Web Service Intrusion Detection Using a Probabilistic Framework

Hassen Sallay, Sami Bourouis, and Nizar Bouguila

1 Introduction

Recent advances in web technologies and the constant growth of the Internet have led to many online service applications. Examples include e-commerce, social networks, online banking, business intelligence, web search engines, etc. An important feature of these web services is that they are based on software applications running at the server-side and generating new web content in an online fashion, which makes them flexible to exchange information on the Internet [32, 10, 29]. The flexibility of web services poses also vulnerabilities which make them the targets for attacks (e.g. code injection attacks, SQL/XML injection, buffer overflow attacks, denial of service, etc.) by cyber-criminals who can collect confidential information from servers or even compromise them [34, 9, 40, 14, 17]. Then, there is an urgent need to protect the servers on which the applications are running [45, 44, 18, 8]. Indeed, intrusion detection systems (IDSs) need to be deployed. An overview of current intrusion detection techniques and related issues was proposed in [42, 38]. Recently, data mining and machine learning approaches have been used in this growing area in order to improve the performance of existing systems [31, 22, 41, 12, 16, 20]. The key idea for these works is to use machine learning techniques (e.g. decision trees, artificial neural networks, support vector machines, mixture models, etc.) to train a classifier and to recognize attacks based on a list of features, which generally reduces the intrusion detection problem to an adversarial learning task [24].

Based on the analysis methods, IDSs are usually classified into two main categories: misuse (i.e. signature-based) detection and anomaly detection systems [35]. In misuse detection systems, the goal is to detect the occurrence of attacks that have been previously identified as intrusions. For this type of IDS, attacks must be known a priori. Misuse detection can be viewed then as a supervised learning problem. Alternatively, anomaly detection systems detect unknown attacks by observing deviations from normal activities of the system. It is based on the assumption that intrusive activities are noticeably different from normal system activities and hence detectable. Data clustering and unsupervised learning approaches have been widely used to develop anomaly detection systems. Several of recent clustering approaches quantify deviation from normal behavior using thresholds (see, for instance, [33, 45, 44]). Unlike these approaches we consider a robust finite Gaussian mixtures to model normal traffic and then to automatically detect potential intrusions (i.e. anomalous traffic). Our main idea is based on incorporating into the Gaussian mixture an auxiliary outlier component, to which we associate a uniform density, to represent abnormal requests. The resulting model is learned using an expectation-maximization algorithm.

The rest of this paper is organized as follows: the proposed web service intrusion detection model is described in Section 2. Then, obtained results using a data set containing both normal and intrusive requests which were collected from a large real-life web service. are given and analyzed in Section 3. Finally, Section 4 concludes the paper.

2 A finite mixture model with outliers detection

In this section, we present our mixture model for web service intrusion detection and the motivations behind it. Moreover, we propose a detailed approach to learn the parameters of the proposed model.

H. Sallay (✉)
Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia
e-mail: hmsallay@imamu.edu.sa

S. Bourouis
Taif University, Taif, Saudi Arabia
e-mail: s.bourouis@tu.edu.sa

N. Bouguila
Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

H. Selvaraj et al. (eds.), *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*, Advances in Intelligent Systems and Computing 1089,
DOI 10.1007/978-3-319-08422-0_24, © Springer International Publishing Switzerland 2015

2.1 The model

The mixture model Let us consider a training data set of N vectors $\chi = X_1, \ldots, X_N$, where each $X_i = (X_{i1}, \ldots, X_{iD})$ is a D-dimensional vector of features representing a given request on a web server. Set of vectors generally contains examples that belong to many clusters (i.e. categories of requests) and can be modeled by a finite mixture of distributions [28]

$$p(X_i|\Theta_K) = \sum_{k=1}^{K} p_k\, p(X_i|\theta_k) \tag{1}$$

where $p_k > 0$, are the mixing proportions, M is the number of mixture components, and $\Theta_K = P = (p_1, \ldots, p_K), \theta = (\theta_1, \ldots, \theta_K)$ is the set of parameters in the mixture model. A critical problem in this case is the choice of the probability density function to represent each component. In this paper, we consider a classic choice namely a Gaussian distribution with parameters μ_k and Σ_k:

$$p(X_i|\theta_k) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}_k} \exp\left(-\frac{1}{2}(X_i - \mu_k)^T \Sigma_k^{-1}(X_i - \mu_k)\right) \tag{2}$$

