Mutating Network Scans for Classifier Ensemble Assessment

J. Sedano[a], S. González[a], Á. Herrero[b], B. Baruque[b], and E. Corchado[c,d]

[a]*Instituto Tecnológico de Castilla y León, C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001, Burgos, Spain. E-mail: javier.sedano@itcl.es, silvia.gonzalez@itcl.es*

[b]*Department of Civil Engineering, University of Burgos, University of Burgos, C/ Francisco de Vitoria s/n, 09006 Burgos, Spain. E-mail: ahcosio@ubu.es, bbaruque@ubu.es*

[c]*Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced, s/n, 37008 Salamanca, Spain. E-mail: escorchado@usal.es*

[d]*Visiting Professor at Department of Computer Science. VŠB-Technical University of Ostrava, Ostrava, Czech Republic*

Abstract. As it is well known, some Intrusion Detection Systems (IDSs) suffers from high rates of false positives and negatives. A novel mutation technique is proposed in this study to test and evaluate the performance of a full range of classifier ensembles for Network Intrusion Detection when tackling the task of recognizing new attacks. The technique applies mutant operators that randomly modify the features of the captured network packets to generate situations that could not otherwise be provided to IDSs while learning. A comprehensive comparison of classifiers and classifier ensembles is performed to assess their generalization capability when confronting brand new network attacks. Finally, an example application of the proposed testing model is specially applied to the identification of network scans and related mutations.

KEYWORDS: Network Intrusion Detection, Classifier Ensembles, Machine Learning, Intrusion Detection Systems, IDS Performance, Zero-day Attacks.

1. Introduction & Motivation

The ever-changing nature of attack strategies and technologies is certainly one of the most harmful issues of attacks and intrusions. It greatly increases the difficulty of protecting computer networks and systems. As is well known, Intrusion Detection Systems (IDSs) figure prominently among the various kinds of tools proposed to date to confront such problems [3, 20]

In the context of computer networks, an IDS can roughly be defined as a tool that is designed to analyze the computer data network, in order to detect suspicious patterns that may be related to a network or system attack, assuming that some other security (prevention) tools may have failed. The main purpose of an IDS is to alert the security personnel to any suspicious and possibly intrusive event taking place in the system that is under analysis. Intrusion Detection (ID) is therefore a field that focuses on the identification of attempted or ongoing attacks on a computer system (Host IDS - HIDS) or network (Network IDS - NIDS).

A standard characterization of IDSs, based on their detection method, or model of intrusions, defines the following paradigms:

- **Anomaly-based ID** (also known as behaviour-based ID): the IDS detects intrusions by looking for activity that differs from the previously defined “normal” behaviour of users and/or systems. In keeping with this idea, the observed activity is compared against "predefined" profiles of expected normal usage. It is assumed that all intrusive activities are necessarily anomalous. In real-life environments, instead of their being identical, the set of intrusive activities only intersects the set of anomalous activities in certain cases. As a consequence, [58] some anomalous activities that are not intrusive are flagged as intrusive (i.e. false positives) and intrusive activities that are not anomalous are not flagged up (i.e. false negatives). Anomaly-based IDSs can support detection of novel (zero-day) attack strategies, but may suffer from a relatively high rate of false positives. [41]

- **Misuse-based ID** (also known as knowledge-based ID): intrusions are detected by checking activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities. Misuse-based IDSs are therefore commonly known as signature-based IDSs. They detect intrusions by exploiting the available knowledge on specific attacks and vulnerabilities. As opposed to anomaly detection, misuse detection assumes that each intrusive activity can be represented by a unique pattern or signature [48]. The main problem with this approach is that intrusions with signatures that have not been archived by the system cannot be detected. As a consequence, a misuse-based IDS will never detect a 0-day attack [48]. The completeness of such IDSs requires regular updating of their knowledge of attacks.
- **Specification-based ID**: program behavioural specifications reflect system policies that are used as a basis to detect attacks [6].

