Mining Audit Data for Intrusion Detection Systems Using Support Vector Machines and Neural Networks

Ramamoorthy Subbureddiar, Srinivas Mukkamala, Madhukumar Shankarpani, Andrew H. Sung
Department of Computer Science
Sathyabama University, New Mexico Tech
Chennai, Tamilnadu | Socorro, New Mexico 87801

Abstract

This paper concerns using learning machines for intrusion detection. Two classes of learning machines are studied: Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). We show that SVMs are superior to ANNs for intrusion detection in three critical respects: SVMs train, and run, an order of magnitude faster; SVMs scale much better; and SVMs give higher classification accuracy. We also address the related issue of ranking the importance of input features, which is itself a problem of great interest in modeling. Since elimination of the insignificant and/or useless inputs leads to a simplification of the problem and possibly faster and more accurate detection, feature selection is very important in intrusion detection. Two methods for feature ranking are presented: the first one is independent of the modeling tool, while the second method is specific to SVMs. The two methods are applied to identify the important features in the 1999 DARPA intrusion data. It is shown that the two methods produce results that are largely consistent. We present various experimental results that indicate that SVM-based intrusion detection using a reduced number of features can deliver enhanced or comparable performance. An SVM-based IDS for class-specific detection is thereby proposed.

Key words: Intrusion detection, Feature selection, Machine learning

I. INTRODUCTION

This paper concerns intrusion detection and the related issue of identifying important input features for intrusion detection. Intrusion detection is a problem of great significance to critical infrastructure protection owing to the fact that computer networks are at the core of the nations operational control. We use two types of learning machines to build Intrusion Detection Systems (IDSs): Artificial Neural Networks or ANNs (1) and Support Vector Machines or SVMs (2). Since the ability to identify the important inputs and redundant inputs of a classifier leads directly to reduced size, faster training and possibly more accurate results, it is critical to be able to identify the important features of network traffic data for intrusion detection in order for the IDS to achieve maximal performance. Therefore, we also study feature ranking and selection, which is itself a problem of great interest in building models based on experimental data.

Since most of the intrusions can be uncovered by examining patterns of user activities, many IDSs have been built by utilizing the recognized attack and misuse patterns to develop learning machines (3,4,5,6,7,8,9,10,11). In our recent work, SVMs are found to be superior to ANNs in many important respects of intrusion detection (12,13,14); we will concentrate on SVMs and briefly summarize the results of ANNs.

The data we used in our experiments originated from MIT's Lincoln Lab. It was developed for intrusion detection system evaluations by DARPA and is considered a benchmark for intrusion detection evaluations (15).

We performed experiments to rank the importance of input features for each of the five classes (normal, probe, denial of service, user to super-user, and remote to local) of patterns in the DARPA data. It is shown that using only the important features for classification gives good accuracies and, in certain cases, reduces the training time and testing time of the SVM classifier.

In the rest of the paper, a brief introduction to the data we used is given in section 2. In section 3 we describe the method of deleting one input feature at a time and the performance metrics considered for deciding the importance of a particular feature. In section 4 we present the experimental results of using SVMs for feature ranking. In section 5 we present the experimental results of using ANNs for feature ranking. In section 6 we summarize our results and give a brief description of our proposed IDS architecture.

II. THE DATA

In the 1998 DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted (16). Of this database a subset of 494021 data were used, of which 20% represent normal patterns.

Attack types fall into four main categories:

1. DOS: denial of service
2. R2L: unauthorized access from a remote machine
3. U2Su: unauthorized access to local super user (root) privileges
4. Probing: surveillance and other probing

A. Denial of Service Attacks

A denial of service attack is a class of attacks in which an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm.

B. User to Root Attacks

User to root exploits are a class of attacks in which an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Examples are Eject, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm.

C. Remote to User Attacks

A remote to user attack is a class of attacks in which an attacker sends packets to a machine over a networkbut who does not have an account on that machine; exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop.

D. Probing

Probing is a class of attacks in which an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Nmap, Saint, Satan.

