Matrix Factorization Approach for Feature Deduction and Design of Intrusion Detection Systems

Václav Snášel, Jan Platoš, Pavel Krömer
Department of Computer Science, FEECS
VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{vaclav.snasel, jan.platos, pavel.kromer.fei}@vsb.cz

Ajith Abraham
Center for Quantifiable Quality of Service
Norwegian University of Science and Technology,
Trondheim, Norway
ajith.abraham@ieee.org

Abstract

Current Intrusion Detection Systems (IDS) examine all data features to detect intrusion or misuse patterns. Some of the features may be redundant or contribute little (if anything) to the detection process. The purpose of this research is to identify important input features in building an IDS that is computationally efficient and effective. This paper propose a novel matrix factorization approach for feature deduction and design of intrusion detection systems. Experiment results indicate that the proposed method is efficient.

1 Introduction to Intrusion Detection Systems

Intrusion Detection Systems (IDS) were proposed to complement prevention-based security measures. An intrusion is defined to be a violation of the security policy of the system; intrusion detection thus refers to the mechanisms that are developed to detect violations of system security policy. Intrusion detection is based on the assumption that intrusive activities are noticeably different from normal system activities and thus detectable. Intrusion detection is not introduced to replace prevention-based techniques such as authentication and access control; instead, it is intended to complement existing security measures and detect actions that bypass the security monitoring and control component of the system.

Some specific examples of intrusions that concern system administrators include [5]:

- Unauthorized modifications of system files so as to facilitate illegal access to either system or user information.
- Unauthorized access or modification of user files or information.
- Unauthorized modifications of tables or other system information in network components (e.g. modifications of router tables in an internet to deny use of the network).
- Unauthorized use of computing resources (perhaps through the creation of unauthorized accounts or perhaps through the unauthorized use of existing accounts).

An IDS may be a combination of software and hardware. Most IDSs try to perform their task in real time. However, there are also IDSs that do not operate in real time, either because of the nature of the analysis they perform or because they are meant for forensic analysis (analysis of what happened in the past to a system). There are some intrusion detection systems that try to react when they detect an unauthorized action. This reaction usually includes trying to limit the damage, for example by terminating a network connection.

Since the amount of audit data that an IDS needs to examine is very large even for a small network, analysis is difficult even with computer assistance because extraneous features can make it harder to detect suspicious behavior patterns [15]. Audit data captures various features of the connections. For example, the audit data would show the source and destination bytes of a TCP connection, or the number of failed login attempts or duration of a connection. Complex relationships exist between the features, which are difficult for humans to discover. An IDS must therefore reduce the amount of data to be processed. This is very important if real-time detection is desired. Some data may not be useful to the IDS and thus can be eliminated before processing. In complex classification domains, features may contain false correlations, which hinder the process of detecting intrusions. Further, some features may be redundant since the information they add is contained in other features. Extra features can increase computation time, and can have an impact on the accuracy of the IDS. Feature selection improves classification by searching for the subset of features, which best classifies the training data [43].

In the literature several machine learning paradigms, fuzzy inference systems and expert systems, have been used to develop IDSs [15, 16]. The authors of [43] have demonstrated that a large number of features are unimportant and may be eliminated, without significantly lowering the performance of the IDS. The literature indicates very little scientific efforts aimed at modeling efficient IDS feature selection. The task of an IDS is often modeled as a classification problem in a machine-learning context. In this paper we investigate matrix factorization approach for selecting a subset of significant features from a feature set for network data. This reduced feature set is then employed in an ensemble design to implement an IDS. Experiment results indicate that the proposed approach is efficient.

2 Intrusion Detection Methods

The signatures of some attacks are known, whereas other attacks only reflect some deviation from normal patterns. Consequently, two main approaches have been devised to detect intruders.

