Different approaches for the detection of SSH anomalous connections

S. GONZÁLEZ*, *Instituto Tecnológico de Castilla y León, C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001, Burgos, Spain.*

Á. HERRERO**, *Department of Civil Engineering, University of Burgos, Avenida de Cantabria s/n, 09006 Burgos, Spain.*

J. SEDANO†, *Instituto Tecnológico de Castilla y León, C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001, Burgos, Spain.*

URKO ZURUTUZA‡, *Electronics and Computing Department, Mondragon University, Goiru Kalea, 2, 20500 Arrasate-Mondragon, Spain.*

E. CORCHADO§, *Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced, s/n, 37008 Salamanca, Spain.*

Abstract

The Secure Shell Protocol (SSH) is a well-known standard protocol, mainly used for remotely accessing shell accounts on Unix-like operating systems to perform administrative tasks. As a result, the SSH service has been an appealing target for attackers, aiming to guess root passwords performing dictionary attacks or to directly exploit the service itself. To identify such situations, this article addresses the detection of SSH anomalous connections from an intrusion detection perspective. The main idea is to compare several strategies and approaches for a better detection of SSH-based attacks. To test the classification performance of different classifiers and combinations of them, SSH data coming from a real-world honeynet are gathered and analysed. For comparison purposes and to draw conclusions about data collection, both packet-based and flow data are analysed. A wide range of classifiers and ensembles are applied to these data, as well as different validation schemes for better analysis of the obtained results. The high-rate classification results lead to positive conclusions about the identification of malicious SSH connections.

Keywords: Secure Shell Protocol, SSH, honeynet, intrusion detection, classifier, ensemble, cross-validation.

1 Introduction

The Secure Shell Protocol (SSH) is a standard application-layer (under the TCP/IP stack) protocol for remote login and is also used for other secure network services over an insecure network. It consists

*E-mail: silvia.gonzalez@itcl.es
**E-mail: ahcosio@ubu.es
†E-mail: javier.sedano@itcl.es
‡E-mail: uzurutuza@mondragon.edu
§E-mail: escorchado@usal.es

doi:10.1093/jigpal/jzv047 Advance Access published 20 October 2015

of three major components:

- Transport Layer Protocol: it provides server authentication, confidentiality and integrity with perfect forward secrecy.
- User Authentication Protocol: it authenticates the client to the server.
- Connection Protocol: it multiplexes the encrypted tunnel into several logical channels.

The main usage of the SSH protocol is for remotely accessing shell accounts on Unix-like operating systems. As a result, most of the tasks and activities performed over this protocol are related to administrative purposes, such as user management, device configuration, permission assignment, etc. For this reason, the SSH service has been for years an attractive target for attackers, as crucial information travel over it. Intruders then try to guess passwords for malicious purposes, performing dictionary attacks, or to directly exploit the service itself. Weak passwords are targeted as there is no need for attackers to get the password of a root user account; there are many ways to increase the privileges of a user once logged-in. Furthermore, getting SSH access to remote hosts may be one of the first steps for further attacks over SSH tunnelling, such as SPAM sending.

Differentiating from other remote-communication protocols (File Transfer Protocol or Telnet), SSH encrypts the login session as a prevention mechanism to avoid the collection of unencrypted data (passwords and some other crucial data). Actually, SSH was conceived as a secure protocol to replace previous unsecure solutions to run sessions on remote host. However, the SANS Institute's Internet Storm Center [24] keeps monitoring an average of 100,000 targets being attacked on SSH default port number every day on Internet. From time to time, some attack peaks are produced, as the one on March 2014 (8× baseline).

As a result of the above mentioned, being able to distinguish between malicious and benign SSH traffic for server administration may play an indispensable role in defending system administrators against malicious adversaries. This is one of the targets of intrusion detection systems (IDSs) [6, 7, 9], which have become an essential asset in addition to the computer security infrastructure of most organizations. In the context of computer networks, an IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack. 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)]. While prevention mechanisms are aimed at avoiding intrusions, ID relies on the idea that intrusions will succeed and then they must be identified for security response. In this case, identification of SSH anomalous connections while they are being run certainly is an important task as it may reduce the impact of attacks, thanks to abortion mechanisms.