III. RANKING THE SIGNIFICANCE OF INPUTS

Feature selection and ranking (17,18) is an important issue in intrusion detection. Of the large number of features that can be monitored for intrusion detection purpose, which are truly useful, which are less significant, and which may be useless? The question is relevant because the elimination of useless features (or audit trail reduction) enhances the accuracy of detection while speeding up the computation, thus improving the overall performance of an IDS. In cases where there are no useless features, by concentrating on the most important ones we may well improve the time performance of an IDS without affecting the accuracy of detection in statistically significant ways.

The feature ranking and selection problem for intrusion detection is similar in nature to various engineering problems that are characterized by:

Having a large number of input variables x = (x1, x2,_,_,_ xn) of varying degrees of importance; i.e., some elements of x are essential, some are less important, some of them may not be mutually independent, and some may be useless or noise

Lacking an analytical model or mathematical formula that precisely describes the input-output relationship,

$$Y = F(x).$$

Having available a finite set of experimental data, based on which a model (e.g. neural networks) can be built for simulation and prediction purposes Due to the lack of an analytical model, one can only seek to determine the relative importance of the input variables through empirical methods. A complete analysis would require examination of all possibilities, e.g., taking two variables at a time to analyze their dependence or correlation, then taking three at a time, etc. This, however, is both infeasible (requiring 2n experiments!) and not infallible (since the available data may be of poor quality in sampling the whole input space). In the following, therefore, we apply the technique of deleting one feature at a time (14) to rank the input features and identify the most important ones for intrusion detection using SVMs (19).

A. Performance-Based Method for Ranking Importance

We first describe a general (i.e., independent of the modeling tools being used), performance-based input ranking methodology: One input feature is deleted from the data at a time, the resultant data set is then used for the training and testing of the classifier. Then the classifier's performance is compared to that of the original classifier (based on all features) in terms of relevant performance criteria. Finally, the importance of the feature is ranked according to a set of rules based on the performance comparison.

The procedure is summarized as follows:

1. Compose the training set and the testing set; for each feature do the following
2. Delete the feature from the (training and testing) data;
3. Use the resultant data set to train the classifier;
4. Analyze the performance of the classifier using the

test set, in terms of the selected performance criteria;

5. Rank the importance of the feature according to the rules;

B. Performance Metrics

To rank the importance of the 41 features (of the DARPA data) in an SVM-based IDS, we consider three main performance criteria: overall accuracy of (5-class) classification; training time; and testing time. Each feature will be ranked as "important", "secondary", or "insignificant", according to the following rules that are applied to the result of performance comparison of the original 41-feature SVM and the 40-feature SVM:

Rule set:

1. If accuracy decreases and training time increases and testing time decreases, then the feature is important
2. If accuracy decreases and training time increases and testing time increases, then the feature is important
3. If accuracy decreases and training time decreases and testing time increases, then the feature is important
4. If accuracy unchanges and training time increases and testing time increases, then the feature is important
5. If accuracy unchanges and training time decreases and testing time increases, then the feature is secondary
6. If accuracy unchanges and training time increases and testing time decreases, then the feature is secondary
7. If accuracy unchanges and training time decreases and testing time decreases, then the feature is unimportant
8. If accuracy increases and training time increases and testing time decreases, then the feature is secondary
9. If accuracy increases and training time decreases and testing time increases, then the feature is secondary
10. If accuracy increases and training time decreases and testing time decreases, then the feature is unimportant

According to the above rules, the 41 features are ranked into the 3 types of {Important}, <Secondary>, or (Unimportant), for each of the 5 classes of patterns, as follows:

class 1 Normal: {1,3,5,6,8-10,14,15,17,20-23,25, 29, 33,35,36,38,39,41}, <2,4,7,11,12,16,18,19,24,30, 31 34,37,40>, (13,32)

class 2 Probe: {3,5,6,23,24,32,33}, <1,4,7-9,12-19, 21,22,25-28,34-41>, (2,10,11,20,29,30,31,36,37)

class 3 DOS: {1,3,5,6,8,19,23-28,32,33,35,36,38-41}, <2,7,9-11,14,17,20,22,29,30,34,37>, (4,12,13,15,16, 18,19,21,3)

class 4 U2Su: {5,6,15,16,18,32,33}, <7,8,11,13,17, 19-24,26,30,36-39>, (9,10,12,14,27,29,31,34,35, 40, 41)

class 5: {3,5,6,24,32,33}, <2,4,7-23,26-31,34-41>, (1,20,25,38)

Because SVMs are only capable of binary classifications, we will need to employ five SVMs for the five-class identification problem in intrusion detection. But since the set of important features may differ from class to class, using five SVMs becomes an advantage rather than a hindrance, i.e., in building an IDS using five SVMs, each SVM can use only the important features for that class which it is responsible for making classifications.