2.1 Anomaly Detection

Anomaly detection assumes that an intrusion will always reflect some deviations from normal patterns. Anomaly detection may be divided into static and dynamic anomaly detection. A static anomaly detector is based on the assumption that there is a portion of the system being monitored that does not change. Usually, static detectors only address the software portion of a system and are based on the assumption that the hardware need not be checked. The static portion of a system is the code for the system and the constant portion of data upon which the correct functioning of the system depends. For example, the operating systems software and data to bootstrap a computer never change. If the static portion of the system ever deviates from its original form, an error has occurred or an intruder has altered the static portion of the system. Therefore Static anomaly detectors focus on integrity checking [12, 11]. Dynamic anomaly detection typically operate on audit records or on monitored networked traffic data. Audit records of operating systems do not record all events; they only record events of interest. Therefore only behavior that results in an event that is recorded in the audit will be observed and these events may occur in a sequence. In distributed systems, partial ordering of events is sufficient for detection. In other cases, the order is not directly represented; only cumulative information, such as cumulative processor resource used during a time interval, is maintained. In this case, thresholds are defined to separate normal resource consumption from anomalous resource consumption.

2.2 Misuse Detection

Misuse detection is based on the knowledge of system vulnerabilities and known attack patterns. Misuse detection is concerned with finding intruders who are attempting to break into a system by exploiting some known vulnerability. Ideally, a system security administrator should be aware of all the known vulnerabilities and eliminate them. The term intrusion scenario is used as a description of a known kind of intrusion; it is a sequence of events that would result in an intrusion without some outside preventive intervention. An intrusion detection system continually compares recent activity to known intrusion scenarios to ensure that one or more attackers are not attempting to exploit known vulnerabilities. To perform this, each intrusion scenario must be described or modeled. The main difference between the misuse techniques is in how they describe or model the behavior that constitutes an intrusion. The original misuse detection systems used rules to describe events indicative of intrusive actions that a security administrator looked for within the system. Large numbers of rules can be difficult to interpret. If-then rules are not grouped by intrusion scenarios and therefore making modifications to the rule set can be difficult as the affected rules are spread out across the rule set. To overcome these difficulties, new rule organizational techniques include model-based rule organization and state transition diagrams. Misuse detection systems use the rules to look for events that possibly fit an intrusion scenario. The events may be monitored live by monitoring system calls or later using audit records.

3 Types of Intrusion Detection Systems

There are two types of intrusion detection systems that employ one or both of the intrusion detection methods outlined above. Host-based systems base their decisions on information obtained from a single host (usually audit trails), while network-based intrusion detection systems obtain data by monitoring the traffic in the network to which the hosts are connected.

3.1 Host-Based Intrusion Detection

A generic intrusion detection model proposed by Denning [9] is a rule-based pattern matching system in which the intrusion detection tasks are conducted by checking the similarity between the current audit record and the corresponding profiles. If the current audit record deviates from the normal patterns, it will be considered an anomaly. Several IDSs were developed using profile and rule-based approaches to identify intrusive activity [18].

3.2 Network-Based Intrusion Detection

With the proliferation of computer networks, more and more individual hosts are connected into local area networks and/or wide area networks. However, the hosts, as well as the networks, are exposed to intrusions due to the vulnerabilities of network devices and network protocols. The TCP/IP protocol can be also exploited by network intrusions such as IP spoofing, port scanning, and so on. Therefore, network-based intrusion detection has become important and is designed to protect a computer network as well as all of its hosts. The installation of a network-based intrusion detection system can also decrease the burden of the intrusion detection task on every individual host.

4 Data Mining Approaches Toward Intrusion Detection

In this paper we propose a data mining approach for intrusion detection. A review of intrusion detection systems that employ non-data mining techniques is therefore not presented. Data mining approaches are new methods in intrusion detection systems. Data mining is defined as the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data [36]. Data mining attempts to extract knowledge in the form of models from data, which may not be seen easily with the naked eye. There exist many different types of data mining algorithms including classification, regression, clustering, association rule abduction, deviation analysis, sequence analysis etc.