Outliers detection Legitimate requests are generally in higher number than malicious ones. Thus, it is possible to formalize abnormal requests as outliers when considering statistical models. Many outliers detection approaches have been proposed in the past [15, 1, 43]. Indeed, the problem is a fundamental classic task in data mining and is generally related to fraud detection problems in the security domain (e.g. credit-card fraud, fraudulent cellular call detection, etc.) [39]. Here we approach the problem by incorporating an auxiliary outlier component, to which we associate a uniform density [37, 19, 4, 25], into the mixture model:

$$p(X_i|\Theta_K) = \sum_{k=1}^{K} p_k p(X_i|\theta_k) + p_{K+1}U(X_i) \tag{3}$$

where $p_{K+1} = 1 - \sum_{k=1}^{K} p_k$ is the probability that X_i was not generated by the central mixture model and $U(X_i)$ is a uniform distribution common for all data to model isolated vectors which are not in any of the K clusters and which show significantly less differentiation among clusters. Assuming uniform distribution for outliers is a common assumption that has been previously used successfully in [37, 43, 26]. It is noteworthy that when $p_{M+1} = 0$ the outlier component is removed and the previous equation is reduced to Eq. 1.

2.2 Model Learning

The most widely used approach for unknown parameters estimation is maximum likelihood (ML) based on maximizing the log-likelihood function as following

$$\begin{aligned}\widehat{\Theta} = \arg\max_{\Theta}\Bigg\{ & \log p(\chi|\Theta) \\ & = \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} p_k p(X_i|\theta_k) + p_{K+1}U(X_i)\right\}\end{aligned}$$

ML estimation is generally performed with expectation maximization (EM) algorithm [27] which is well-known in the case of finite mixture models. The main modification, in our case, is related to the E-step in which the posterior probabilities are calculated as following in iteration q:

$$\begin{cases} \gamma_{ik}^{(q)} := P\left(c_k/X_i, \theta_k^{(q)}\right) := \dfrac{p_k^{(q-1)} p\left(X_i|\theta_k^{(q-1)}\right)}{\sum_{k=1}^{K} p_k^{(q-1)} p\left(X_i|\theta_k^{(q-1)}\right) + p_{K+1}^{(q-1)} U(X_i)} & \text{for} \quad k = 1, \ldots, K \\ \gamma_{i(K+1)}^{(q)} := \dfrac{p_{K+1}^{(q-1)} U(X_i)}{\sum_{k=1}^{K} p_k^{(q-1)} p\left(X_i|\theta_k^{(q-1)}\right) + p_{K+1}^{(q-1)} U(X_i)} & \end{cases} \tag{4}$$

where $\gamma_{i(K+1)}$ is the probability of affecting the vector X_i to the set of outliers (or abnormal requests). Thus, the complete estimation algorithm is summarized as follows:

1. Initialization-step: initialization using the K-means algorithm

For each iteration q:

2. Expectation-step: Calculate ${\gamma_{ik}}^{(q)}$, $i = 1, \ldots, N$, $k = 1, \ldots, K + 1$ using Eq. 4.
3. Maximization-step:

$$\begin{cases} p_k^q := \dfrac{\sum_{i=1}^{N} \gamma_{ik}^{(q)}}{N} \\ \mu_k^q := \dfrac{\sum_{i=1}^{N} \gamma_{ik}^{(q)} p\left(X_i|\theta_k^{(q-1)}\right)}{\sum_{i=1}^{N} \gamma_{ik}^{(q)}} \\ \Sigma_k^q := \dfrac{\sum_{i=1}^{N} \gamma_{ik}^{(q)} \left(X_i - \mu_k^{(q)}\right)\left(X_i - \mu_k^{(q)}\right)^t}{\sum_{i=1}^{N} \gamma_{ik}^{(q)}} \end{cases} \quad (5)$$

