, Saudi Arabia.

Accepted 19 December, 2011

Intrusions are serious questions in network systems. Numerous intrusion detection techniques are present to tackle these problems but the dilemma is performance. To raise performance, it is momentous to raise the detection rates and decrease false alarm rates. The contemporary methods use Principal Component Analysis (PCA) to project features space to principal feature space and choose features corresponding to the highest eigenvalues, but the features corresponding to the highest eigenvalues may not have the best possible sensitivity for the classifier due to ignoring several sensitive features. Therefore, we applied a Genetic Algorithm (GA) to search the principal feature space for a subset of features with optimal sensitivity. So, in this research, a method for optimal features subset selection is proposed to overcome performance issues using PCA, GA and Multilayer Perceptron (MLP). The KDD-cup dataset is used. This method is capable to minimize amount of features and maximize the detection rates.

Key words: KDD-cup, PCA, MLP, GA, detection rate, and false alarm.

INTRODUCTION

The previous techniques of intrusion detection have concentrated on the problems of feature extraction and classification. But, somewhat less attention has been given to the significant subject of feature selection. The prime trend in feature extraction has been representing the data into another feature space (the PCA space) using principal component analysis (PCA). In this process of selecting features on the basis of highest eigenvectors is not fitting because the features corresponding to the highest eigenvalues may not have the best sensitivity for the classifier due to ignoring many sensitive features (Ahmad et al., 2010, 2011a, 2011b). Consequently, there must be an efficient method to select a suitable set of features in the PCA space. This will lead the classifier to work in a competent way and enhance the overall performance of the intrusion analysis engine. Since, the redundant and irrelevant features increase overheads as well as confuse the classifier. Therefore, in this paper, we argue that feature selection is an imperative dilemma in intrusion detection and exhibit that genetic algorithms (GAs) provide a simple, general, and potent framework for selecting first-class subsets of features that advance detection rates (Sun et al., 2002).

Furthermore, we considered PCA for features transformation and MLP for classification. The goal is searching the PCA space using GA to select a subset of principal components. This is in contrast to conventional methods selecting some percentage of the top principal components to represent the target concept, independently of the classification task. We have tested the proposed framework on intrusion detection. Our experimental results demonstrate important performance improvements. A number of approaches have been described in the area of intrusion detection but the key

*Corresponding author. E-mail: wattoohu@gmail.com.

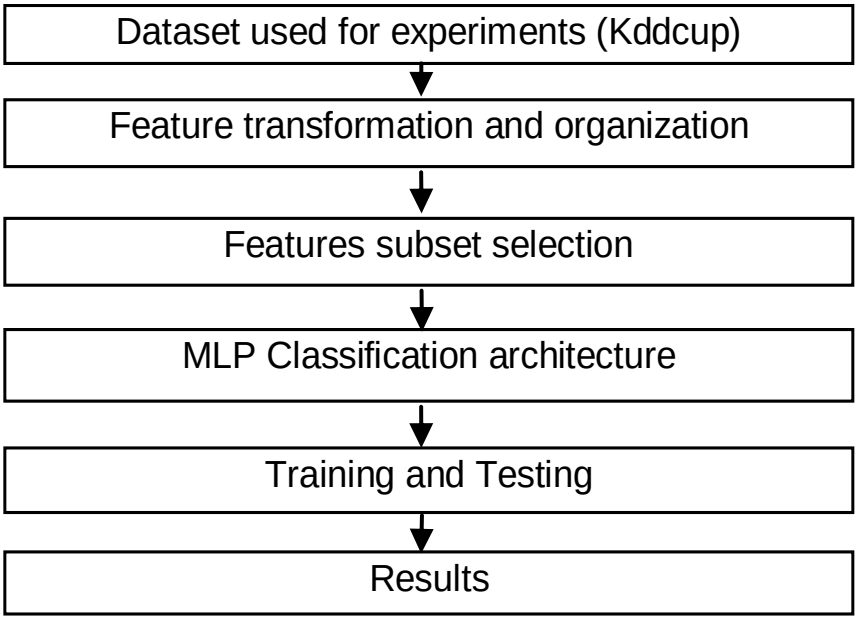

Figure 1. Proposed model.

centre is on classification.