Some authors claim that one of the main challenges for researchers seeking to implement and to validate a new ID method is to assess and compare its performance with other available approaches [28]. Benchmarking an IDS is a common way of establishing its effectiveness, although this task is neither easy nor clear-cut [46, 8].

Some interesting works have proposed IDS testing in real (or as realistic as possible) environments [37] and in experimental settings [29, 45]. These two approaches have been proven to have some advantages and disadvantages, as pointed in [9], in which a combination is proposed.

The most widely accepted approach to IDS benchmarking is based on a publicly available data set. In this approach, the performance of a new IDS can be compared against those of previous IDSs by analysing the data set. Up to now, DARPA [33] and the associated KDD Cup 1999 [2] data sets are available for such purposes. DARPA initiated its ID Evaluation program in 1998 [1]. These evaluations generate background traffic interlaced with malicious activity so that IDSs and algorithms can be tested and compared. Certain problems concerning the inner structure and features of this data set have been identified [36, 50]. It also has some other up-to-date problems such as the outdated attacks and the data rate, which is not comparable with that in real current networks [13]. Despite these limitations, this is still the *de-facto* standard corpus for IDS testing so far, as there are important constraints (privacy and confidentiality mainly) that prevent a new network traffic data set from being made publicly available.

Apart from testing data sets, some methodologies [9, 45], frameworks [29, 26], and tools [32, 44] for assessing IDSs have also been proposed. Additionally, comparative studies on commercial IDSs have been presented, mainly in network journals.

Despite the fact that increasing effort has been devoted over recent years to IDSs in general, and to IDSs assessment in particular, there is no comprehensive and scientifically rigorous methodology to test the effectiveness of these systems [42]. Thus, it remains an open issue and a significant challenge [28].

In view of the above-mentioned issues, a fair comparison between different methods for ID is difficult to achieve. In consequence, a mutation-based technique [24, 22] is applied in this study to check the generalization capability of different classifiers and classifier ensembles [14, 53]. This novel testing technique, which specializes in IDSs that rely on numerical packet features, is based on measuring the evaluated IDS results when they encounter unknown anomalous situations, as the identification of zero-day (previously unseen) attacks is possibly the most challenging and important ID problem. [46, 24, 22] In this case, the method is used to assess a comprehensive set of classifiers and classifier ensembles in the detection of mutated network scans. The ability to detect such scans can help identify wider and potentially more dangerous threats to a network. This testing model has the main advantage of providing the classifiers with brand new attacks - network scans in this case -.

Three types of scans (or sweeps) were defined: network scans, port scans, and their hybrid block scans [57]. Unlike other attacks, scans must use a real source IP address, because the results of the scan (open ports or responding IP addresses) must be returned to the attacker [47].

A port scan (or sweep) may be defined as series of messages sent to different port numbers to gain information on its activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services that this host is providing. So, a scan is an attempt to count the services running on a machine (or a set of machines) by probing each port for a response. A network scan is one of the most common techniques used to identify services that might then be accessed without permission [7].

A port scan may be defined as a series of messages sent to different port numbers to gain information on their activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services the host is providing. A port scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity. This work focuses on the identification of network scans, in which the same port is the target for a number of computers in an IP address range.

The remaining sections of this paper are structured as follows: section 2 introduces the mutation testing technique. The applied classifiers and classifier ensembles are then described in section 3, the experimental results in section 4. Finally the conclusions of this study along with future work are discussed in section 5.

2. A Mutation Testing Technique for IDSs

Misuse IDSs rely on models of known attacks. The effectiveness of these IDSs depends on the "goodness" of their models. This is to say, if a model of an attack does not cover all the possible modifications, the performance of the IDS will be greatly impaired. As previously stated, this is one of the main drawbacks of the misuse-based ID approach.

This study focuses on verifying the performance of many machine-learning models (classifiers and classifier ensembles) when applied to ID and when confronted with unknown (not previously seen) anomalous situations. To generate such brand new situations, a mutation testing model is used.