The iterations in the previous algorithm are repeated until we reach convergence. Examples of convergence tests include the stabilization of the likelihood function, the parameters, or the posterior probabilities. Concerning the determination of K, the number of mixture components, which can be viewed as the number of request categories, different selection criteria have been proposed in the past and could be considered [28, 6]. In this work, we use the mixture minimum description length (MMDL) criterion developed in [13]:

$$MMDL(K) = -\log(p(\chi|\Theta)) + \frac{1}{2}N_p\log(N) + \frac{c}{2}\sum_{k=1}^{K}\log(p_k) \quad (6)$$

where N_p is the total number of free parameters in the model and c is the number of parameters describing each component. We select the K that yields the minimum value of $MMDL(M)$. Moreover, according to our operational definition of outliers, they should have a uniform distribution, since they do not follow the pattern of the majority of the data. A common approach, to define this uniform distribution, is to suppose that the data follow a single component model averaged over all the observation [37]. Thus, in our case, we choose the following:

$$U(X) = \frac{1}{N}\sum_{i=1}^{N} p\left(X_i|\widehat{\theta}\right) \quad (7)$$

where the parameters $\widehat{\theta}$ is estimated using ML technique. This formulation takes into account the fact that outliers should be sparsely distributed.

3 Experimental Results

The proposed framework is tested using logs collected from a real-life web service (from several Apache servers) in a two weeks time interval. The collected data set contains normal requests, anomalies as well as intrusions. More specifically, our training data is collected at the beginning and is composed of 10000 requests. The majority of these requests are legitimate, but some are attacks (e.g. cross-site scripting, SQL injections, buffer overflows, etc.). After using these data to train our mixture model, by considering 3-gram representation as done in [44], new requests are considered and classified as normal or abnormal (i.e. outlier) using the technique proposed in this paper. These new requests constitute the testing set and their number is equal to 35000. It is noteworthy that these data are used also to update the model using the approach proposed in [36]. Other incremental versions of the EM algorithm such as those in [30, 23] could be used, also. Updating the model's parameters allows to take into account new request categories and new intrusion pattern which didn't appear in initial training data but may emerge in future data. It is noteworthy that this is crucial in practice in order to plan and design new countermeasures.The evaluation of the performance of our approach has been based on the following measures:

- True positive rate which represents the number of correctly detected intrusions over the number of intrusions in the testing set.
- False positive rate which represents the number of normal requests considered as intrusions over the total number of normal requests in the testing set.
- True negative rate which represents the number of correctly classified normal requests over the total number of normal requests in the testing set.
- False negative rate which represents the number of misclassified intrusions over the number of intrusions in the testing set.
- Accuracy which represents the number of correctly classified requests over the total number of requests in the testing set.
- Precision which represents the number of correctly classified intrusions over the number of intrusions

The performance results of our approach are presented in Table 1. According to this table, it is clear that our approach provides excellent detection results and that the different N-gram approaches perform comparably. We compared our algorithm with the SDEM and SDPU approaches in [39] based on Gaussian mixture models and kernel mixtures

Table 1 Performance of the method in detecting web service intrusion when considering 1-gram, 2-gram and 3-gram models to describe features.