In one existing research work by Liu and his colleagues, PCA is applied for classification and neural networks are used for online computing. They selected 22 principal components as features subset selection to obtain the best performance. But there is a possibility to miss many vital principal components having sensitive information for intrusion detection during selection phase (Liu et al., 2007).

Horng and his co-workers observed the important features based on the accuracy and the number of false positives of the system with and without the feature. In other words, the feature selection of is “leave-one-out”; remove one feature from the original dataset, redo the experiment, then compare the new results with the original result, if any case of the described cases occurs. The feature is regarded as significant; otherwise it is regarded as insignificant. Since there are 41 features recommended in the KDD-cup99, the experiment is repeated 41 times to certify that each feature is either essential or insignificant. This process involved complication as well as overheads on massive dataset (Horng et al., 2010).

One of the most important works is done by Tong and his associates in which they employed the radial basis function (RBF) network as a real-time pattern classification and the Elman network is applied to reinstate the memory of past events. They used full featured KDD-cup dataset. This increases training and testing overheads on the system (Tong et al., 2009).

PCA method is used by Zargar and his colleagues to determine an optimal feature set. An appropriate feature set helps to build efficient decision model as well as to reduce the population of the feature set. Feature reduction will speed up the training and the testing process for the attack identification system considerably but this will be a compromise between training efficiency (few PCA components) and the accurate results (a large number of PCA components) (Zargar et al., 2010).

In one of the research works by Kim and his team, the fusions of Genetic Algorithm (GA) and Support Vector Machines (SVM) are described for optimization of both features and parameters for detection models. This method was able to minimize amounts of features and maximize the detection rates but the problem is features uniformity. The features in original forms are not consistent so these must be transformed in new feature space in order to well organized form (Kim et al., 2005).

PROPOSED MODEL

The model consists of different parts; dataset used for experiments, feature transformation and organization, optimal feature subset selection, MLP classification architecture, training and testing, and results. The block diagram of model is shown in the Figure 1.

Dataset used for experiments

We used kddcup99 dataset for our experiments. The selection of this dataset is due to its standardization, content richness and it helps to evaluate our results with existing researches in the area of intrusion detection. The raw dataset consists of 41 features:

$$x_1, x_2, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots x_n \quad (1)$$