The mutation testing model was previously applied to a visualization-based IDS [24, 22] and is based on mutating attack traffic. In general, a mutation can be defined as a random change. In keeping with this idea, the testing model modifies different features of the numerical information extracted from the packet headers.

The modifications created by this model may involve changes in such aspects as: attack length (amount of time that each attack lasts), packet density (number of packets per time unit), attack density (number of attacks per time unit) and time intervals between attacks. The mutations can also concern both source and destination ports, varying between the different three ranges of TCP/UDP port numbers: well known (from 0 to 1,023), registered (from 1,024 to 49,151) and dynamic and/or private (from 49,152 to 65,535).

Time is a further issue of great importance when considering intrusions since the chances of detecting an attack increase in relation to their duration. There are therefore two main strategies:

- To make a significant reduction in the time used to perform a scan.
- To spread the packets out over time, which is to say, to reduce the number of packets sent per time unit that are likely to slip past unnoticed.

In this study, the mutations are applied to data that relate to network scans. Any of the possible mutations may be meaningless, such as a sweep of less than 5 hosts in the case of a network scan.

Changes can be made to attack packets taking the following issues into account:

- Number of scans in the attack (that is, number of addressed port numbers).
- Destination port numbers at which scans are aimed.
- Time intervals when scans are performed.
- Number of packets (density) forming the scans (number of scanned hosts).

3. Classifiers and Ensembles for Intrusion Detection

As previously explained, one of the most interesting features of an ideal IDSs would be automatic detection of traffic that constitutes an attack against the network as opposed to normal traffic. Automated learning techniques are algorithms specifically designed to take decisions with regard to new data that are presented on the basis of information extracted from previous data.

The current study is centred on the use of supervised algorithms [15] to detect new anomalies. Although some previous researches have proposed classifier ensembles for ID [17, 39, 38], in this case, ensembles of supervised learning algorithms are presented to assess their generalization capability. The intention is to ascertain whether, once a certain type of attack is discovered, such information can be incorporated in the IDS, in order to detect new attacks in the future –with a certain similarity to those detected previously–.

Usually, most automated learning algorithms suffer from common problems; such as over-fitting of the data used for training –and therefore, poor generalization capabilities-, getting stuck on local minima in their learning function, and high computational complexity when dealing with complex data [43]. One of the most widespread and useful techniques to avoid such problems is the ensemble learning scheme [55]. The main idea behind this kind of meta-algorithm is to train several, less complex classifiers and combine their results in a single and usually more complex classifier, which will generate even better results [49].

Several different algorithms have been considered both for the base classifiers and for the ensemble construction, in order to obtain a significant wide array of possible algorithms and to compare their performance results on mutated data sets. Among the base classifiers, all of which belong to the family of instance-based statistical classification algorithms, an array of techniques has been studied with different approaches to that problem; the instance based k-Nearest Neighbours (IBk), classification trees (such as the Simple Classification and Regression Decision Tree (CART) and the REP-Tree), and Artificial Neural Networks such as the Radial Basis Function Network [15, 14].

Among the ensemble meta-algorithms that use the above-mentioned simple algorithms, this study tests basic algorithms such as the MultiClass Classifier, used to adapt binary classifiers to multi-class problems; Bagging, Adaptative Boosting (AdaBoost), and Random Forest [43]. It compares their results with more modern boosting algorithms such as the LogitBoost [27] or the StackingC [60].

All of these techniques have previously been used to good effect, in order to solve practical problems. The Multi-Class classifier has been used in the field of image recognition [51] or bio-informatics 31]; Bagging and AdaBoost are among the best known algorithms that have been used in a wide array of fields such as food quality assessment [21], financial risk assessment [59], robotics [10], image [16] or signal processing [11], weather conditions forecasting [12] or operational costs optimization [52], to name but a few.