	1-gram	2-gram	3-gram
True positive rate	98.02	98.03	98.12
False positive rate	0.97	0.98	0.98
True negative rate	98.60	98.63	98.77
False negative rate	1.04	1.01	1.01
Accuracy	97.89	97.92	97.95
Precision	98.04	98.13	98.21

Table 2 Performance of the SDEM method in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	97.90	97.97	98.09
False positive rate	1.00	1.01	1.02
True negative rate	98.56	98.60	98.65
False negative rate	1.02	1.02	1.05
Accuracy	97.87	97.92	97.92
Precision	98.02	98.11	98.17

Table 3 Performance of the SDPU method in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	97.91	97.98	98.08
False positive rate	1.01	1.02	1.03
True negative rate	98.50	98.65	98.66
False negative rate	1.03	1.03	1.05
Accuracy	97.85	97.90	97.91
Precision	98.08	98.13	98.18

Table 4 Performance of K-nearest neighbor method in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	95.02	95.09	95.15
False positive rate	1.33	2.22	2.13
True negative rate	94.14	94.35	94.41
False negative rate	2.51	2.43	2.39
Accuracy	94.38	94.56	94.77
Precision	94.66	94.73	94.85

Table 5 Performance of the growing hierarchical self organizing maps (GHSOMs) [45] method in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	98.01	98.01	98.08
False positive rate	1.02	1.02	1.01
True negative rate	98.60	98.60	98.65
False negative rate	1.07	1.04	1.04
Accuracy	97.70	97.76	97.88
Precision	98.08	98.09	98.17

as shown in Tables 2 and 3, respectively. Moreover, we performed comparisons with the well-known nearest-neighbor technique as shown in Table 4 and three recent state of the art approaches, namely GHSOMs [45], diffusion maps [21], and the algorithm proposed in [44] as shown in Tables 4, 5, and 6, respectively. The results shown in all the tables demonstrate that our statistical framework is promising.

Table 6 Performance of diffusion maps method [21] in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	98.00	98.00	98.03
False positive rate	1.07	1.06	1.06
True negative rate	98.55	98.56	98.63
False negative rate	1.24	1.24	1.15
Accuracy	97.74	97.74	97.77
Precision	98.03	98.08	98.09

Table 7 Performance of the algorithm proposed in [44] in detecting web service intrusion.

	1-gram	2-gram	3-gram
True positive rate	98.04	98.04	98.05
False positive rate	1.03	1.03	1.02
True negative rate	98.65	98.68	98.68
False negative rate	1.06	1.10	1.10
Accuracy	97.79	97.79	97.81
Precision	98.15	98.18	98.19

4 Conclusion

Machine learning techniques have been widely used recently for computer security purposes [7]. Following this interesting trend, we introduce in this paper a new method to detect intrusion attacks on web services using a finite mixture model. The proposed finite Gaussian mixture model is augmented with an auxiliary uniform component to detect suspicious requests which are viewed as outliers. The proposed statistical framework is learned using an EM algorithm. The proposed technique is theoretically reliable and robust and has been validated using real-world data extracted from real web services. There are many avenues for future research. For instance, it is possible to extend the proposed mixture model to the infinite case using Dirichlet processes which allow to model outliers implicitly. Another promising future work is to integrate feature selection within the proposed framework or to consider other mixture models [5, 3, 2, 11]. Other applications such as spam filtering and credit card fraud detection are also possible.

Acknowledgments. The first author would like to thank King Abdulaziz City for Science and Technology (KACST), Kingdom of Saudi Arabia, for their funding support under grant number 11-INF1787-08.

References

1. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley & Sons (1994)
2. Bouguila, N., Ziou, D.: Dirichlet-based probability model applied to human skin detection. In: Proc. of the IEEE International

Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. 521–524 (2004)