Various data mining techniques have been applied to intrusion detection because it has the advantage of discovering useful knowledge that describes a user's or program's behavior from large audit data sets. Data mining has been used for anomaly detection [23, 24]. Statistics [1, 8], Artificial Neural Network (ANN) [26, 27] and Hidden Markov Model) (HMM) [28], Rule Learning [29], Outlier Detection scheme [30], Support Vector Machines [2], Neuro-Fuzzy (NF) computing [40], Multivariate Adaptive Regression Splines [4] and Linear Genetic Programming [21] are the main data mining techniques widely used for anomaly and misuse detections.

Statistics is the most widely used technique, which defines normal behavior by collecting data relating to the behavior of legitimate users over a period of time [1]. NIDES (Next-generation Intrusion Detection Expert Systems) is the representative IDS based on statistics that measures the similarity between a subject's long-term behavior and short term behavior for intrusion detection [8]. The detection rate is high because it can use various types of audit data and detect intrusion based on the previous experimental data.

Hyperview is a representative IDS using neural networks [26]. It consists of 2 modules: a neural network and an expert system. R. Lippmann et al. have applied neural networks to a keyword-based detection system [27]. An Hidden Markov Model (HMM) is a useful tool to model the sequence of observed symbols of which the construction mechanism cannot be known [28]. While HMM produces better performance in modeling system call events compared to other methods, it requires a very long time for modeling normal behaviors. Using this model, raw data is first converted into ASCII network packet information, which in turn is converted into connection level information using Mining Audit Data for Automated Models for Intrusion Detection (MADAMID) [19]. RIPPER [29], a rule learning tool, is then applied to the data generated by MADAMID. RIPPER automatically mines the patterns of intrusion. Although it is a good tool for discovering known patterns, an anomaly detection technique is required for the detection of novel intrusions. Another data mining technique, the outlier detection scheme attempts to identify a data point that is very different from the rest of the data. A. Lazarevic et al.[30] have applied it to anomaly detection .

Support Vector Machines (SVM) have proven to be a good candidate for intrusion detection because of its speed and scalability [40]. An Adaptive neuro-fuzzy [2] IDS is proposed in [22]. An IDS based on Multivariate Adaptive Regression Splines (MARS) [4, 3] is proposed in [3]. In Linear Genetic Programming (LGP) (as opposed to tree-based Genetic Programming (GP)) [37] computer programs are evolved at the machine code level, using lower level representations for the individuals. This can tremendously hasten up the evolution process. LGP based IDS is presented in [20]. To overcome the drawbacks of single-measure detectors, a multiple measure intrusion detection method is proposed in [31]. In this approach hidden Markov model, statistical method and rule-base method are integrated with a rule-based approach. In [39], the authors have proposed an ensemble IDS that combines the strengths of Bayesian Networks and Classification and Regression Trees for intrusion detection.

5 The Data Mining Process of Building Intrusion Detection Models

Raw (binary) audit data is first processed into ASCII network packet information (or host event data), which is in turn summarized into connection records (or host session records) containing a number of within- connection features, e.g., service, duration, flag etc. (indicating the normal or error status according to the protocols). Data mining programs are then applied to the connection records to compute the frequent patterns i.e. association rules and frequent episodes, which are in turn analyzed to construct additional features for the connection records. Classification algorithms are then used to inductively learn the detection model.

5.1 Importance of Data Reduction for IDS

IDSs have become important and widely used tools for ensuring network security. Since the amount of audit data that an IDS needs to examine is very large even for a small network, classification by hand is impossible. Analysis is difficult even with computer assistance because extraneous features can make it harder to detect suspicious behavior patterns. Complex relationships exist between the features, which are practically impossible for humans to discover. An IDS must therefore reduce the amount of data to be processed. This is extremely important if real-time detection is desired. Reduction can occur in one of several ways. Data that is not considered useful can be filtered, leaving only the potentially interesting data. Data can be grouped or clustered to reveal hidden patterns. By storing the characteristics of the clusters instead of the individual data, overhead can be significantly reduced. Finally, some data sources can be eliminated using feature selection. The amount of amount of audit data that an IDS needs to examine is very large even for a small network and between analyzed features may exists false correlations, complex relationships and features may be redundant. Therefore, IDSs uses several techniques which solves these problems.