As explained in previous studies [54], ensemble meta-algorithms work well, amongst other reasons, because their use makes it relatively easy to increase the number of hypotheses used to classify new data. This increase is achieved by partitioning the data used to train the different components of the ensemble, allowing each of them to concentrate on a specific data space within the overall space of the problem. In the case of anomaly/attack detection, this aspect is very interesting, as an anomaly/attack can be regarded as the kind of data that the classifier is unable to recognize. This could be because the data has never been presented to the classifier before or because the size of the anomaly data is very small compared to normal data, which the classifier fails to notice. By partitioning the dataset to construct different –and hopefully sufficiently diverse– classifiers, the probability of a classifier failing to notice data decreases; also decreasing the probability of the ensemble classifiers not noticing an attack.

As results prove, ensemble learning adds an important value to the analysis, as almost all variants consistently improve on the results obtained by a single classifier.

4. Experiments & Results

This section describes the data sets used for evaluating the proposed testing method and how they were generated, before detailing their results.

4.1. Data sets

Real-life data sets have previously been applied to perform ID [24, 22]. This data set was generated "made-to-measure" in a small-size (28 hosts) network where "normal" traffic was known in advance. In addition to the SNMP packets, the data set contains traffic related to some other protocols, considered as "normal" traffic. As this network was isolated and protected from external attacks, "normal" traffic was known in advance and has been empirically tested. The captured packets relate to 63 different protocols. Further details about the network in which the traffic was captured are unavailable due to the security policy. This data set generation methodology has previously been described in detail [22].

A set of features extracted from the packet headers characterized packets travelling along the monitored network. Once codified, the following five features contribute to building up the input vectors of the machine-learning models, $x \in \Re^2$:

- Timestamp: the time when the packet was sent.
- Source port: the port number of the device that sent the packet.
- Destination port: the port number of the target host, i.e. the host to which the packet is sent.
- Protocol ID: an integer number that identifies the protocol over TCP of the packet.
- Size: the packet size (in Bytes).

It has been proven that this low-dimensional data sets allow for the detection of some anomalous situations, mainly those related to SNMP [34, 22].

Different data sets were generated, containing different examples of SNMP anomalous situations. SNMP (Simple Network Management Protocol) [19, 25] was chosen because it was ranked as one of the top five most vulnerable services by CISCO [4]. Especially the first two versions [19, 18] of this protocol that still are the most widely used at present. SNMP attacks were also listed by the SANS Institute as one of the top 10 most critical internet security threats [5, 40].

The three main anomalous situations related to the SNMP are distributed throughout the different segments in this study, namely: network scans, SNMP community searches and MIB (Management Information Base) information transfers.

A hacker generates an SNMP community search by sending SNMP queries to the same port number of different hosts, and by using different strategies (brute force, dictionary, etc.) to guess the SNMP community string. Once the community string has been obtained, all the information stored in the MIB is available to the intruder. The unencrypted community string can be seen as the SNMP password for versions 1 and 2. In fact, it is the only SNMP authentication mechanism.

As reported in [5], "*The default community string used by the vast majority of SNMP devices is "public", with a few clever network equipment vendors changing the string to "private" for more sensitive information*". This facilitates SNMP intrusions, as intruders can guess the community string with little effort.

MIB information transfer involves a transfer of some (or all the) information contained in the SNMP MIB, generally through the *Get* command or similar primitives such as *GetBulk* [35, 56]. This kind of transfer is potentially quite a dangerous situation because anybody possessing some free tools, basic SNMP knowledge and the community string (in SNMP versions 1 and 2), will be able to access all sorts of interesting and sometimes useful information. As specified by the Internet Activities Board, the SNMP is used to access MIB objects. Thus, protecting a network from malicious MIB information transfer is crucial. However, the "normal" behaviour of a network may include queries to the MIB. This is a situation in which visualisation-based IDSs are quite useful; these situations may be visualised as anomalous by an IDS, but it is the responsibility of the network administrator to decide whether or not they constitute an intrusion.