3. Bouguila, N., Ziou, D.: A powerful finite mixture model based on the generalized Dirichlet distribution: Unsupervised learning and applications. In: Proc. of the 17th International Conference on Pattern Recognition (ICPR). pp. 280–283 (2004)
4. Bouguila, N., Almakadmeh, K., Boutemedjet, S.: A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection. Expert Systems with Applications 39(7), 6641–6656 (2012)
5. Bouguila, N., Ziou, D.: Using unsupervised learning of a finite dirichlet mixture model to improve pattern recognition applications. Pattern Recognition Letters 26(12), 1916–1925 (2005)
6. Bouguila, N., Ziou, D.: Unsupervised selection of a finite dirichlet mixture model: An mml-based approach. IEEE Transactions on Knowledge and Data Engineering 18(8), 993–1009 (2006)
7. Chan, P.K., Lippmann, R.: Machine learning for computer security. Journal of Machine Learning Research 6, 2669–2672 (2006)
8. Corona, I., Giacinto, G.: Detection of server-side web attacks. In: Diethe, T., Cristianini, N., Shawe-Taylor, J. (eds.) WAPA. JMLR Proceedings, vol. 11, pp. 160–166. JMLR.org (2010)
9. Dagdee, N., Thakar, U.: Intrusion attack pattern analysis and signature extraction for web services using honeypots. In: Proc. of the First International Conference on Emerging Trends in Engineering and Technology (ICETET). pp. 1232–1237 (2008)
10. Desmet, L., Jacobs, B., Piessens, F., Joosen, W.: Threat modelling for web services based web applications. In: Chadwick, D., Preneel, B. (eds.) Communications and Multimedia Security, IFIP The International Federation for Information Processing, vol. 175, pp. 131–144. Springer US (2005)
11. Elguebaly, T., Bouguila, N.: Bayesian learning of finite generalized gaussian mixture models on images. Signal Processing 91(4), 801–820 (2011)
12. Fan, W., Bouguila, N., Ziou, D.: Unsupervised anomaly intrusion detection via localized bayesian feature selection. In: Proc. of the EEE International Conference on Data Mining (ICDM). pp. 1032–1037 (2011)
13. Figueiredo, M.A.T., Leitão, J.M.N., Jain, A.K.: On fitting mixture models. In: Hancock, E.R., Pelillo, M. (eds.) EMMCVPR. Lecture Notes in Computer Science, vol. 1654, pp. 54–69. Springer (1999)
14. Gruschka, N., Luttenberger, N.: Protecting web services from dos attacks by soap message validation. In: Fischer-Hebner, S., Rannenberg, K., Yngstrm, L., Lindskog, S. (eds.) Security and Privacy in Dynamic Environments, IFIP International Federation for Information Processing, vol. 201, pp. 171–182. Springer US (2006)
15. Hawkins, D.M.: Identification of Outliers. Chapman and Hall, London (1980)
16. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 38(1), 306 – 313 (2011)
17. Jensen, M., Gruschka, N., Herkenhoner, R., Luttenberger, N.: Soa and web services: New technologies, new standards - new attacks. In: Proc. of the Fifth European Conference on Web Services (ECOWS). pp. 35–44 (2007)
18. Jensen, M., Gruschka, N., Herkenhener, R.: A survey of attacks on web services. Computer Science - Research and Development 24 (4), 185–197 (2009)
19. Ke, Q., Kanade, T.: Robust subspace clustering by combined use of *k*ndd and svd algorithm. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 592–599 (2004)
20. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB Journal 16(4), 507–521 (2007)
21. Kirchner, M.: A framework for detecting anomalies in http traffic using instance-based learning and k-nearest neighbor classification. In: Proc. of the 2nd International Workshop on Security and Communication Networks (IWSCN). pp. 1–8 (May 2010)
22. Laskov, P., Dessel, P., Schefer, C., Rieck, K.: Learning intrusion detection: Supervised or unsupervised? In: Roli, F., Vitulano, S. (eds.) Image Analysis and Processing (ICIAP), Lecture Notes in Computer Science, vol. 3617, pp. 50–57. Springer Berlin Heidelberg (2005)
23. Liang, P., Klein, D.: Online em for unsupervised models. In: Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 611–619. NAACL '09, Association for Computational Linguistics (2009)
24. Lowd, D., Meek, C.: Adversarial learning. In: Proc. of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 641–647 (2005)
25. Mashrgy, M.A., Bdiri, T., Bouguila, N.: Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowledge-Based Systems 59, 182–195 (2014)
26. Mashrgy, M.A., Bouguila, N., Daoudi, K.: A robust approach for multivariate binary vectors clustering and feature selection. In: Lu, B.L., Zhang, L., Kwok, J.T. (eds.) ICONIP (2). Lecture Notes in Computer Science, vol. 7063, pp. 125–132. Springer (2011)
27. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. New York: Wiley (1997)
28. McLachlan, G., Peel, D.: Finite Mixture Models. New York: Wiley (2000)
29. Mehdi, M., Bouguila, N., Bentahar, J.: Trustworthy web service selection using probabilistic models. In: Proc. of the IEEE 19th International Conference on Web Services (ICWS). pp. 17–24 (2012)
30. Neal, R.M., Hinton, G.E.: A new view of the em algorithm that justifies incremental and other variants. In: Learning in Graphical Models. pp. 355–368. Kluwer Academic Publishers (1993)
31. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448 – 3470 (2007)
32. Pearce, C., Bertok, P., Schyndel, R.: Protecting consumer data in composite web services. In: Sasaki, R., Qing, S., Okamoto, E., Yoshiura, H. (eds.) Security and Privacy in the Age of Ubiquitous Computing, IFIP Advances in Information and Communication Technology, vol. 181, pp. 19–34. Springer US (2005)
33. Pereira, H., Jamhour, E.: A clustering-based method for intrusion detection in web servers. In: Proc. of the 20th International Conference on Telecommunications (ICT). pp. 1–5 (2013)
34. Pinzen, C., Paz, J.F., Zato, C., Perez, J.: Protecting web services against dos attacks: A case-based reasoning approach. In: Romay, M., Corchado, E., Garcia Sebastian, M. (eds.) Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science, vol. 6076, pp. 229–236. Springer Berlin Heidelberg (2010)
35. S. Northcutt and J. Novak: Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing (2002)
36. Samé, A., Ambroise, C., Govaert, G.: An online classification em algorithm based on the mixture model. Statistics and Computing 17 (3), 209–218 (2007)
37. Titsias, M.K., Williams, C.K.I.: Sequentially fitting mixture models using an outlier component. In: Proc. of the 6th International Workshop on Advances in Scattering and Biomedical Engineering. pp. 386–393 (2003)
38. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Review: Intrusion detection by machine learning: A review. Expert Systems with Applications 36(10), 11994–12000 (2009)
39. Yamanishi, K., ichi Takeuchi, J., Williams, G.J., Milne, P.: On-line unsupervised outlier detection using finite mixtures with

