# A Parameter-Free Method for the Detection of Web Attacks

Gonzalo de la Torre-Abaitua[1(✉)], Luis F. Lago-Fernández[2], and David Arroyo[2]

[1] Instituto Nacional de Ciberseguridad, INCIBE, León, Spain
gonzalo.torre@incibe.es
[2] Departamento de Ingeniería Informática, Escuela Politécnica, Superior, Universidad Autónoma de Madrid, Madrid, Spain
{luis.lago,david.arroyo}@uam.es

**Abstract.** Logs integration is one of the most challenging concerns in current security systems. Certainly, the accurate identification of security events requires to handle and merge highly heterogeneous sources of information. As a result, there is an urge to construct general codification and classification procedures to be applied on any type of security log. This work is focused on defining such a method using the so-called Normalised Compression Distance (NCD). NCD is parameter-free and can be applied to determine the distance between events expressed using strings. On the grounds of the NCD, we propose an anomaly-based procedure for identifying web attacks from web logs. Given a web query as stored in a security log, a NCD-based feature vector is created and classified using a Support Vector Machine (SVM). The method is tested using the CSIC-2010 dataset, and the results are analysed with respect to similar proposals.

**Keywords:** Intrusion detection systems · Anomaly detection · Web attacks · Parameter-free anomaly detection · NCD

## 1 Introduction

The number of different devices connected to the Internet and the related traffic grows each year [1] popularisation, adoption and exploitation of cloud services, smartphones, and more recently the Internet of Things (IoT). From the perspective of an Information Technology (IT) security engineer, the irruption of these facilities has consequences over the classical concept of security perimeter [2]. This being the case, nowadays it is very difficult to properly authenticate and trace the entities in information systems. Moreover, even when authentication and traceability are performed adequately, entities' ubiquity leads to a vast collection of events records. Certainly, the management of information systems is highly dependent on the accurate generation of event logs, but also on its efficient treatment and interpretation. The complexity of this task increases with the diversification of the interfaces to access information systems, which makes


H. Pérez García et al. (eds.), *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceedings*, Advances in Intelligent Systems and Computing 649,
DOI 10.1007/978-3-319-67180-2_64

it necessary to build up data-driven procedures for the agile identification of security incidents [3].

Besides, we have to take into account that *one-size-fits-all* solutions for automatic data processing and classification are not possible [4]. Although we can design a generic procedure for such a goal, we have to evaluate its likelihood and, if possible, to assess its accuracy with respect to some form of expert knowledge [5]. Given this fact, it seems convenient to have a flexible and easy-to-configure tool for bridging data classification and evaluation in an adaptive way. On this point, parameter-free methods could be useful to avoid the burden associated to the proper selection of configuration parameters [6].

The main contribution of this paper is a procedure that is able to detect anomalies by analysing with the same method each source of information regardless of the specific nature of each log or event. The method is based on string comparison using distance metrics, in particular the Normalized Compression Distance (NCD) [7]. This method not only identifies anomalies, but it can also work as an auxiliary visualisation tool for security analysts. As a first step in the design of a generic procedure, we address here the case of the detection of web attacks. Indeed, albeit the goal is to have a generic procedure, it is necessary to start from a specific scenario, verify the goodness of the method, and afterwards proceed with its generalisation. The rest of the paper is organised as follows. Section 2 gives a taxonomy of the main techniques used in HTTP traffic anomaly detection. In Sect. 3 we describe how to use the NCD to design a new anomaly-based procedure for web attack detection. Section 4 explains the proposed methodology and the results obtained. And finally, in Sect. 5 the conclusions and future work lines are drawn.

## 2 Taxonomy of the Main Procedures for Anomaly Detection in HTTP Traffic

Anomaly detection provides a means to implement Intrusion Detection Systems (IDSs) [8]. An IDS is a mechanism able to detect malicious events and subsequently generate an alert [9]. There are two types of IDS: host IDS (HIDS) and Network IDS (NIDS). The former seeks for intrusions on a host, checking for changes at an operative system level, monitoring systems calls or the modification of critical files. On the other hand, NIDSs are targeted to the analysis of network activity, which comprises the checking of network connections and packets, URLs queries, and other network metadata. IDSs use two types of detections, one based in patterns and another based on anomaly identification. The first one is also known as misuse detection. It performs detection based on signatures with a high accuracy but its main drawbacks are the easiness to overpass it modifying the way the attack is done. As a result it cannot detect new attacks (0-day), which highlights the importance of having signatures up-to-date.

On the other hand, it is possible to adopt anomaly-based IDS. An anomaly is any event that is not expected or does not follow the normal behaviour in a system. They are also known as exceptions or outliers [10]. The main advantage of

this technique is that it is possible to identify new types of attacks. Nonetheless, its main drawback is the high rate of false positives since selecting the right set of features to be measured is something that is domain specific and based on the experience [11]. Therefore, it is needed a knowledge on the network and systems involved to manually tune it and reduce the false positive rate. Besides, several anomaly-based techniques have to perform a previous training phase to learn what the normal behaviour of the system is [11].