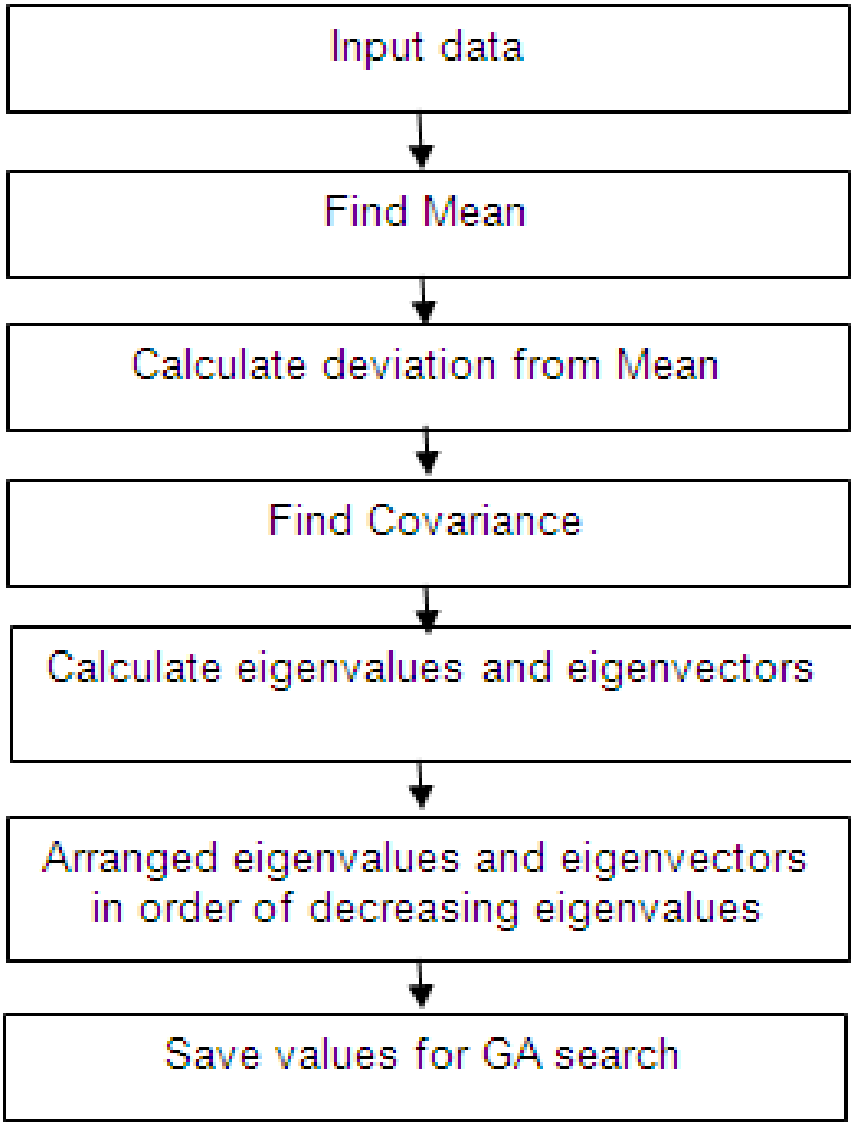

Figure 2. PCA flow chart.

Where n = 41

After selection of the dataset, first, we pre-processed on the raw dataset so that it can be given to the selected classifiers; MLP. The raw dataset is pre-processed. First of all, we discarded three symbolic values (for example, udp, private and SF) out of 41 features of the dataset. The resultant features are:

$$x_1, x_2, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots x_m \quad (2)$$

Where m = 38

Feature transformation and organization

We applied PCA on 38 features of the dataset. The PCA flow chart is shown in Figure 2.

PCA Algorithm

Suppose $x = (x_1, x_2, x_3, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, x_M)$ are NX1 vectors. Where M = 38.

Step 1: Find mean:

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$$

Step 2: Calculate deviation from mean: Subtract the mean: $\Phi_i = (x_1 - \bar{x})_i$

Where i=1, 2 … M.

Step 3: Find covariance matrix C:

From the matrix A= [$\Phi 1, \Phi 2, \Phi 3 \ldots \ldots \ldots \ldots \Phi M$] (N*M Matrix), compute C:

$$C = \frac{1}{M}\sum_{N=1}^{M} \Phi_N \, \Phi_N = AA^T$$

Step 4: Compute the eigenvalues of $C: \lambda_1 > \lambda_2 > \lambda_3 > \ldots\ldots\ldots \lambda_N$

Step 5: Compute the eigenvectors of $C: \mu_1, \mu_2, \mu_3, \ldots\ldots\ldots \mu_N$
Since C is symmetric, $C: \mu_1, \mu_2, \mu_3, \ldots\ldots\ldots \mu_N$ form a basis, (i.e. any vector x or actually $(x_1 - \bar{x})$, can be written as a linear combination of the eigenvectors):

$$(x_1 - \bar{x}) = b_1\mu_1 + b_2\mu_2 + \ldots\ldots\ldots\ldots\ldots\ldots + b_N\mu_N = \sum_{i=1}^{N} b_i\mu_i$$

Step 6: Arranged eigenvalues and eigenvectors in descending order.

Step 7: The dimensionality reduction step (based on largest eigenvalues) is skipped as the selection of principal components is dine using GA. Mostly, PCA is used for data reduction, but here, we used it for feature transformation into principal components feature space and then organized principal components in descending order:

$$pc_1 > pc_2 > pc_3 \ldots\ldots\ldots\ldots > pc_l \quad (3)$$

Where l=38

Feature subset selection

We applied genetic algorithm (GA) for optimal features subset selection from principal components search space.