The aim of this study is to assess data collection, classifiers and ensembles in the useful task of detecting bad-intentioned SSH connections. As a result, best practices for SSH connection filtering may be proposed. To do so, real data, coming from the Euskalert honeynet [10], are gathered and analysed as described in the remaining sections of the article.

A honeypot has no authorized function or productive value within the corporate network other than to be explored, attacked or compromised [4]. Thus, a honeypot should not receive any traffic at all. Any connection attempt with a honeypot is then an attack or attempt to compromise the device or service that it is offering. From the security point of view, there is a great deal of information that may be learnt from a honeypot about a hacker's tools and methods to improve the protection of information systems.

In a honeynet, all the traffic received by the sensors is suspicious by default. Thus, every packet should be considered as an attack or at least as a piece of a multistep attack. But, in the case of SSH, a honeynet also receives legitimate connections for the administration of the honeynet itself.

As a result, for this study, data from 'normal' SSH connections are also available. Numerous studies propose the use of honeypots to detect automatic large-scale attacks, honeyd [22] and nepenthes [2] among others. The first Internet traffic monitors known as Network Telescopes, Black Holes or Internet Sinks were presented by Moore *et al*. [21].

The Euskalert honeynet [16], whose SSH data are analysed in this article, has been monitoring attacks against well-known services. A network of honeypots has been deployed in the Basque Country (northern Spain), where eight companies and institutions have installed one of the project's sensors behind the firewalls of their corporate networks. The honeypot sensor transmits all the traffic received to a database via a secure communication channel. These partners can consult information relative to their sensor (after a login process) as well as general statistics in the project's website. Once the system is fully established, the information available can be used to analyse attacks suffered by the honeynet at network and application level. Euskalert is a distributed honeypot network based on a Honeynet GenIII architecture [11]. As previously mentioned, the Euskalert sensors have also recorded the SSH sessions used to administer and maintain the different devices of the infrastructure.

ID has been previously approached from several different points of view; many different computational intelligence techniques—such as Genetic Programming [1], Data Mining [5, 12, 17], Expert Systems [18], Fuzzy Logic [29] or Neural Networks [15, 25, 31] among others—together with statistical [20] and signature verification [23] techniques have been applied mainly to perform a two-class classification (normal/anomalous or intrusive/non-intrusive). More precisely, attacks to SSH service have attracted researchers' attention for a long time. Song *et al*. [28] analysed timing and keystroke attacks. Researchers have also used honeypots to study and analyse attacks to this protocol, focusing on login attempts and dictionary attacks [8, 19]. In Koniaris *et al*. [19], the authors analyse SSH attacks on honeypots focusing on visualization of the gathered data.

Considering the data capture, as previously introduced, this study takes advantage of the Euskalert project. Its data have been analysed and processed in different ways to determine the best approach for the detection of SSH anomalous connections.

In this contribution, Section 2 presents the approaches under study, applied to the SSH data described in Section 3. The experimental results are described in Section 4, while some conclusions and lines of future work are introduced in Section 5.

2 Proposed approaches

Many different formulae could be applied for the detection of SSH-based attacks. This study analyses some of them, based on three main stages:

1. Data collection: network data may be summarized in several different ways. In this work, honeynet data are proposed to be collected at the packet level and as TCP flows.
2. Data analysis: several different classifiers and classifier ensembles are proposed as different combinations for the modelling of SSH connections.
3. Result evaluation: there exist different cross-validation (CV) schemes to check the significance of supervised classification results. In this work, 10K-fold and 5×2 CV have been applied.

Further details of the proposed approaches for each one of these stages are described in the following sections.

2.1 Data collection

As previously mentioned, data from Euskalert honeynet regarding SSH sessions are targeted in this work. Those data are proposed to be analysed in two different ways.