By performing the SNMP community searches between the port/network scans and the MIB transfers, all anomalous situations constitute an SNMP attack from the outset. After attempting to establish whether and where (hosts and port numbers) SNMP is running, the community string is guessed, in order to access the information contained in the MIB. When the community string has been found, the MIB information is read.

The experimental setting of the present study comprises two different data sets:

– Data set 1: contains examples of the three anomalous situations (scans, community searches and MIB). All packets have been labelled according to the following six classes:
  1. Normal traffic.
  2. Scans to port number 161 (SNMP default port number).
  3. Scans to port number 162 (SNMP default port number).
  4. Scans to port number 3,750.
  5. MIB information transfer [23].
  6. Community search.
– Data set 2: this data set was generated by applying the mutation testing technique to Data set 1. As a result, it only contains examples of one of the anomalous situations, namely network scans. The network scans aimed at 161, 162 and 3,750 were mutated, while the other two anomalous situations were not present in the data set. As a consequence of the mutations, the scan to port number 161 was removed, the scan to port number 162 was redirected to port number 1,434 and the remaining one is now aimed at port number 65,788. Moreover, the mutations also modified the number of packets in the network scans. Therefore, the scans of this data set are labelled as classes 3 (scan to port number 1,434) and 4 (scan to port number 65,788).

4.2. Experiments

A combination of classifier ensembles was used to identify the new anomalous situations. In the experiment, a 10 K-fold cross-validation schema was selected. The final classification rate is obtained using the two previously described data sets. Two different data sets were therefore used for training and testing. All the classifiers were trained on data set 1; while validation was conducted only on the data set 2 (the one which contains the mutated network scans). The number of samples used for training and validation was 5,866 and 64, respectively. WEKA software [30] was used to train and classify the data sets, with ensembles and classifiers.

Each ensemble uses 10 base classifiers of the same type. The experimentation comprised a total of 1,534 tests, from the combination of 25 ensembles: including “Bagging”, “Boosting”, “Adaboost”, “Random Subspaces”, “Decorate”, “Rotation Forest”, etc., and 59 classifiers, including “NaiveBayes”, “Ibk”, “LinearRegression”, “JRip”, “RBFNetwork”, “SMO”, etc.; adding the baseline classifiers tests (that is, without ensembles). The best results are presented in the following section.

4.3. Results

The classifier ensembles try to differentiate between the six possible situations to identify these anomalous situations in the network, thereby labelling the training data sets in different ways. Thus, two different labels were assigned to differentiate attacks from normal traffic; three different classes were assigned to differentiate among MIB and community search, scans to all ports and normal traffic; and finally, four different classes were defined to obtain differences between normal traffic, community search, scans and MIB transfer. Table 1 presents the number of classes and the labels in each class, the situations in the network and the grouping of classes.

Table 1. Grouping of classes in the training and the validation data set.

Tables 2, 3, 4 and 5 show the training and validation/classification performance of the applied models. The values are obtained for each combination ensemble-classifier.The tables present the worst indices of training and classification in the chosen sets of classifiers.

The training percentage is calculated from k-folder strategy using the data set 1, while the classification percentage corresponds to the classification of anomalous situations of scans -data set 2-, with a classifier ensemble that has previously been trained with all values of the training data set.

Table 2. Clasification results for normal traffic and other situations in the network (two classes).

Table 3. Clasification results for normal traffic, scans and other situations in the network (three classes).

Table 4. Clasification results for normal traffic, scans, MIB and community search (four classes).

Table 5. Clasification results for the whole classes (six classes).

From Tables 2, 3, 4 and 5, it can be concluded that classification models are able to carry out the classification of the data set and get the right classification for the whole classes. Besides, proving that the use of classifier-ensembles to identify the attacks improves the use the baseline classifiers in one value between 5 and 50% according to the different classifiers and the class studied. Therefore, these models can classify different conditions of the attack network that are associated with the scans labelled as classes 3 and 4. In Table 6 the characteristics and options of the chosen ensembles, together with their tuned values are shown.