Data filtering techniques reduce the amount of data directly processed by the IDS. Clustering can be performed to find hidden patterns in data and significant features for use in detection. Clustering can also be used as a reduction technique by storing the characteristics of the clusters instead of the individual data. In previous work a number of experiments have been performed to measure the performance of different machine learning paradigms. Classifications were performed on the binary (normal/attack) as well as five-class classifications (normal, and four classes of attacks). It has been demonstrated that a large number of the (41) input features are unimportant and may be eliminated, without significantly lowering the performance of the IDS [43]. In terms of the five-class classification, the authors of [43] found that by using only 19 of the most important features, instead of the entire 41 feature set, the change in accuracy of intrusion detection was statistically insignificant. In [43] the authors applied the technique of deleting one feature at time. Each reduced feature set was then tested on Support Vector Machines and Neural Networks to rank the importance of input features. The reduced feature set that yielded the best detection rate in the experiments was considered to be the set of important features.

Unlike the work reported in [43], which employed a trial-and-error based approach, we investigate feature reduction using data mining techniques. Our research correspondingly focuses on approaches that will improve the performance of IDSs by providing real-time intrusion detection. This is achieved by using matrix factorization or factor analysis for reducing the data space and then classifying intrusions based on the reduced feature space. Performance is compared with Bayesian networks and Classification and Regression Trees (CART) [39], [38]. Bayesian Networks not only classify the data, but also selects features based on the Markov Blanket of the target variables. CART classifies data by constructing a decision tree. Furthermore, the CART algorithm automatically produces a predictor ranking (variable importance) based on the contribution predictors make to the construction of the decision tree, thus helping to identify which features are important for intrusion detection.

6 Matrix Factorization for Feature Selection and Classification

Matrix factorization or factor analysis is an important method in the analysis of high dimensional real world data. There are several well known methods and algorithms for factorization of real data but many application areas including information retrieval, pattern recognition and data mining require processing of binary rather than real data (see [13, 25, 42]).

Non-negative matrix factorization is really a class of decompositions whose members are not necessarily closely related to each other [10, 41]. They share the property that are designed for datasets in which attribute values are never negative - and its does not make sense for the decomposition matrices to contain negative values either. A side-effect of this non-negativity property is that the mixing of components that we have seen is one way to understand decompositions can only be additive. A set of data S can be expressed as a $m \times n$ matrix V, where m is the number of attributes and n is the number of records in S. Each column V_j of V is an encoding of a record in S and each entry v_{ij} of vector V_j is the value of i-th term with regard to the semantics of V_j, where i ranges across attributes.

The NMF problem is defined as finding an approximation of V in terms of some metric (e.g., the norm) by factoring V into the product WH of two reduced-dimensional matrices W and H. Each column of W is a basis vector. It contains an encoding of a semantic space or concept from V and each column of H contains an encoding of the linear combination of the basis vectors that approximates the corresponding column of V. Dimensions of W and H are $m \times k$ and $k \times n$, where k is the reduced rank. Usually k is chosen to be much smaller than n. Finding the appropriate value of k depends on the application and is also influenced by the nature of the collection itself. Common approaches to NMF obtain an approximation of V by computing a (W, H) pair to minimize the Frobenius norm of the difference $V - WH$. The matrices W and H are not unique. Usually H is initialized to zero and W to a randomly generated matrix where each $W_{ij} > 0$ and these initial values are improved with iterations of the algorithm.