C. SVM-specific Feature Ranking Method

Information about the features and their contribution towards classification is hidden in the support vector decision function. Using this information one can rank their significance, i.e., in the equation

$$F(X) = ? WiXi + b$$

The point X belongs to the positive class if F(X) is a positive value. The point X belongs to the negative class if F(X) is negative. The value of F(X) depends on the contribution of each value of X and Wi. The absolute value of Wi measures the strength of the classification. If Wi is a large positive value then the ith feature is a key factor for positive class. If Wi is a large negative value then the ith feature is a key factor for negative class. If Wi is a value close to zero on either the positive or the negative side, then the ith feature does not contribute significantly to the classification. Based on this idea, a ranking can be done by considering the support vector decision function.

D. Support Vector Decision Function (SVDF)

The input ranking is done as follows: First the original data set is used for the training of the classifier. Then the classifier's decision function is used to rank the importance of the features. The procedure is: Calculate the weights from the support vector decision function; Rank the importance of the features by the absolute values of the weights;

According to the ranking method, the 41 features are placed into the 3 categories of {Important}, <Secondary> or

(Unimportant), for each of the 5 classes of patterns, as follows:

class 1 Normal:

{1,2,3,4,5,6,10,12,17,23,24,27,28,29,31,32,33,34,36,39}, <11,13,14,16,19,22,25,26,30,35,37,38,40,41>, (7,8,9,15,18,20,21)

class 2 Probe: {1,2,3,4,5,6,23,24,29,32,33}, <10,12, 22,28, 34,35,36,38,39,41 >, (7,8,9,11,13,14, 15,16, 17,18, 19 20,21,25,26,27,30,31,37,40)

class 3 DOS: {1,5,6,23,24,25,26,32,36,38,39}, <2,3, 4,10, 12,29,33,34 >(7,8,9,11,13,14,15,16, 17 18,19, 20,21, 22,27,28,30,31,35,36,37,40,41)

class 4 U2Su: {1,2,3,5,6,12,23,24,32,33}, <4,10,13, 14,17 22,27,29,31,34,36,37,39 > (7,8,9,11,15, 16,18, 19,20, 21,25,26,28,30,35,38,40,41)

class 5 R2L: {1,3,5,6,32,33}, <2,4,10,12,22,23,24, 29,31,34,36,37,38,40 >, (7,8,9,11,13,14,15,16,17, 18, 19,20,21,25,26,27,28,30,35,39,41)

IV. EXPERIMENTS

SVMs are used, in each of the two methods, for ranking the importance of the input features. Once the importance of the input features was ranked, the classifiers were trained and tested with only the important features. Further, we validate the ranking by comparing the performance of the classifier (20,21) using all input features to that using the important features; and we also compare the performance of a classifier using the union of the important features for all fives classes.

A. SVM Performance Statistics

Our results are summarized in the following tables. Table 1 gives the performance results of the five SVMs for each respective class of data. Table 2 shows the results of SVMs performing classification, with each SVM using as input the important features for all five classes. Table 3 shows the results of SVMs performing classification, with each SVM using as input the union of the important features for all five classes. Table 4 shows the result of SVMs performing classification, with each SVM using as input the important and secondary features for each respective class. Table 5 shows the results of SVMs performing classification, with each SVM using as input the important features obtained from the SVDF ranking. Table 6 shows the results of SVMs performing classification, with each SVM using as input the union of the important features for each class as obtained from the SVDF ranking; the union has 23 features. Table 7 shows the result of SVMs performing classification, with each SVM using as input the important and secondary features for each respective class.