GA Algorithm

Step 1. (Start)

Generate random population of n chromosomes.

Step 2. (Fitness)

Evaluate the fitness f (x) of each chromosome x in the population.

a. (New population) Create a new population by repeating following steps:
b. (Selection) Select two parent chromosomes from a population.
c. (Crossover) With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
d. (Mutation) With a mutation probability, mutate new offspring at each locus (position in chromosome).
e. (Accepting) Place new offspring in a new population.

Step 3 (Replace)

Use new generated population for a further run of algorithm.

Step 4 (Test)

If the end condition is satisfied, stop, and return the best solution in current population.

Step 5 (Loop)

Go to step 2. The working flow of GA is shown in Figure 3. We used the fitness function shown to combine the two terms:

$$fitness = 10^4 Accuracy + 0.5 Zeros \quad (4)$$

Where Accuracy corresponds to the classification accuracy on a validation set for a particular subset of principal components and zeros corresponds to the number principal components not selected. The accuracy term ranges roughly from 0.50 to 0.99, thus, the first term assumes values from 5000 to 9900. The zeros term ranges from 0 to L − 1 where L is the length of the chromosome, thus, the second term assumes values from 0 to 37 (L = 38).

Classification architecture

A multilayer perceptron (MLP) is a feedforward neural network that maps sets of input data onto a set of appropriate output. Here, we used a MLP architecture consists of three layers; input, hidden and output. In this architecture, hidden layer and output layer consist of neurons (processing elements) and each neuron has a nonlinear activation function. The layers are fully connected from one layer to the next. MLP is an amendment of the standard linear perceptron, which can discriminate data that is not linearly separable. The architecture we used here is shown in Figure 4. The overall performance of MLP with 12, 20 and 27 features are shown in Table 4.

Training and testing of the system

The aim of training is the adjustment of networks weights on base of the difference between the output produced by the system and the desired output. The training dataset consists of five thousand (5000) labelled connections (network packets with label as normal or intrusive) that are randomly selected from 20,000 connections. Further, we divide the training dataset (five thousand) into three parts; (i) cross validation dataset (1000), (ii) test dataset (1500) and (iii) training dataset (2500).

We used confusion matrix to verify the training. When the training is completed then weights of the system are frozen and performance of the system is evaluated. Testing the system involves two steps; (i) verification step, and (ii) generalization step (Ahmad et al., 2011c). In the verification step, the system is tested against the data which are used in training. Aim of the verification step is to test how well trained system learned the training patterns in the training dataset. In generalization step, testing is conducted with data which is not used in training. Aim of the generalization step is to measure generalization ability of the trained network. We used a dataset of fifteen thousand (15,000) as a production dataset. We also tested our system performance on total dataset (20,000) that consist of both training dataset and production dataset.