1. Initialize W and H with nonnegative values, and scale the columns of W to unit norm.

2. Iterate until convergence or after l iterations:

   - $W_{ic} = W_{ic}\frac{(VH^T)_{ic}}{(WHH^T)_{ic}+\epsilon}$, for c and i [$\epsilon = 10^{-9}$]
   - Rescale the columns of W to unit norm
   - Solve the constrained least squares problem where $min_{H_j}\{||V_j - WH_j||_2^2 + \lambda||H_j||2^2$ the subscript j denotes the j-th column, for $j = 1, \ldots, m$. Any negative values in H_j are set to zero. The parameter k is a regularization value that is used to balance the reduction of the metric $||V_j - WH_j||_2^2$ with the enforcement of smoothness and sparsity in H.

For any given matrix V, matrix W has k columns or basis vectors that represent k clusters, matrix H has n columns that represent n documents. A column vector in H has k components, each of which denotes the contribution of the corresponding basis vector to that column or document. The clustering of documents is then performed based on the index of the highest value of k for each document. For document $i (i = 1 \ldots, n)$, if the maximum value is the j-th entry $(j = 1, \ldots, k)$, document i is assigned to cluster j.

There is a number of optimization tasks related to NMF [45]. *Unconstrained NMF*, which corresponds to the original NMF introduced above, is a non-convex problem with known algorithms that can compute its global optimum. Unfortunatelly, they are not able to deal with real-world sized data and so algorithms finding local optimum have to be employed. Although the NMF codes are rather sparse, in *Sparsity constrained NMF* is the sparsity controlled directly. The sparsity constraints control to what extent basis functions are sparse, and how much each basis function contributes to the reconstruction of only a subset of the original data matrix V. In *Supervised NMF*, the information about class membership is incorporated to the learning process. The NMF then describes well the processed data and in addition allows good discrimination in a subsequent classification stage.

Attack Class	41 variables			12 variables		
	Train (sec)	Test (sec)	Accuracy (%)	Train (sec)	Test (sec)	Accuracy (%)
Normal	23.36	27.39	77.68	5.17	9.64	77.42
Probe	23.51	32.48	89.87	5.00	6.31	95.09
DOS	23.37	34.95	78.13	4.82	7.61	81.04
U2R	23.23	29.90	97.45	5.29	8.84	92.50
R2L	22.78	30.00	98.55	5.53	9.29	98.59

Table 1. Performance using matrix factorization approach

Attack Class	41 variables			17 variables		
	Train (sec)	Test (sec)	Accuracy (%)	Train (sec)	Test (sec)	Accuracy (%)
Normal	42.14	19.02	99.57	23.29	11.16	99.64
Probe	49.15	21.04	99.43	25.07	13.04	98.57
DOS	54.52	23.02	99.69	28.49	14.14	98.16
U2R	30.02	15.23	64.00	14.13	7.49	60.00
R2L	47.28	12.11	99.11	21.13	13.57	98.93

Table 2. Performance of Bayesian Belief Network

NMF can be used to organize data collections into partitioned structures or clusters directly derived from the nonnegative factors. Potential applications include the monitoring, tracking and clustering of semantic features (topics) and can be use for intrusion detection.

Attack Class	41 variables			12 variables		
	Train (sec)	Test (sec)	Accuracy (%)	Train (sec)	Test (sec)	Accuracy (%)
Normal	1.15	0.18	99.64	0.80	0.02	100.00
Probe	1.25	0.03	97.85	0.85	0.05	97.71
DOS	2.32	0.05	99.47	0.97	0.07	85.34
U2R	1.10	0.02	48.00	0.45	0.03	64.00
R2L	1.56	0.03	90.58	0.79	0.02	95.56