Table1: Performance of SVMs using 41 features

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.66	1.26	99.55
Probe	49.13	2.10	99.70
DOS	22.87	1.92	99.25
U2Su	3.38	1.05	99.87
R2L	11.54	1.02	99.78

Table2: Performance of SVMs using important features

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy
Normal	25	9.36	1.07	99.59
Probe	7	37.71	1.87	99.38
DOS	19	22.79	1.84	99.22
U2Su	8	2.56	0.85	99.87
R2L	6	8.76	0.73	99.78

Table3: Performance of SVMs using union of important features (30)

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.67	1.02	99.51
Probe	44.38	2.07	99.67
DOS	18.64	1.41	99.22
U2Su	3.23	0.98	99.87
R2L	9.81	1.01	99.78

Table4: Performance of SVMs using important and secondary features

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	39	8.15	1.22	99.59
Probe	32	47.56	2.09	99.65
DOS	32	19.72	2.11	99.25
U2Su	25	2.72	0.92	99.87
R2L	37	8.25	1.25	99.80

Table 5. Performance of SVMs using important features as ranked by SVDF

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	20	4.58	0.78	99.55
Probe	11	40.56	1.20	99.36
DOS	11	18.93	1.00	99.16
U2Su	10	1.46	0.70	99.87
R2L	6	6.79	0.72	99.72

Table 6. Performance of SVMs using union of important features (total 23) as ranked by SVDF

Class	Training time	Testing time	Accuracy
Normal	4.85	0.82	99.55%
Probe	36.23	1.40	99.71%
DOS	7.77	1.32	99.20%
U2Su	1.72	0.75	99.87%
R2L	5.91	0.88	99.78%

Table7. Performance of SVMs using important and secondary features using SVDF

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	34	4.61	0.97	99.55
Probe	21	39.69	1.45	99.56
DOS	19	73.55	1.50	99.56
U2Su	23	1.73	0.79	99.87
R2L	20	5.94	0.91	99.78

V. NEURAL NETWORK EXPERIMENTS

This section summarizes the authors' recent work in comparing ANNs and SVMs for intrusion detection (10,11,12). Since a (multi-layer feed forward) ANN is capable of making multi-class classifications, a single ANN is employed to perform the intrusion detection, using the same training and testing sets as those for the SVMs.

Neural networks are used for ranking the importance of the input features, taking training time, testing time, and classification accuracy as the performance measure; and a set of rules is used for ranking. Therefore, the method is an extension of the feature ranking method described in (17) where cement bonding quality problem is used as the engineering application. Once the importance of the input feature was ranked, the ANNs are trained and tested with the data set containing only the important features. We then compare the performance of the trained classifier against the original ANN trained with data containing all input features.

A. ANN Performance Statistics

Table 13 in the appendix gives the results of 42 experiments in ranking the input features: the performance statistics of the original ANN with 41 features, and the performance of the 41 ANNs, each with 40 features. Table 7 below gives the comparison of the ANN with all 41 features to that of using 34 important features that have been obtained by our feature-ranking algorithm described above.

Table 7. Neural network results using all 34 important features

Number of features	Accuracy	False positive rate	False negative rate	Number of epochs
41	87.07	6.66	6.27	412
34	81.57	18.19	0.25	27

VI. SUMMARY & CONCLUSIONS

- A number of observations and conclusions are drawn from the results reported:
- SVMs outperform ANNs in the important respects of scalability (SVMs can train with a larger number of patterns, while would ANNs take a long time to train or fail to converge at all when the number of patterns gets large); training time and running time (SVMs run an order of magnitude faster); and prediction accuracy.
- SVMs easily achieve high detection accuracy (higher than 99%) for each of the 5 classes of data, regardless of whether all 41 features are used, only the important features for each class are used, or the union of all important features for all classes are used.

We note, however, that the difference in accuracy figures tend to be very small and may not be statistically significant, especially in view of the fact that the 5 classes of patterns differ in their sizes tremendously. More definitive conclusions can only be made after analyzing more comprehensive sets of network traffic data.