TABLE 1. Collected features for SSH packets

Feature	Description
Src	IP address of the source host
Timestamp	Daytime when the packet was sent
Size	Size (number of bytes) of the packet
Numflags	Amount of different flags used

TABLE 2. Collected features for SSH sessions

Feature	Description
Src	IP address of the source host
Length	Duration of the session
Numpac	Number of packets that the source host sent
Minlen	Minimum size of the packets
Maxlen	Maximum size of the packets
Size	Average size of the packets
Numflags	Amount of different flags used

First, data are collected at packet level. It means that the values in the headers at different layers are extracted for further analysis. Table 1 shows the features considered for every single packet targeting the honeynet address pool and SSH port.

Then, the traffic has been processed to obtain the SSH sessions out of the packets. The approach for defining an SSH session was based on the TCP logic, using packets with the same source IP, same destination IP and a common source and destination port. Source port is a non-privileged port number that remains the same during any TCP session. The features that were extracted from each one of the sessions in the data set are described in Table 2.

2.2 Data analysis

One of the most interesting features of IDSs would be their capability to automatically detect whether a portion of the traffic circulating the network is an attack or normal traffic. This task is more challenging when confronting brand-new bad-intentioned activities with no previous samples.

Automated learning models (classifiers) [3] are well-known algorithms designed specifically for the purpose of deciding about previously unseen data. This issue makes them suitable for the IDS task. Going one step further, ensemble methods [27] combine multiple algorithms into one usually more accurate than the best of its components. Hence, the main idea behind ensemble learning is to take advantage of classification algorithms diversity to face more complex data [26]. For this reason, this study proposes the combination of classifiers to get more accurate results when detecting anomalous and intrusive events.

A wide variety of automated learning techniques have been applied in this study to classify SSH connections. Several base classifiers as well as different ways of combining them have been considered for the analysis of Euskalert data. These base classifiers have been combined according to

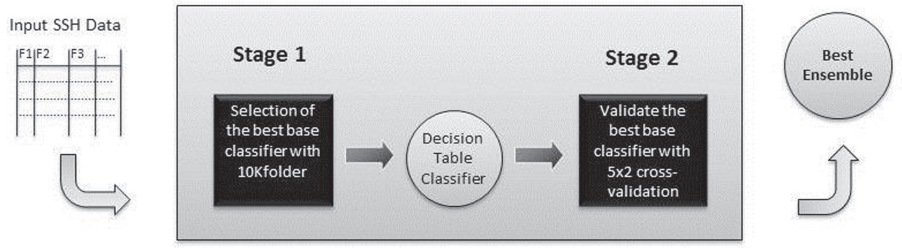

FIG. 1. The proposed two-stage method of validation.

the ensemble paradigm by the following strategies: Bagging, Boosting, Adaboost, MultiboostingAB and Rotation Forest.

2.3 Validation schemes

For a more comprehensive study, two different CV schemes have been applied in this work, namely 10K-fold and 5×2 [30]. They are applied to check how suitable are the learning models for the addressed problem (detection of SSH malicious connections). In case where the best model is found for both CV schemes when applied to a certain data arrangement, it could be said that the classifier and the ensemble are clearly the best ones.

To fulfil the requirements of the ID task, a two-stage process is proposed, as depicted in Figure 1. The first stage was already discussed in González *et al*. [13] and was carried out by using the features of the SSH sessions to test the classification performance of different classifiers and ensembles, by means of WEKA software [14]. The best combination (with a classification rate of 100%) of a base classifier and an ensemble was obtained by combining the decision table classifier and the Adaboost ensemble.

Nevertheless, for real-life SSH ID it is important to take into account that the amount of training data will always be smaller than the amount of data gathered when working on a real context. As a consequence, the selection of data analysis models (both base classifier and ensemble) should be assessed with a more realistic validation scheme. Thereby, a second stage has been accomplished to assure that the chosen classifier may be applied in a real situation, where there are usually more samples for testing than for training. In the second stage, proposed in this article, the base classifier 'decision table' is applied. This classifier has been trained again in combination with five different ensembles for verifying that it is suitable for real data. For this new training, a 5×2 CV scheme has being used. The new ensembles and validation schemas have been developed specifically.