Table 3. Performance of classification and regression trees

7 Experiment Setup and Results

The data for our experiments was prepared by the 1998 DARPA intrusion detection evaluation program by MIT Lincoln Labs [17]. The data set contains 24 attack types that could be classified into four main categories namely Denial of Service (DOS), Remote to User (R2L), User to Root (U2R) and Probing. The original data contains 744 MB data with 4,940,000 records. The data set has 41 attributes for each connection record plus one class label. Some features are derived features, which are useful in distinguishing normal connection from attacks. These features are either nominal or numeric. Some features examine only the connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc. These are called same host features. Some features examine only the connections in the past two seconds that have the same service as the current connection and are called same service features. Some other connection records were also sorted by destination host, and features were constructed using a window of 100 connections to the same host instead of a time window. These are called host-based traffic features. R2L and U2R attacks do not have any sequential patterns like DOS and Probe because the former attacks have the attacks embedded in the data packets whereas the later attacks have many connections in a short amount of time. So some features that look for suspicious behavior in the data packets like number of failed logins are constructed and these are called content features. Our experiments have three phases namely data reduction, a training phase and a testing phase.

In the data reduction phase, important variables for real-time intrusion detection are selected by feature selection. In the training phase, matrix factorization method, Bayesian neural network and classification and regression trees construct a model using the training data to give maximum generalization accuracy on the unseen data. The test data is then passed through the saved trained model to detect intrusions in the testing phase. The data set for our experiments contains randomly generated 11982 records having 41 features [14].

This data set has five different classes namely Normal, DOS, R2L, U2R and Probes. The training and test comprises of 5092 and 6890 records respectively. All the IDS models were trained and tested with the same set of data. As the data set has five different classes we performed a 5-class binary classification. The Normal data belongs to class 1, Probe belongs to class 2, DOS belongs to class 3, U2R belongs to class 4 and R2L belongs to class 5.

Matrix factorization method (NMF) is used in two phases - training and testing. In training phase, training collection, number of components(variables) and number of iterations are input parameters. Matrices W and H are created (their size is based on number of variables) and filled with random numbers. Then, both matrices are updated using rules defined in previous section in iterative process. When all iterations are completed, matrix W contains base vectors in its columns and matrix H contains coefficients in its rows. Clusters centers are calculated as average from coefficients according to known classification.

The testing phase has testing collection, number of iterations, matrix W from training phase and calculated clusters as input parameters. Matrix H is created and filled with random non-negative numbers. The iterative process is based only on updating of matrix H. The calculated coefficients are then classified in reference to clusters computed in training phase. We use 100 iterations and number of variables as is defined in table in out experiments.

Empirical results using the matrix factorization method is illustrated in Table 1. Tables 2 and 3 illustrate the empirical results obtained using Bayesian and CART approaches [39], [38]. As evident, the proposed factorization approach performed extremely well for the U2R attack category in terms of reduced features and classification accuracy.

8 Conclusions

In this research, we have investigated new techniques for intrusion detection and performed data reduction and evaluated their performance on the DARPA benchmark intrusion data. We used the feature selection method using matrix factorization method and compared the performance with Markov blanket model and decision tree analysis. The developed detection model based on clustering using NMF dimension reduction method seem to work very well especially for the UR2 attack category. As future research, we plan to develop ensemble combinations using the various approaches for intrusion detection.

Acknowledgements Authors would like to thank the anonymous reviewers for the constructive comments which helped to improve the clarity and presentation of the paper.

References

[1] D. Anderson, T.F. Lunt, H. Javits, A. Tamaru and A. Valdes, ”Detecting unusual program behavior using the statistical components of NIDES,” NIDES Technical Report, SRI International, May 1995.

[2] Ajith Abraham, ”Neuro-Fuzzy Systems: State-of-the-Art Modeling Techniques, Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence”, Lecture Notes in Computer Science Volume 2084, Springer-Verlag Germany, Jose Mira and Alberto Prieto (Eds.), Granada, Spain, pp. 269-276, 2001.

[3] Ajith Abraham and Dan Steinberg, ”MARS: Still an Alien Planet in Soft Computing?” Lecture Notes in Computer Sci-

ence 2074, Springer-Verlag Germany, Vassil N. Alexandrov et al. (Eds.), San Francisco, pp. 235-244, 2001.