Regarding feature ranking, we observe that

- The two feature ranking methods produce largely consistent results: except for the class 1 (Normal) and class 4 (U2Su) data, the features ranked as Important by the two methods heavily overlap.
- The most important features for the two classes of 'Normal' and 'DOS' heavily overlap.
- 'U2Su' and 'R2L', the two smallest classes representing the most serious attacks, each has a small number of important features and a large number of secondary features.
- The performances of (a) using the important features for each class, Table 2, Table 5, (b) using the union of important features, Table 3, Table 6, and (c) using the union of important and secondary features for each class Table 4 and Table 7, do not show significant differences, and are all similar to that of using all 41 features.
- Using the important features for each class gives the most remarkable performance: the testing time decreases in each class; the accuracy increases slightly for one class 'Normal', decreases slightly for two classes 'Probe' and 'DOS', and remains the same for the two most serious attack classes.

Our ongoing experiments include making 23-class (22 specific attacks and normal) feature identification using SVMs, for the purpose of designing a cost-effective and real time intrusion detection tool. Finally, we propose a five SVM based intrusion detection architecture, where we use the set of important features for each class that are responsible for making classifications.

VII. ACKNOWLEDGEMENTS

Support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech) and a U.S. Department of Defense IASP capacity building grant is gratefully acknowledged. We would also like to acknowledge many insightful conversations with Dr. Jean-Louis Lassez and David Duggan that helped clarify some of our ideas.

Table 8. List of features [16]. Type C is continuous, while D is discrete.

#	Feature name	Description	Type
1	duration	Length (# of seconds) of the connection	C
2	protocol type	Type of the protocol, e.g. tcp, udp, etc.	D
3	service	Network service on the destination, e.g., http, telnet, etc.	D
4	flag	Normal or error status of the connection	D
5	src_bytes	# of data bytes from source to destination	C
6	dst_bytes	# of data bytes from destination to source	C
7	land	1 if connection is from/to the same host/port; 0 otherwise	D
8	wrong_fragment	# of "wrong" fragments	C
9	urgent	# of urgent packets	C
10	hot	# of "hot" indicators	C
11	num_failed_logins	# of failed login attempts	C
12	logged in	1 if successfully logged in; 0 otherwise	D
13	num_compromised	# of compromised conditions	C
14	root_shell	1 if root shell is obtained; 0 otherwise	D
15	su_attempted	1 if "su root" command attempted; 0 otherwise	D
16	num_root	# of "root" accesses	C
17	num_file_creations	# of file creation operations	C
18	num_shells	# of shell prompts	C
19	num_access_files	# of operations on access control files	C
20	num_outbound_cmds	# of outbound commands in an ftp session	C
21	is_host_login	1 if the login belongs to the "hot" list; 0 otherwise	D
22	is_guest_login	1 if the login is a "guest' login; 0 otherwise	D
23	count	# connections to the same host as the current one during past two seconds	C

24	srv_count	# of connections to the same service as the current connection in the past two seconds	C
25	serror_rate	% of connections that have “SYN” errors	C
26	srv_serror_rate	% of connections that have “SYN” errors	C
27	rerror_rate	% of connections that have “REJ” errors	C
28	srv_rerror_rate	% of connections that have “REJ” errors	C
29	same_srv_rate	% of connections to the same service	C
30	diff_srv_rate	% of connections to different services	C
31	srv_diff_host_rate	% of connections to different hosts	C
32	dst_host_count		C
33	dst_host_srv_count		C
34	dst_host_same_srv_rate		C
35	dst_host_diff_srv_rate		C
36	dst_host_same_src_port_rate		C
37	dst_host_srv_diff_host_rate		C
38	dst_host_serror_rate		C
39	dst_host_srv_serror_rate		C
40	dst_host_rerror_rate		C
41	dst_host_srv_rerror_rate		C