Table 6. Selected options of the ensembles for experimental study.

5. Conclusions and Future Work

This study has proposed a mutation testing model for classifiers ensembles performing ID on numerical traffic data sets. It aims to assess the generalization capability of the applied classifier ensembles when confronting potential zero-day attacks.

Experimental results show that the applied classifier ensembles deal properly with the analyzed data, containing mutated network scans. It can then be concluded that the applied models are able to properly detect new attacks related to previously unseen scans.

Future work will be based on the mutation of some other attack situations and the broadening of considered classifiers and ensembles.

Acknowledgments

This research has been partially supported through the projects of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 and CIT-020000-2009-12 (both funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT. Project also funded by the MICINN, the Spanish Ministry of Science and Innovation PID 560300-2009-11 and the Regional Government of Castile-Leon CCTT/10/BU/0002. This work was also supported in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by the Operational Program 'Research and Development for Innovations' funded through the Structural Funds of the European Union and the state budget of the Czech Republic.

References

[1] Darpa intrusion detection evaluation.
[2] Kdd cup 1999 dataset.
[3] Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co, 1980.
[4] Vulnerability statistics report,. Cisco Secure Consulting, 2000.
[5] The top 10 most critical internet security threats (2000-2001 archive). SANS Institute, 2001.
[6] *Using specification-based intrusion detection for automated response*, 2003.