3 Experimental validation on real data

The performance of the proposed approaches has been assessed using real data sets, coming from the Euskalert project [10]. Detailed information about the data and the run experiments is provided in this section.

The experimental study has been performed by extracting SSH data related to 34 months of real attacks and administration tasks that reached the eight sensors of the Euskalert project. Data from such a long time period guarantee that a broad variety of situations are considered. This honeynet system receives 4000 packets a month on average.

TABLE 3. Range of features for SSH packets

Feature	Type	Anomalous	Administration
Src	inet	–	–
Timestamp	Time	00:00:02–23:58:20	00:02:40–23:09:37
Size	Integer	40–220	40–380
Numflags	Integer	2–194	2–24

TABLE 4. Range of features for SSH sessions

Feature	Type	Anomalous	Administration
Src	inet	—	—
Time	Interval	00:00:00–352 days 09:48:19.891	00:00:00.004–519 days 18:24:05.446
Numpac	Integer	1–95	1–23
Minlen	Integer	40–64	40–380
Maxlen	Integer	40–220	40–380
Avglen	Numeric (8, 2)	40–96	40–380
Numflags	Integer	1–6	1–4

The complete data set from Euskalert contains a total of 2,647,074 packets, including TCP, UDP and ICMP traffic received by the distributed honeypot sensors. For this experiment, SSH data from connections that happened between May 2008 and March 2011 have been selected. Additionally, traffic containing real attacks to the SSH port and SSH connections to the system management port have been filtered out. This way, system management traffic is considered as benign traffic, and SSH connections coming from unknown IP sources are considered as malicious.

As stated in Section 2.1, two different data sets have been generated: the first data set contains the packet-level data and amounts to 209 administration (legitimate) packets and 37,990 anomalous packets. Table 3 shows the range of each selected feature (described in Table 1), depending on the nature of the session (administration or anomalous).

Out of the packets, the session-based data set was also extracted, containing 82 administration sessions and 8477 anomalous sessions. Table 4 shows the range of each feature, depending on the nature of the session (administrator or attack).

4 Results

This section presents the results obtained by the different approaches proposed in this article. For clarity and brevity, the average classification rate (%) for each ensemble applied to the n folders is provided, together with the maximum value and the standard deviation in Tables 5 and 6. Figures 2 and 3 show the boxplots associated with the classification rates of the different ensembles when applying the two alternative validation schemes. For independent analysis, classification rates for both packet-based and TCP-sessions data sets are presented. The results are then discussed for each one of the stages to which alternatives are proposed.

As can be seen in Tables 5 and 6, the best-performance classifier in the different proposed alternatives is Bagging. Thus, Table 7 shows the confusion matrices for such ensemble. The amounts

TABLE 5. Classification results from ensembles on packet-level data set

		10K-fold			5 × 2 CV		
No.	Ensemble	Average	Max.	Deviation	Average	Max.	Deviation
1	Bagging	99.93	99.97	0.000474790	99.92	99.95	0.000158140
2	Boosting	99.83	99.97	0.000981427	99.79	99.87	0.000550786
3	Adaboost	99.83	99.95	0.000792667	99.78	99.90	0.000640341
4	MultiboostingAB	99.84	99.92	0.00646569	99.85	99.87	0.00022946
5	Rotation Forest	99.91	99.97	0.000413983	99.85	99.88	0.00017873
–	Average	99.867	99.958	0.000661887	99.838	99.892	0.00035149

TABLE 6. Classification results from ensembles on session-based data set

		10K-fold			5 × 2 CV		
No.	Ensemble	Average	Max.	Deviation	Average	Max.	Deviation
1	Bagging	99.82	100	0.00099298	99.74	99.84	0.0008003
2	Boosting	99.61	100	0.00270065	99.07	99.58	0.0035945
3	Adaboost	99.47	99.88	0.0022871	99.18	99.46	0.0029858
4	MultiboostingAB	99.46	99.65	0.00157671	99.32	99.51	0.0016225
5	Rotation Forest	99.72	100	0.00228562	99.58	99.70	0.0012137
–	Average	99.616	99.907	0.001968612	99.378	99.617	0.00204341