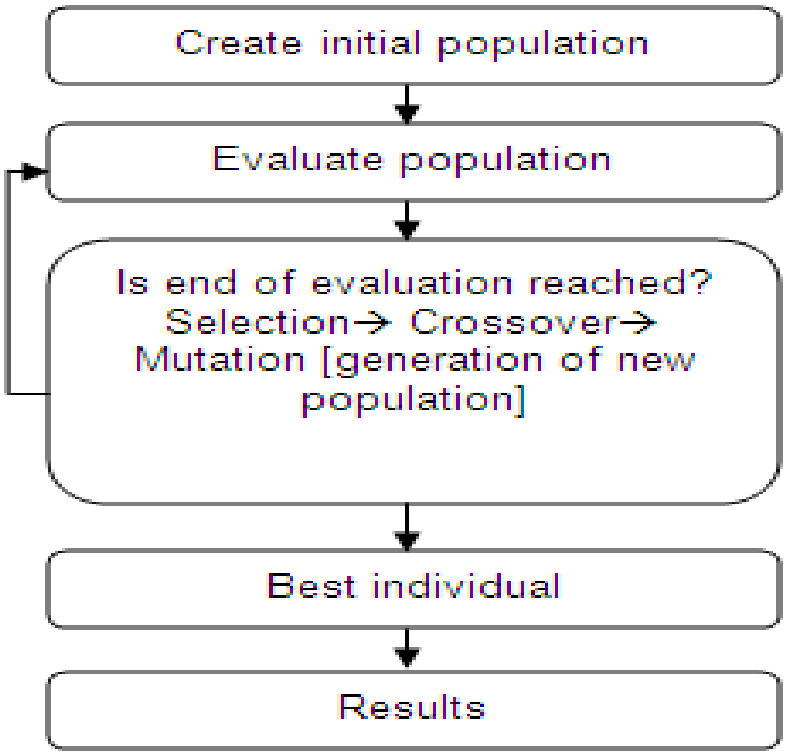

Figure 3. GA flow chart.

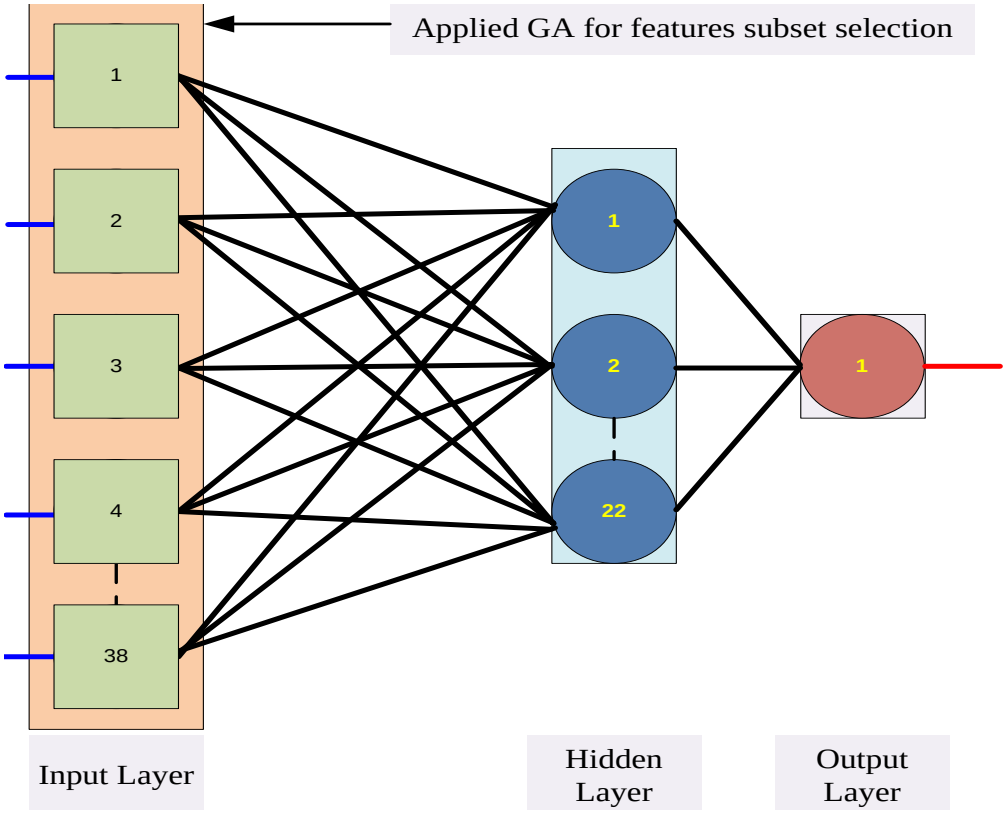

Figure 4. MLP for Intrusion Analysis

Table 1. Experimental results.

Expt #	Time (h)	No. of PCs	No of non selected PCs	Accuracy	Fitness
1-MLP	72	12	26	0.99	9913
2-MLP	78	20	18	0.98	9808
3-MLP	83	27	11	0.99	9911

Table 2. GA features.