Regardless of these drawbacks, the benefits of anomaly-based IDSs have drawn the attention of the research community. Many surveys and taxonomies have been developed focusing on different aspects. In [12] the authors divide network anomaly detection techniques into six big groups: statistical, classification based, clustering and outlier based, soft computing, knowledge based and combination of learners. Complementarily, in [13] a big picture of the techniques employed in anomaly detection is given, focusing on the main advances achieved by using machine learning techniques, specifically neural and deep networks. Similarly, in [11] it is presented a review of data mining and machine learning techniques, classifying data mining techniques into three groups: classification, association and grouping. A different approach is followed in [8], where it is shown a taxonomy with three big groups of techniques: statistical based, knowledge based and machine learning based. Finally, there are other surveys done with a more concrete scope. In [14] it is provided a classification and description of different web application vulnerabilities and attacks. The work performed in [15] establishes a framework for testing anomaly-based IDS. It is focused on the six measures defined in [16]: length, character distribution measure, Markov Model, presence/absence of parameters, order of parameters, and whether parameter values were enumerated or random. A different approach is followed in [17], where the taxonomy is divided into HTTP attack detection and web attack detection. HTTP attack detection works upon HTTP packets, whereas web attack detection is targeted at web queries. For the sake of concretion, we summarise all the underlined techniques in Fig. 1.

Previous works were mainly developed on a parameter based URL anomaly detection. Here we explore the possibilities of a parameter-free approximation using NCD. Next section explains NCD and gives an overview of related works in the security anomaly detection field.

## 3 URL Anomaly Detection Method Based on NCD

One key component to build up anomaly detection procedures is to have a metric to compare events. Consequently, we could say that similarity can be interpreted as a basic brick in anomaly detection systems, although it is not an essential component. To the best of our knowledge, [45] is the only survey focused on the application of similarity measures for anomalies identification. From this work, it can be extracted that the most frequently used distance in clustering techniques and supervised classification is the Euclidean distance, although different distances and metrics are used as well, such as entropy, affinity, swap metric or permutations.

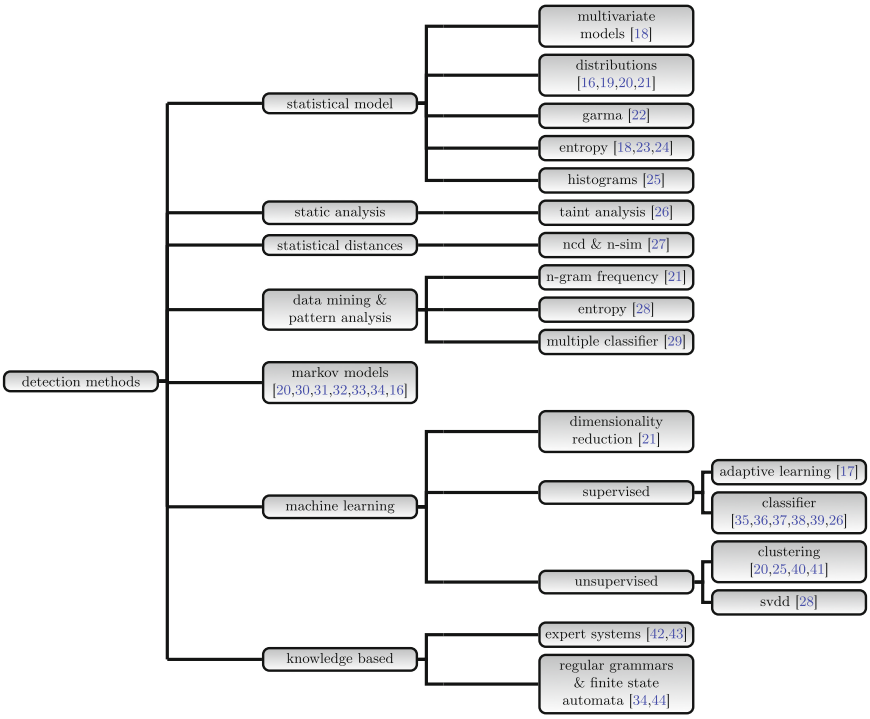


**Fig. 1.** Taxonomy of the main procedures for anomaly detection in HTTP traffic.

In this work we use the NCD, which is agnostic to the nature of the information source, universal and parameter-free. Moreover, it does not use any feature or previous knowledge about the data and it is able to extract information by itself [7]. The NCD is based on the use of compression methods to measure differences between two files or elements. This idea is established on the intuition that when you compress an element $x$ all the redundant information is removed. In this vein, the compressor keeps the minimum fundamental structure that allows to recover the whole document when it is decompressed. Hence, when two elements $x$ and $y$ are compressed better together than they do separately, we may assume that they contain similar information. This idea leads to the definition of the NCD, which was first described by Cilibrasi and Vitányi [46]. They found a practical approximation of the normalised Information Distance (NID), approximating the uncomputable Kolgomorov complexity by using real life compressors. Let $x$ and $y$ be two arbitrary items, such as strings or files. Let $C(x)$ denote the length of the compressed version of $x$, and let $xy$ be the concatenation of $x$ and $y$. Then the NCD between the two items is defined as follows:

$$NCD(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

It can be demonstrated that this distance is a metric, since it satisfies the identity, monotonicity, symmetry and distributive axioms. Although some compressors are not guaranteed to satisfy the identity axiom, Cilibrasi and Vitányi simplified their NCD formulation using $C(xy)$ instead of min{C(xy),C(yx)}, because they observed that blocking code compressors (such as gzip [47], bzip [48] or PPMZ [49]) are symmetric almost by definition, with only small deviations being observed in practice [50].

Therefore, the NCD seems to be a suitable tool for anomaly detection. It can be applied in unsupervised clustering algorithms or it can be used to detect differences between strings. The NCD has been successfully applied to the detection of masqueraders, based on user command line and enriched command line inputs obtaining an accuracy comparable to other statistical methods. However, the NCD has the advantage of not requiring a previous knowledge about the data [51,52]. In web attack detection, the NCD can successfully measure the similarities between URLs. As a result, it can help in the identification of URL clusters, which in the end paves the way for the distinction between normal URLs and malicious ones. In fact, it can be correctly applied to get accurate clusters [27].

Therefore, the NCD properties can be used to construct a parameter-free method for the identification of web attacks. In this work we use a similar approach based on normalized conditional compressed information [50] to derive a set of attributes that describe URL requests, as described in Sect. 4. Those attributes are subsequently used as the basis on a classification procedure that makes possible the discrimination of malicious URLs.

## 4 Methodology and Results

The methodology we follow in this paper is based on the identification of anomalies using the strings internal information. It is done by catching the similarities between a single string and different agglomerations of anomalous or normal web queries. Each string is then represented by a vector of features, where each feature is the distance between the string and one of these agglomerations. We test this approach using the CSIC-2010 dataset [53].

### 4.1 Data Preparation

The CSIC-2010 dataset contains both normal and anomalous HTTP queries. We have preprocessed these data by extracting the parameters of all the POST requests. After removing duplicates and balancing the number of normal and anomalous instances, we obtain a dataset with 9600 instances (4800 of each class), where each instance is a single text string. Finally, the strings are randomly sorted to prevent from any biases associated to query order.[1]

[1] The preprocessed data can be accessed upon request.

After preprocessing, data are partitioned in two subsets. The first one contains 1600 query strings that are used as instances to train and test the classifiers. We will refer to these data as *instance requests*. The second one, which contains the remaining 8000 requests (4000 of each class), is used to generate the attributes that describe each instance string. These requests are further partitioned into $k$ groups, each one containing the same number of requests from one single class. We will refer to each of these groups as an *attribute generator*. The attribute generators are used to generate a set of $k$ attributes describing each instance string, where each attribute measures the amount of information about the instance that is not contained in the corresponding generator. Instance $i$ is described by the attribute vector $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ik})$, where $x_{ij}$ is given by:

$$x_{ij} = D(attribute\, generator\, j, instance\, i) \tag{1}$$

and $D(a, b)$ is the normalized conditional compressed information [50], defined as:

$$D(a, b) = \frac{C(b|a)}{C(b)} = \frac{C(ab) - C(a)}{C(b)} \tag{2}$$

To compute the compressed size of $x$, $C(x)$, we use the gzip compressor, because it performs better than other algorithms in terms of speed [54]. Figure 2 describes all the preprocessing stage for $k = 8$ attribute generators, each one containing 1000 requests from one single class (normal or anomalous).

The number $D(a, b)$ measures the extra number of bits that are needed to compress the instance $b$ when added to the generator $a$, normalized to $C(b)$. It can be interpreted as the amount of information contained in $b$ but not in $a$, and so it is a measure of the dissimilarity between $a$ and $b$. Note that $D(a, b)$ as defined above is not the standard NCD distance. In particular we have $NCD(x, y) = D(E_{max}\{x, y\}, E_{min}\{x, y\})$, where $E_{max}\{x, y\}$ is the element in $\{x, y\}$ with the largest compressed size, and $E_{min}\{x, y\}$ is the element in $\{x, y\}$ with the smallest compressed size. This definition ensures the symmetry of the NCD, and also reflects the general intuition that a very long string $x$ and a very short one $y$ should be considered different, with $NCD(x, y)$ evaluating to a value close to 1. Our problem is however not symmetric, as we are trying to measure the similarity between a single string $b$ and a group of strings $a$, and so the use of $D(a, b)$ seems more natural in this case.