[4] Banzhaf. W., Nordin. P., Keller. E. R., Francone F. D., Genetic Programming: An Introduction on The Automatic Evolution of Computer Programs and its Applications, Morgan Kaufmann Publishers, Inc., 1998.

[5] Matt Bishop, Computer Security - Art and Science, Addison Wesley, 2003

[6] L. Brieman, J.Friedman, R. Olshen and C. Stone, Classification of Regression Trees. Wadsworth Inc., 1984.

[7] J. Cheng, R. Greiner, J. Kelly, D.A. Bell and W. Liu, "Learning Bayesian Networks from Data: an Information-Theory Based Approach", The Artificial Intelligence Journal, Volume 137, pp. 43-90, 2002.

[8] H. Debar, M. Becker and D. Siboni, "A neural network component for an intrusion detection system," Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy, pp. 240-250, Oakland, CA, May 1992.

[9] D. E. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, Volume SE-13, No. 2, pp. 222-232, 1987.

[10] L. Elden. Matrix Methods in Data Mining and Pattern Recognition. SIAM 2007.

[11] S. Forrest, A.S. Perelson, L. Allen, R. and Cherukuri, "Self-Nonself Discrimination in a Computer", Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Computer Society Press (1994)

[12] Gene H. Kim, Eugene H. Spafford, "Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection", http://citeseer.ist.psu.edu/kim95experiences.html (1995)

[13] D. Húsek, P. Moravec, V. Snášel, A.A. Frolov, H. Řezanková, P. Polyakov: Comparison of Neural Network Boolean Factor Analysis Method with Some Other Dimension Reduction Methods on Bars Problem. Springer, LNCS 4815, PReMI 2007: pp. 235-243

[14] KDD cup 99 Intrusion detection data set
http://kdd.ics.uci.edu/databases/kddcup99/
kddcup.data_10_percent.gz

[15] W. Lee., S. Stolfo. and K. Mok, "A Data Mining Framework for Building Intrusion Detection Models", Proceedings of the IEEE Symposium on Security and Privacy, 1999.

[16] J. Luo and S. M. Bridges, "Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection", International Journal of Intelligent Systems, John Wiley & Sons, Vol. 15, No. 8, pp. 687-704, 2000.

[17] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/

[18] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, H. S. Javitz and Al Valdes, "IDES: The Enhanced Prototype - A Real-Time Intrusion-Detection Expert System", Number SRI-CSL-88-12. Computer Science Laboratory, SRI International, Menlo Park, CA, 1988.

[19] W. Lee and S. Stolfo and K. Mok. "A Data Mining Framework for Building Intrusion Detection Models". Proceedings of the IEEE Symposium on Security and Privacy, 1999.

[20] Srinivas Mukkamala, Andrew H. Sung and Ajith Abraham, "Modeling Intrusion Detection Systems Using Linear Genetic Programming Approach", The 17th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Innovations in Applied Artificial Intelligence, Robert Orchard, Chunsheng Yang, Moonis Ali (Eds.), LNCS 3029, Springer Verlag, Germany, pp. 633-642, 2004.

[21] Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham and Vitorino Ramos, "Intrusion Detection Systems Using Adaptive Regression Splines", 6th International Conference on Enterprise Information Systems, ICEIS'04, Portugal, I. Seruca, J. Filipe, S. Hammoudi and J. Cordeiro (Eds.), Vol. 3, pp. 26-33, 2004.

[22] Khusbu Shah, Neha Dave, Sampada Chavan, Sanghamitra Mukherjee, Ajith Abraham and Sugata Sanyal, "Adaptive Neuro-Fuzzy Intrusion Detection System", IEEE International Conference on Information Technology: Coding and Computing (ITCC'04), USA, IEEE Computer Society, Volume 1, pp. 70-74, 2004.

[23] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. of the 7th USENIX Security Symposium, San Antonio, Texas, January 26-29, 1998.

[24] T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, and P.G. Neuman, "A real-time intrusion- detection expert system (IDES)," Technical Report Project 6784, CSL, SRI International, Computer Science Laboratory, SRI International, February 1992.