Table 9. Class 1, Normal

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy
None	7.66	1.26	99.55
1.	10.19	1.11	99.51
2.	6.56	1.46	99.55
3.	9.06	1.47	99.48
4.	9.96	1.08	99.55
5.	33.11	1.62	99.19
6.	7.56	1.79	98.75
7.	7.11	1.43	99.55
8.	8.33	1.41	99.55
9.	8.37	1.37	99.55
10.	8.68	1.35	99.55
11.	7.49	1.33	99.55
12.	8.01	1.38	99.55
13.	7.14	0.81	99.55
14.	8.00	1.46	99.55
15.	9.81	1.43	99.55
16.	8.15	1.04	99.55
17.	8.12	1.47	99.55
18.	7.36	1.30	99.55
19.	8.00	1.12	99.55
20.	8.15	1.38	99.55
21.	7.98	1.42	99.55
22.	8.12	1.43	99.55
23.	7.65	1.34	99.56
24.	7.29	1.30	99.55
25.	8.32	1.35	99.55
26.	7.71	1.30	99.55
27.	7.73	1.38	99.55
28.	7.90	1.47	99.55
29.	7.81	1.39	99.55
30.	7.57	1.38	99.55
31.	7.11	1.30	99.55
32.	6.17	1.26	99.55
33.	8.53	1.51	99.48
34.	7.23	1.48	99.55
35.	6.96	1.35	99.55
36.	10.19	1.36	99.55
37.	6.74	1.33	99.55
38.	8.17	1.43	99.55
39.	7.75	1.32	99.55
40.	7.20	1.45	99.55
41.	9.38	1.43	99.55

Table10. Class 2, Probe

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy
None	49.13	2.10	99.70
1.	58.93	2.01	99.70
2.	44.07	1.79	99.70
3.	51.00	2.19	99.61
4.	62.42	1.85	99.70
5.	75.67	1.97	98.14
6.	51.03	1.17	99.52
7.	51.62	1.98	99.70
8.	55.34	1.88	99.72
9.	53.05	1.99	99.70
10.	46.29	2.00	99.70
11.	45.68	1.96	99.70
12.	53.18	1.95	99.70
13.	55.27	1.95	99.70
14.	50.67	1.92	99.70
15.	49.50	2.07	99.70
16.	47.61	2.16	99.70
17.	49.38	1.93	99.70
18.	50.28	1.91	99.70
19.	50.33	1.94	99.70
20.	48.61	1.93	99.70
21.	50.40	1.89	99.70
22.	51.50	1.96	99.70
23.	49.00	2.63	99.46
24.	42.86	1.97	99.61
25.	52.40	1.95	99.71
26.	52.42	1.99	99.71
27.	62.51	2.05	99.71
28.	71.80	1.91	99.71
29.	45.95	1.78	99.70
30.	46.62	2.00	99.70
31.	46.35	1.93	99.70
32.	31.89	1.82	99.67
33.	50.90	1.83	99.62
34.	47.64	1.30	99.70
35.	49.49	1.87	99.70
36.	47.39	1.97	99.70
37.	48.19	2.03	99.70
38.	57.51	1.85	99.71
39.	52.54	1.94	99.71
40.	56.45	1.98	99.70
41.	51.66	1.71	99.70

Table11. Class 3, Denial of Service

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy
None	22.87	1.92	99.25
1.	21.76	1.87	99.23
2.	23.60	1.89	99.25
3.	17.88	2.03	99.10
4.	20.00	1.79	99.25
5.	39.57	1.61	97.55
6.	19.63	0.84	98.07
7.	23.76	1.87	99.25
8.	31.23	1.86	99.20
9.	23.80	1.78	99.25
10.	27.01	1.82	99.25
11.	22.03	1.86	99.25
12.	19.69	1.84	99.25
13.	21.30	1.93	99.25
14.	20.18	2.02	99.25
15.	18.76	1.89	99.25
16.	21.56	1.78	99.25
17.	22.98	2.09	99.25
18.	21.47	1.95	99.25
19.	20.79	1.97	99.25
20.	21.49	1.96	99.25
21.	21.75	1.94	99.25
22.	24.93	2.01	99.25
23.	23.94	3.01	98.58
24.	25.43	2.05	99.20
25.	21.70	1.80	99.19
26.	25.93	1.98	99.19
27.	24.21	1.41	99.20
28.	26.16	1.80	99.20
29.	29.99	1.93	99.25
30.	18.27	1.79	99.20
31.	19.85	1.79	99.25
32.	11.70	0.95	98.69
33.	44.19	1.74	99.19
34.	28.27	1.88	99.25
35.	28.94	1.75	99.22
36.	27.39	1.80	99.22
37.	22.40	1.86	99.25
38.	22.45	1.95	99.19
39.	23.81	1.92	99.20
40.	50.15	1.84	99.22
41.	25.36	2.03	99.19