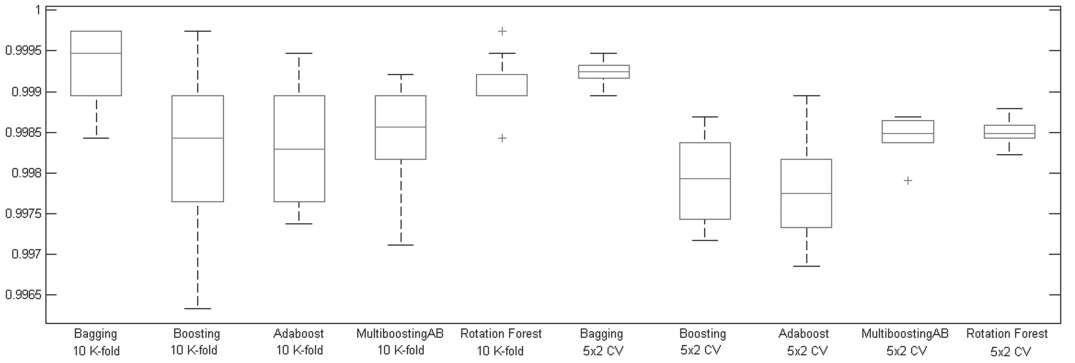

FIG. 2. Boxplot results from ensembles on packet-level data set.

in this table are the cumulative results, according to the validation schemes (10K and 5 × 2). In the case of the 10K folder scheme, data in confusion matrices are the sum of the results for the 10 folders. As each one of them applies to 10% of the data, the total amount of data is the size of the data set (10 times 10%), i.e. 38,199 packets and 8559 sessions. On the other hand, for the 5 × 2 validation scheme, each folder comprises 50% of the data and there are 10 folders as well. Thus, the matrices on the right side of Table 7 are related to 190,995 packets and 42,795 (10 times 50%).

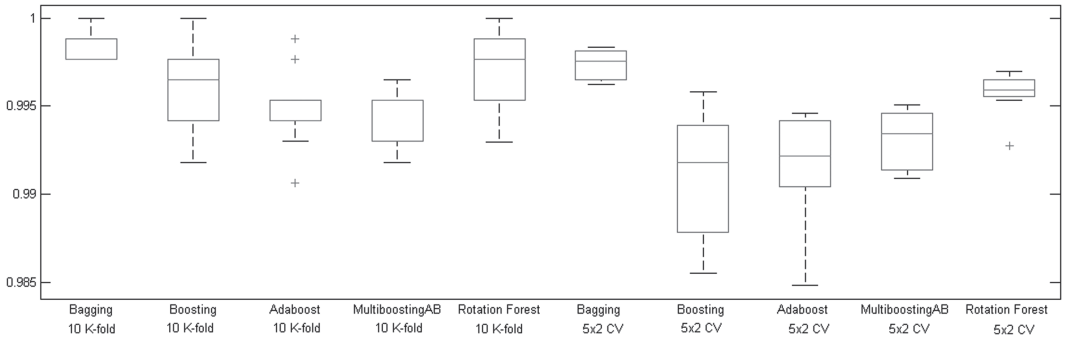

FIG. 3. Boxplot results from ensembles on session-based data set.