[25] P. Moravec and V. Snášel. Dimension Reduction Methods for Image Retrieval. In Proceedings of the Conference on Intelligent Systems Design and Applications (ISDA2006), 6 pages, Jinan, Shandong, China, October 2006. IEEE Press.

[26] R. Lippmann and S. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," Computer Networks, vol. 34, no. 4, pp. 594-603, 2000.

[27] S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows with hidden Markov model," Computers & Security, vol. 22, no. 1, pp. 45-55, 2003.

[28] W.W. Cohen, "Fast effective rule induction," Proceedings of the 12th International Conference on Machine Learning, pp. 115-123, July 1995.

[29] A. Lazarevic, L. Ertoz,, V. Kumar, A. Ozgur and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," Proceedings of Third SIAM Conference on Data Mining, May 2003.

[30] Sang-Jun Han and Sung-Bae Cho, "Detecting intrusion with rule-based integration of multiple models," Computers & Security, Volume 22, Issue 7, October 2003, pp. 613-623

[31] I.T. Jolliffe. Principal Component Analysis. Springer-Verlag, 1986.

[32] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley & Sons, 2001

[33] R. E. Neapolitan, Probabilistic reasoning in expert systems: theory and algorithms, John Wiley & Sons, 1990.

[34] S. Dzeroski, B. Zenko, ”Is Combining Classifiers Better than Selecting the Best One”. ICML 2002, pp. 123-130, 2002.

[35] C. Ji and S. Ma, ”Combinations of weak classifiers”. IEEE Transaction on Neural Networks, 8(1), pp. 32-42, 1997.

[36] David J. Hand, Heikki Mannila, Padhraic Smyth, Principles of Data Mining (Adaptive Computation and Machine Learning), Bradford Books, 2001

[37] Karlton Sequeira and Mohammed Zaki, ”ADMIT: Anomaly based Data Mining for Intrusions”, SIGKDD 2002, Edmonton, Alberta, Canada.

[38] Srilatha Chebrolu, Ajith Abraham and Johnson Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computers and Security, Elsevier Science, Volume 24/4, pp. 295-307, 2005.

[39] Srilatha Chebrolu, Ajith Abraham and Johnson Thomas, Hybrid Feature Selection for Modeling Intrusion Detection Systems, 11th International Conference on Neural Information Processing, ICONIP’04, Pal N.R. et al. (Eds.) Springer Verlag, Germany, Lecture Notes in Computer Science, Vol. 3316, ISBN 3-540-23931-6, pp. 1020-1025, 2004.

[40] Srinivas Mukkamala, Andrew H. Sung and Ajith Abraham, ”Intrusion Detection Using Ensemble of Soft Computing Paradigms”, Third International Conference on Intelligent Systems Design and Applications, Intelligent Systems Design and Applications, Advances in Soft Computing, Springer Verlag, Germany, pp. 239-248, 2003.

[41] D. Skillicorn. Understanding Complex Datasets: Data Mining with Matrix Decomposition. Chapman & Hall, 2007.

[42] V. Snášel, D. Húsek, Alexander A. Frolov, H. Řezanková, P. Moravec, P. Polyakov: Bars Problem Solving - New Neural Network Method and Comparison. Lecture Notes in Computer Science 4827, MICAI 2007: pp. 671-682

[43] A. H. Sung and S. Mukkamala, ”Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks”, Proceedings of International Symposium on Applications and the Internet (SAINT 2003), pp. 209-217, 2003.

[44] I. Tsamardinos, C. F. Aliferis and A. Statnikov , ”Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations”, 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, USA, ACM Press, pp. 673-678, 2003.

[45] M. Heiler, Ch. Schnrr, ”Learning Sparse Representations by Non-Negative Matrix Factorization and Sequential Cone Programming”, The Journal of Machine Learning Research, vol. 7, pp. 1385-1407, 2006.