[7] K. Abdullah, C. Lee, G. Conti, and J. A. Copeland. Visualizing network data for intrusion detection. In *Sixth Annual IEEE Information Assurance Workshop - Systems, Man and Cybernetics*, pages 100–108, 2005.
[8] J. Allen, A. Christie, W. Fithen, J. Mchugh, J. Pickel, and E. Stoner. State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99-TR-028, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA 15213-3890, January 2000.
[9] N. Athanasiades, R. Abler, J. Levine, H. Owen, and G. Riley. Intrusion detection testing and benchmarking methodologies. In *IWIAS 2003: First IEEE International Workshop on Information Assurance*, pages 63–72, 2003.
[10] B. Bacca, J. Salvi, and X. Cufi. Appearance-based mapping and localization for mobile robots using a feature stability histogram. *ROBOTICS AND AUTONOMOUS SYSTEMS*, 59(10):840–857, OCT 2011.
[11] P. Baraldi, G. Gola, E. Zio, D. Roverso, and M. Hoffmann. A randomized model ensemble approach for reconstructing signals from faulty sensors. *EXPERT SYSTEMS WITH APPLICATIONS*, 38(8):9211–9224, AUG 2011.
[12] Bruno Baruque, Emilio Corchado, Aitor Mata, and Juan M. Corchado. A forecasting solution to the oil spill problem based on a hybrid intelligent system. *Information Sciences*, 180(10):2029 – 2043, 2010. Special Issue on Intelligent Distributed Information Systems.
[13] M. Bermúdez-Edo, R. Salazar-Hernádez, J. Díaz-Verdejo, and P. García-Teodoro. Proposals on assessment environments for anomaly-based network intrusion detection systems. In Javier Lopez, editor, *Critical Information Infrastructures Security*, volume 4347 of *Lecture Notes in Computer Science*, pages 210–221. Springer Berlin / Heidelberg, 2006.
[14] Michael Berry and Gordon Linoff. *Data Mining Techniques*. Wiley Computer Publishing, 2004.
[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.
[16] Rachel Blagojevic, Beryl Plimmer, John Grundy, and Yong Wang. Using data mining for digital ink recognition: Dividing text and shapes in sketched diagrams. *COMPUTERS & GRAPHICS-UK*, 35(5):976–991, OCT 2011.
[17] João B. D. Cabrera, Carlos Gutiérrez, and Raman K. Mehra. Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks. *Inf. Fusion*, 9:96–119, January 2008.
[18] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. Introduction to version 2 of the internet-standard network management framework, 1993.
[19] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. Simple network management protocol (snmp), 1990.
[20] Tsai Chih-Fong, Hsu Yu-Feng, Lin Chia-Ying, and Lin Wei-Yang. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.
[21] Emilio Corchado, Bruno Baruque, and Hujun Yin. Boosting unsupervised competitive learning ensembles. In *ICANN (1)*, pages 339–348, 2007.
[22] Emilio Corchado and Álvaro Herrero. Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing*, 11(2):2042ï¿½2056, 2011.
[23] Emilio Corchado, Álvaro Herrero, and José Manuel Sáiz. Detecting compounded anomalous snmp situations using cooperative unsupervised pattern recognition. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, editors, *15th International Conference on Artificial Neural Networks (ICANN 2005)*, volume 3697 of *LNCS*, pages 905–910. Springer, Heidelberg, 2005.
[24] Emilio Corchado, Álvaro Herrero, and José Manuel Sáiz. Testing cab-ids through mutations: on the identification of network scans. In Howlett R.J. Jain L.C. Gabrys, B., editor, *International Conference in Knowledge-Based and Intelligent Engineering & Information Systems (KES 2006)*, volume 4252 of *LNAI*, pages 433–441. Springer, Heidelberg, 2006.
[25] J. Davin, J. Galvin, and K. McCloghrie. SNMP Administrative Model. RFC 1351 (Historic), July 1992.
[26] H Debar and Benjamin Morin. Evaluation of the diagnostic capabilities of commercial intrusion detection systems. In Andreas Wespi, Giovanni Vigna, and Luca Deri, editors, *Recent Advances in Intrusion Detection*, volume 2516 of *Lecture Notes in Computer Science*, pages 177–198. Springer Berlin / Heidelberg, 2002. 10.1007/3-540-36084-0_10.
[27] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.
[28] Pedro Garcia-Teodoro, Jesús E. Dáz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009.
[29] Debar H, Dacier M, Wespi A, and Lampart S An. Experimentation workbench for intrusion detection systems. research report rz 2998. *IBM Reserarch Division Zurich Research Laboratory*, 1998.
[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
[31] Nima Hatami. Thinned ecoc decomposition for gene expression based cancer classification. In Pan, JS and Abraham, A and Chang, CC, editor, *Eigth international conference on intelligent systems design and applications ISDA 2008*, volume 1, pages 216–221, 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, CA 90720-1264 USA, 2008. IEEE; IEEE SMC Soc; Natl Sci Council; Minist Educ, IEEE COMPUTER SOC. 8th International Conference on Intelligent Systems Design and Applications (ISDA 2008), Kaohsiung, Taiwan, nov 26-28, 2008.
[32] R. P. Lippmann J. W. Haines, L. M. Rossey and R. Cunningham. Extending the darpa off-line intrusion detection evaluations. *DARPA Information Survivability Conference and Exposition II*, pages 77–78, 2001.
[33] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, October 2000.
[34] Álvaro Herrero, Emilio Corchado, Paolo Gastaldo, and Rodolfo Zunino. Neural projection techniques for the visual inspection of network traffic. *Neurocomputing*, 72(16-18):3649–3658, 2009.
[35] M. Malowidzki. Getbulk worth fixing. *The Simple Times, 10*, 1:3–6, 2002.
[36] J. McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur.*, 3:262–294, November 2000.
[37] Patrick Mueller and Greg Shipley. Cover story: dragon claws its way to the top. *Network Computing*, 12:45–67, August 2001.

[38] Art Munson and Rich Caruana. Cluster ensembles for network anomaly detection. Technical Report 2006-2047, Cornell University, September 2006.

[39] S. Nawata, M. Uchida, Yu Gu, M. Tsuru, and Y. Oie. Unsupervised ensemble anomaly detection through time-periodical packet sampling. In *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, 2010.