TABLE 7. Confusion matrix for the best results—Bagging ensemble

			10 K-fold			5x2 CV		
			Output class			Output class		
			Admin	Attack		Admin	Attack	
SSH packets	Target class	Admin	198 (00.518%)	11 (00.028%)	(00.547%)	951 (00.498%)	94 (00.049%)	(00.547%)
		Attack	17 (00.044%)	37973 (99.408%)	(99.453%)	53 (00.028%)	189897 (99.425%)	(99.453%)
			(00.562%)	(99.438%)		(00.526%)	(99.474%)	
SSH sessions	Target class	Admin	68 (00.794%)	14 (00.164%)	(00.958%)	322 (00.753%)	88 (00.206%)	(00.009%)
		Attack	1 (00.012%)	8476 (99.030%)	(99.042%)	24 (00.056%)	42361 (98.985%)	(99.041%)
			(00.806%)	(99.194%)		(00.808%)	(99.192%)	

One of the main issues to be highlighted from Tables 5 and 6 is that Bagging is the ensemble attaining the best performance in all cases (different data sets and validation schemes). Its average classification rate (for the 10 classifiers) is always superior to that of other ensembles. Additionally, the best classification rate (max.) for a single classifier is always one of the highest for such ensemble.

According to the data collection issue, it can be seen that classification results are on average (for the different ensembles) higher in the case of packet-based SSH data set. Furthermore, the maximum classification rates are also obtained for this data set, despite the fact that it reaches 100% for some of the base classifiers combined with certain ensembles on session data set.

Through the evaluation by 10K versus 5×2, as shown in Tables 5 and 6, it can be said that 5×2 is more strict as the classification rates are always lower than in the case of 10K-fold (except for the MultiboostingAB when applied to packet data set). However, for the best results (Table 7), it can be said that false-negative rate (FNR) is lower for 5×2 in the case of SSH packet data set.

From Table 7, it can be said that false-positive rates (FPRs) and FNRs are good enough to validate the proposed stages of SSH ID. The FNR reaches its lowest value (00.012%) when validating the SSH session data set by 10K, as only one attack session is misclassified. In the case of packet-based

data set, the FNR is lower for the 5 × 2 CV scheme. On the other hand, the FPR takes its lowest value (00.028%) when validating the SSH packet data set by 10K.

5 Final conclusions

As the SSH is mainly used for administration purposes, and hence it manages critical information, it is potentially a dangerous protocol. In this article, several strategies and approaches are compared to validate its performance on the detection of SSH-based attacks.

The following main conclusions can be drawn from the proposed alternatives at the different ID stages, according to the experimental results in previous section:

1. Data collection: by gathering data at the packet level, the obtained classification results are better (on average) than those from session-based data.
2. Data analysis: Bagging is the ensemble attaining the best performance in all cases.
3. Result evaluation: 5 × 2 is more strict than 10K-fold.

The promising classification results obtained by ensemble classifiers in this study could be applied to other network protocols and services. Mainly, it would be interesting its application to the attacks received by the honeynets, such as those based on HTTP, SNMTP, or even FTP. Interesting knowledge could be obtained from the honeypots classification models that will later prevent detected attacks from surpassing the organization networks and causing any damage.

Funding

This research was partially supported through projects of the Spanish Ministry of Economy and Competitiveness with ref: TIN2010-21272-C02-01 (funded by the European Regional Development Fund) and Junta de Castilla y León projects SA405A12-2 and BIO/BU09/14.

References

[1] A. Abraham, C. Grosan and C. Martin-Vide. Evolutionary design of intrusion detection programs. *International Journal of Network Security*, **4**, 328–339, 2007.

[2] P. Baecher, M. Koetter, T. Holz, M. Dornseif and F. Freiling. The nepenthes platform: an efficient approach to collect malware. In *9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006)*, vol. 4219, pp. 165–184. Springer, 2006.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[4] K. A. Charles. Decoy systems: a new player in network security and computer incident response. *International Journal of Digital Evidence*, **2**, 2004.

[5] S. Chebrolu, A. Abraham and J. P. Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, **24**, 295–307, 2005.

[6] T. Chih-Fong, H. Yu-Feng, L. Chia-Ying and L. Wei-Yang. Intrusion detection by machine learning: a review. *Expert Systems with Applications*, **36**, 11994–12000, 2009.

[7] Computer Security Threat Monitoring and Surveillance. *Technical Report*. James P. Anderson Co., 1980.