Feature #	MLP(12)	MLP(20)	MLP(27)
1	X	√	√
2	√	√	X
3	√	√	X
4	X	X	√
5	X	√	√
6	X	X	√
7	X	√	√
8	X	√	√
9	√	X	X
10	X	√	√
11	√	√	X
12	√	√	√
13	X	X	X
14	X	X	√
15	√	√	√
16	X	X	X
17	√	X	√
18	√	X	X
19	X	√	X
20	X	√	√
21	X	X	√
22	X	X	√
23	X	X	X
24	√	√	X
25	X	√	√
26	X	X	√
27	√	X	√
28	X	√	√
29	X	√	√
30	X	X	X
31	X	X	√
32	X	X	√
33	X	√	√
34	√	X	√
35	X	X	√
36	√	√	√
37	X	√	√
38	X	√	√

Table 3. Parameter used during feature selection.

S/no.	Genetic operator(s)	Genetic operator value(s)
1	Maximum generation	100
2	Chromosomes	50
3	Selection method	Top percent (10%)
4	Crossover	One-point
5	Crossover probability	0.9
6	Mutation probability	0.01
7	Population size	50
8	Termination type	Fitness threshold (0.001)
9	Architecture	MLP
10	Training algorithm	Online back propagation

Table 4. MLP performance in different experiments.

Classifier	MLP-12	MLP-20	MLP-27
False alarm	03	13	09
Epochs	217	1000	1000
Time	00:23:00	01:09:08	01:29:07
Features	240000	440000	760000
False +	03	13	09
False -	0	0	0
True +	12797	12789	12793
True -	7203	7211	7207

RESULTS

We performed three different experiments as shown in Table 1 and selected a subset of twelve features that indicates better performance as compared to other subsets as shown in Table 2. Our aim is to select minimum features that produce optimal results in accuracy. This definitely impact on overall performance of the system. The features are reduced to 12 from the 41 raw features set. The above experiments show that optimal features increased accuracy, reduced training and computational overheads and simplified the architecture of intrusion analysis engine. This is extended work of our previous work (Ahmad et al., 2011c). Parameters used for genetic feature subset selection is shown in Table 3.

ACKNOWLEDGEMENT

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

REFERENCES

Ahmad I, Abdullah AB, Alghamdi AS (2010). Towards the selection of best neural network system for intrusion detection. Int. J. Phys. Sci., 5(12): 1830-1839.

Ahmad I, Abdullah AB, Alghamdi AS, Hussain M (2011a). Distributed Denial of Service Attacks Detection using Support Vector Machine. INFORMATION An Int. Interdisciplinary J., 14(1): 127-134.

Ahmad I, Abdullah AB, Alghamdi AS, Hussain M (2011b). Optimized intrusion detection mechanism using soft computing techniques, Telecommun. Syst., 48(1-2):1-9.

Ahmad I, Abdullah AB, Alghamdi AS, Hussain M, Nafjan K (2011c). Features Subset Selection for Network Intrusion Detection Mechanism Using Genetic Eigen Vectors. Proceedings of 2011 International Conference on Telecommunication Technology and Applications (ICTTA 2011), pp.75-79.

Kim DS, Nguyen HN, Ohn SY, Park JS (2005). Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System. Advances in Neural Networks. Lecture Notes in Computer Science: 3498/2005, 415-420.

Zargar GR, Kabiri P (2010). Selection of Effective Network Parameters in Attacks for Intrusion Detection. Advances in Data Mining. Applications and Theoretical Aspects. Lecture Notes Comput. Sci., 6171/2010: 643-652.

Liu G, Zhang YI, Yang S (2007). A hierarchical intrusion detection model based on the PCA neural networks. Neurocomputing, 70(7-9): 1561-1568.

Horng SJ, Yang SU, Chen YH, Kao TW, Chen RJ, Lai JL, Perkasa CD (2010). A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst. Appl., 38(1): 306-313.

Tong X, Wang Z, Haining YU (2009). A research using hybrid RBF/Elman neural network for intrusion detection system secure model. Comput. Phys. Commun., 180(10): 1795-1801.

Sun Z, Bebis G, Miller R (2002). Object detection using feature subset selection. Patt. Recog., 37(11): 165-2176.