discounting learning algorithms. Data Mining and Knowledge Discovery 8(3), 275–300 (2004)

40. Yee, C.G., Shin, W.H., Rao, G.S.V.R.K.: An adaptive intrusion detection and prevention (ID/IP) framework for web services. In: Proc. of the International Conference on Convergence Information Technology (ICCIT). pp. 528–534 (2007)
41. Zanero, S., Savaresi, S.M.: Unsupervised learning techniques for an intrusion detection system. In: Proc. of the ACM Symposium on Applied Computing (SAC). pp. 412–419. ACM (2004)
42. Zhou, C.V., Leckie, C., Karunasekera, S.: A survey of coordinated attacks and collaborative intrusion detection. Computers & Security 29(1), 124 – 140 (2010)
43. Zivkovic, Z., Krose, B.: An em-like algorithm for color-histogram-based object tracking. In: Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. I–798–I–803 Vol.1 (2004)
44. Zolotukhin, M., Hamalainen, T.: Detection of anomalous http requests based on advanced n-gram model and clustering techniques. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) Internet of Things, Smart Spaces, and Next Generation Networking, Lecture Notes in Computer Science, vol. 8121, pp. 371–382. Springer Berlin Heidelberg (2013)
45. Zolotukhin, M., Hamalainen, T., Juvonen, A.: Growing hierarchical self-organizing maps and statistical distribution models for online detection of web attacks. In: Cordeiro, J., Krempels, K.H. (eds.) Web Information Systems and Technologies, Lecture Notes in Business Information Processing, vol. 140, pp. 281–295. Springer Berlin Heidelberg (2013)