Table12. Class 4, User to Root

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy
None	3.38	1.05	99.87
1.	2.98	0.96	99.87
2.	3.35	0.98	99.87
3.	3.00	1.04	99.87
4.	3.21	1.04	99.87
5.	3.11	0.65	99.72
6.	1.99	0.18	88.81
7.	3.40	1.07	99.87
8.	3.43	1.10	99.87
9.	3.37	0.97	99.87
10.	3.69	0.97	99.87
11.	3.47	1.06	99.87
12.	3.36	0.99	99.87
13.	3.61	1.01	99.87
14.	3.12	1.02	99.87
15.	3.40	1.11	99.87
16.	3.57	1.14	99.87
17.	3.39	0.98	99.87
18.	3.46	1.07	99.87
19.	3.41	1.05	99.87
20.	3.35	1.10	99.87
21.	3.34	1.08	99.87
22.	3.26	1.07	99.87
23.	3.39	1.05	99.87
24.	3.32	1.07	99.87
25.	3.44	1.09	99.87
26.	3.38	1.06	99.87
27.	3.36	1.05	99.87
28.	3.23	1.00	99.87
29.	3.36	0.98	99.87
30.	3.42	0.98	99.87
31.	3.34	1.00	99.87
32.	3.95	0.92	99.84
33.	4.58	0.99	99.85
34.	3.36	1.02	99.87
35.	2.98	1.05	99.87
36.	3.50	1.05	99.87
37.	3.43	1.00	99.87
38.	3.79	1.05	99.87
39.	3.27	1.07	99.87
40.	3.36	0.99	99.87
41.	3.36	1.01	99.87

Table13. Class 5, Remote to Local

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy
None	11.54	1.02	99.78
1.	7.54	1.04	99.80
2.	8.79	1.23	99.78
3.	9.95	1.11	99.75
4.	8.56	1.26	99.78
5.	12.11	1.79	99.06
6.	16.52	0.63	98.88
7.	10.18	1.34	99.78
8.	9.59	1.31	99.78
9.	8.41	1.23	99.78
10.	9.30	1.32	99.78
11.	10.21	1.23	99.78
12.	9.48	1.33	99.78
13.	9.88	1.29	99.78
14.	8.84	1.22	99.78
15.	9.25	1.28	99.78
16.	8.89	1.20	99.78
17.	9.21	1.24	99.78
18.	9.60	1.30	99.78
19.	10.15	1.30	99.78
20.	10.68	0.99	99.78
21.	10.99	1.26	99.78
22.	10.88	1.26	99.78
23.	8.19	1.26	99.78
24.	7.67	1.22	99.72
25.	9.26	1.05	99.78
26.	10.11	1.30	99.78
27.	9.09	1.24	99.78
28.	9.10	1.23	99.78
29.	11.39	1.11	99.78
30.	10.64	1.26	99.78
31.	8.56	1.26	99.78
32.	11.55	1.05	99.80
33.	12.35	1.25	99.80
34.	10.59	1.14	99.78
35.	9.07	1.18	99.78
36.	9.22	1.22	99.78
37.	9.33	1.30	99.78
38.	8.98	0.95	99.78
39.	8.52	1.26	99.78
40.	8.98	1.11	99.78
41.	10.35	1.26	99.78

Table14. Neural network feature ranking results

Feature deleted	Accuracy	False positive rate	False negative rate	Number of epochs
All	**87.07**	**6.66**	**6.27**	412
1	91.57	7.36	1.07	400
2	77.92	21.22	0.86	420
3	80.68	16.50	2.82	473
4	90.16	9.13	0.71	312
5	90.16	8.88	0.96	438
6	77.23	22.13	0.64	339
7	76.87	22.06	1.07	419
8	72.98	26.28	0.74	389
9	84.89	14.40	0.71	298
10	54.08	45.11	0.81	385
11	75.81	23.79	0.41	331
12	81.64	17.84	0.52	471
13	69.40	4.82	25.78	406
14	71.39	6.14	22.47	494
15	71.93	3.99	24.08	389
16	77.50	4.89	17.61	351
17	75.60	4.21	20.19	377
18	72.09	3.47	24.44	388
19	85.36	4.45	10.29	421
20	79.09	20.30	0.61	314
21	79.09	20.30	0.61	324
22	89.20	9.94	0.86	591
23	62.38	36.76	0.86	379
24	89.16	10.06	0.78	368
25	78.45	20.61	0.94	420
26	77.58	21.61	0.81	427
27	79.03	20.48	0.49	380
28	75.43	24.12	0.45	358
29	94.82	4.21	0.97	345
30	78.01	20.60	1.39	301
31	89.13	10.20	0.67	393
32	82.71	16.36	0.93	398
33	58.72	40.25	1.03	418
34	75.24	23.88	0.89	511
35	53.08	46.28	0.64	436
36	76.62	22.70	0.68	459
37	72.98	26.49	0.54	349
38	74.06	24.82	1.12	387
39	76.42	23.28	0.30	380
40	73.54	26.02	0.44	335
41	74.50	24.70	0.80	402