[40] Stephen Northcutt, Mark Cooper, Karen Fredericks, and Matt Fearnow. *Intrusion Signatures and Analysis*. New Riders Publishing, Thousand Oaks, CA, USA, 2001.

[41] C. Schafer P. Laskov, P. Dussel and K. Rieck. Learning intrusion detection: Supervised or unsupervised? In F. Roli and S. Vitulano, editors, *ICIAP*, pages 50–57. 13th International Conference on Image Analysis and Processing, (Springer, Heidelberg, 2005), 2005.

[42] V. Hu P. Mell and R. Lippmann. An overview of issues in testing intrusion detection systems. *NIST Interagency Reports. National Institute of Standards and Technology - Information Technology Laboratory*, 2003.

[43] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[44] Nicholas Puketza, Mandy Chung, Ronald A. Olsson, and Biswanath Mukherjee. A software platform for testing intrusion detection systems. *IEEE Softw.*, 14:43–51, September 1997.

[45] Nicholas J. Puketza, Kui Zhang, Mandy Chung, Biswanath Mukherjee, and Ronald A. Olsson. A methodology for testing intrusion detection systems. *IEEE Trans. Softw. Eng.*, 22:719–729, October 1996.

[46] M. J. Ranum. Experiences benchmarking intrusion detection systems. Technical report, 2001.

[47] Pin Ren, Yan Gao, Zhichun Li, Yan Chen, and Benjamin Watson. Idgraphs: Intrusion detection and analysis using stream compositing. *IEEE Comput. Graph. Appl.*, 26:28–39, March 2006.

[48] J. M. Rizza. Computer network security. *Springer US*, 2005.

[49] D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information Systems*, 7(1):1–10, 2000.

[50] Maheshkumar Sabhnani and Gürsel Serpen. Why machine learning algorithms fail in misuse detection on kdd intrusion detection data set. *Intell. Data Anal.*, 8(4):403–415, 2004.

[51] B Schuller, BJB Schmitt, D Arsic, S Reiter, M Lang, and G Rigoll. Feature selection and stacking for robust discrimination of speech, monophonic singing, and polyphonic music. In *2005 IEEE International Conference on Multimedia and Expo (ICME), Vols 1 and 2*, pages 840–843, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2005. IEEE, IEEE. IEEE International Conference on Multimedia and Expo (ICME), Amsterdam, NETHERLANDS, JUL 06-08, 2005.

[52] Javier Sedano, Alba Berzosa, Jose R. Villar, E Corchado, and Enrique de la Cal. Optimising operational costs using soft computing techniques. *INTEGRATED COMPUTER-AIDED ENGINEERING*, 18:313–325, 2011.

[53] Martin Sewell. Ensemble learning. Technical report, Univeristy College London - Department of Computer Science, 2011.

[54] A.J.C. Sharkey and N.E. Sharkey. Diversity, selection and ensembles of artificial neural nets. In *Third International Conference on Neural Networks and their Applications. IUSPIM*, pages 205–212, 1997.

[55] Amanda J. C. Sharkey and Noel E. Sharkey. Combining diverse neural nets. *Knowledge engineering review*, 12(3):231–247, 1997.

[56] R. Sprenkels and J. P. Martin-Flatin. Bulk transfers of mib data. *Communication Systems Division. Swiss Federal Institute of Technology Lausanne*, 1999.

[57] Stuart Staniford, James A. Hoagland, and Joseph M. McAlerney. Practical automated detection of stealthy portscans. *J. Comput. Secur.*, 10:105–136, July 2002.

[58] Aurobindo Sundaram. An introduction to intrusion detection. *Crossroads*, 2:3–7, April 1996.

[59] Gang Wang and Jian Ma. Study of corporate credit risk prediction based on integrating boosting and random subspace. *EXPERT SYSTEMS WITH APPLICATIONS*, 38(11):13871–13878, OCT 2011.

[60] D Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.