[8] D. D. Coster and D. Woutersen. Beyond the SSH Brute force attacks. In *10th GOVCERT.NL Symposium*, 2011.

[9] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, **13**, 222–232, 1987.
[10] Euskalert, Basque Honeypot Network. http://www.euskalert.net
[11] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data-analysis. *IEEE Transactions on Computers*, **23**, 881–890, 1974.
[12] G. Giacinto, F. Roli and L. Didaci. Fusion of multiple classifiers for intrusion detection in computer networks. *Pattern Recognition Letters*, **24**, 1795–1803, 2003.
[13] S. González, J. Sedano, U. Zurutuza, E. Ezpeleta, D. Martínez, Á. Herrero and E. Corchado. Classification of SSH anomalous connections. In Á. Herrero, .B. Baruque, F. Klett, A. Abraham, V. Snášel, A. C. P. L. F. Carvalho, P. G. Bringas, I. Zelinka, H. Quintián and E. Corchado, eds. *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, vol. 239, pp. 479–488. Springer International Publishing, 2014.
[14] 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**, 10–18, 2009.
[15] A. Herrero, E. Corchado, P. Gastaldo and R. Zunino. Neural projection techniques for the visual inspection of network traffic. *Neurocomputing*, **72**, 3649–3658, 2009.
[16] Á. Herrero, U. Zurutuza and E. Corchado. A neural-visualization IDS for honeynet data. *International Journal of Neural Systems*, **22**, 1–18, 2012.
[17] K. Julisch. Data mining for intrusion detection: a critical review. In *Applications of Data Mining in Computer Security*, D. Barbará and S. Jajodia, eds, pp. 33–62. Kluwer Academic Publishers, 2002.
[18] H. K. Kim, K. H. Im and S. C. Park. DSS for computer security incident response applying CBR and collaborative response. *Expert Systems with Applications*, **37**, 852–870, 2010.
[19] I. Koniaris, G. Papadimitriou and P. Nicopolitidis. Analysis and visualization of SSH attacks using honeypots. In *IEEE European Conference on Computer as a Tool (IEEE EUROCON 2013)*, 2013.
[20] D. J. Marchette. *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint*. Springer, 2001.
[21] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker and S. Savage. Inferring Internet denial-of-service activity. *ACM Transactions on Computer Systems*, **24**, 115–139, 2006.
[22] N. Provos. A virtual honeypot framework. In *Proceedings of the 13th Conference on USENIX Security Symposium*, vol. 132, 2004.
[23] M. Roesch. Snort–lightweight intrusion detection for networks. In *13th Systems Administration Conference (LISA '99)*, pp. 229–238, 1999.
[24] SANS Institute's Internet Storm Center. https://isc.sans.edu/port.html?port=22
[25] S. T. Sarasamma, Q. M. A. Zhu and J. Huff. Hierarchical Kohonenen net for anomaly detection in network security. *IEEE Transactions on Systems Man and Cybernetics, Part B*, **35**, 302–312, 2005.
[26] J. Sedano, A. Berzosa, J. R. Villar, E. Corchado and E. de la Cal. Optimising operational costs using soft computing techniques. *Integrated Computer-Aided Engineering*, **18**, 313–325, 2011.
[27] G. Seni and J. Elder. Ensemble methods in data mining: improving accuracy through combining predictions. Morgan and Claypool Publishers, 2010.
[28] D. X. Song, D. Wagner and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of the 10th Conference on USENIX Security Symposium*, vol. 10, p. 25. USENIX Association, 2001.
[29] A. Tajbakhsh, M. Rahmati and A. Mirzaei. Intrusion detection using fuzzy association rules. *Applied Soft Computing*, **9**, 462–469, 2009.

[30] J. Villar, S. González, J. Sedano, E. Corchado, L. Puigpinós and J. de Ciurana. Meta-heuristic improvements applied for steel sheet incremental cold shaping. *Memetic Computing*, **4**, 249–261, 2012.
[31] C. Zhang, J. Jiang and M. Kamel. Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters* **26**, 779–791, 2005.

Received 31 July 2014