REFERENCES

[1]. Hertz J., Krogh A., Palmer, R. G. (1991) Introduction to the Theory of Neural Computation, Addison Wesley.

[2]. Joachims T. (1998) Making Large-Scale SVM Learning Practical. LS8-Report, University of Dortmund, LS VIII-Report.

[3]. Denning D. (Feb. 1987) An Intrusion-Detection Model. IEEE Transactions on Software Engineering, Vol.SE-13, No 2.

[4]. Kumar S., Spafford E. H. (1994) An Application of Pattern Matching in Intrusion Detection. Technical Report CSD-TR-94-013. Purdue University.

[5]. Ghosh A. K. (1999). Learning Program Behavior Profiles for Intrusion Detection. USENIX.

[6]. Cannady J. (1998) Artificial Neural Networks for Misuse Detection. National Information Systems Security Conference.

[7]. Ryan J., Lin M-J., Miikkulainen R. (1998) Intrusion Detection with Neural Networks. Advances in Neural Information Processing Systems 10, Cambridge, MA: MIT Press.

[8]. Debar H., Becke M., Siboni D. (1992) A Neural Network Component for an Intrusion Detection System. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.

[9]. Debar H., Dorizzi. B. (1992) An Application of a Recurrent Network to an Intrusion Detection System. Proceedings of the International Joint Conference on Neural Networks, pp.78-83.

[10]. Luo J., Bridges S. M. (2000) Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection. International Journal of Intelligent Systems, John Wiley & Sons, pp.687-703.

[11]. Cramer M., et. al. (1995) New Methods of Intrusion Detection using Control-Loop Measurement. Proceedings of the Technology in Information

Security Conference (TISC) '95, pp.1-10.

[12]. Mukkamala S., Janoski G., Sung A. H. (2001) Monitoring Information System Security. Proceedings of the 11th Annual Workshop on Information Technologies & Systems, pp.139-144.

[13]. Mukkamala S., Janoski G., Sung A. H. (2002) Intrusion Detection Using Neural Networks and

Support Vector Machines. Proceedings of IEEE International Joint Conference on Neural Networks, pp.1702-1707.

[14]. Mukkamala S., Janoski G., Sung A. H. (2002) Comparison of Neural Networks and Support Vector Machines, in Intrusion Detection Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, June 11-13, 2002 http://www.mts.jhu.edu/…cidwkshop/abstracts.html

[15]. http://kdd.ics.uci.edu/databases/kddcup99/task.htm.

[16]. J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project by Salvatore

[17]. Sung A. H. (1998) Ranking Importance of Input Parameters Of Neural Networks. Expert Systems with Applications, pp.405-41.

[18]. Lin, Y., Cunningham, G. A. (1995) A New Approach to Fuzzy-Neural System Modeling. IEEE Transactions on Fuzzy Systems, Vol. 3, No. 2, pp.190-198.

[19]. Joachims T. (2000) SVMlight is an Implementation of Support Vector Machines (SVMs) in C. http://ais.gmd.de/…thorsten/svm_light. University of Dortmund. Collaborative Research Center on "Complexity Reduction in Multivariate Data" (SFB475).

[20]. Vladimir V. N. (1995) The Nature of Statistical Learning Theory. Springer.

[21]. Joachims T. (2000) Estimating the Generalization Performance of a SVM Efficiently. Proceedings of the International Conference on Machine Learning, Morgan Kaufman.APPENDIX