### 4.2 Results

The set of instance requests, described by the distances to the $k$ attribute generators, were used to train and test a classifier that discriminates between normal an anomalous HTTP queries. We applied a SVM classifier with RBF kernel, whose parameters were tuned using cross-validation. We used the *sklearn.svm.SVC* implementation from the *scikit-learn* library [55].

We performed experiments with the following 5 different values for the number of attributes: $k \in \{8, 16, 32, 80, 160\}$. For each $k$, 6 different and disjoint sets of 1600 instance requests are considered, using the remaining requests to

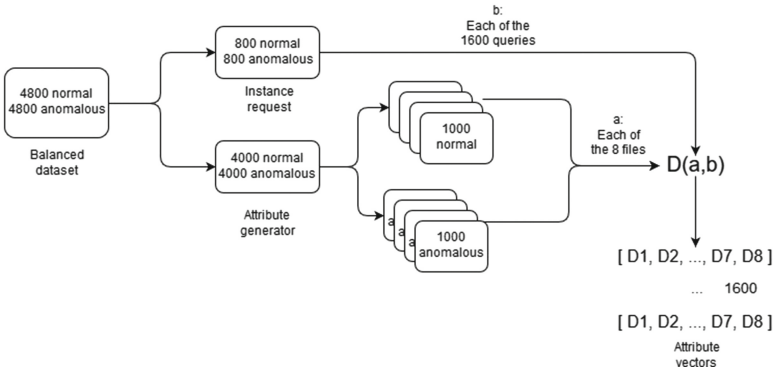


**Fig. 2.** Data preparation for k = 8 attribute generators.

generate the attributes as explained above. The first of these sets is used to tune the classifier parameters $C$ (complexity parameter) and $\gamma$ (kernel parameter) using 5-fold cross-validation. The other 5 sets are used to evaluate the classifier performance, measured by the accuracy and the area under the ROC curve (AUC). We use a 10-fold cross-validation scheme on each set and average over sets. The results are summarised in Table 1. We observe a general increase of the accuracy and the AUC as the number of attributes is increased up to some value where they saturate. The best results, with a mean accuracy of $0.95 \pm 0.02$ and a AUC of 0.975, are obtained for $k = 80$ attributes and 100 queries per generator file. These results are comparable to those reported in the literature [56]. However the present approach has some additional advantages: the features are constructed automatically, there is no need for feature selection, and the number of parameters is minimum (just those of the classifier).

**Table 1.** Results. Column names: $k$, number of attributes; $n_q$, number of queries in each attribute generator file; $C$, complexity parameter, obtained by cross-validation on the first set of instance queries; $\gamma$, kernel parameter, obtained by cross-validation on the first set; Acc., average accuracy over 10-fold cross-validation and over sets 2-6 of instance queries; AUC, average area under ROC.

| $k$ | $n_q$ | $C$ | $\gamma$ | Acc. | AUC |
|---|---|---|---|---|---|
| 8 | 1000 | 1.0 | 100.0 | $0.84 \pm 0.03$ | 0.909 |
| 16 | 500 | 10.0 | 10.0 | $0.88 \pm 0.03$ | 0.946 |
| 32 | 250 | 1.0 | 10.0 | $0.93 \pm 0.02$ | 0.968 |
| 80 | 100 | 1.0 | 10.0 | $\mathbf{0.95 \pm 0.02}$ | **0.975** |
| 160 | 50 | 2.0 | 5.0 | $0.95 \pm 0.02$ | 0.974 |

## 5 Conclusions

Current heterogeneity in logging systems makes necessary the establishment of a mechanism able to detect anomalies regardless of the concrete nature of each log. In this work we have addressed this problem for a specific use case: the detection of web attacks. Provided this context, we have evaluated the convenience of implementing a method for the identification of malicious URLs using similarity measures. Among the different options, we have chosen a parameter-free universal distance, the NCD. Although our approach does not always improve the best results obtained by other researches based on the same data, it leads to a similar performance avoiding the computational cost of any previous feature selection process.

According to the above comments, our future work will be devoted to further evaluate the accuracy of our methodology with respect to similar ones. Concerning logs integration, the procedure will be applied in the analysis of different types of logs, and not only in web logs examination. Finally, we will study in detail the outcomes of our procedure as a visualisation tool for security analysts. Certainly, visualisation techniques help analyst to understand the results obtained after logs analysis, and it also eases the identification of false positives [57,58]. Since our methodology endows security analysts with a means to compare an string with a set of strings, it could be a guide to identify groups of strings and thus to derive a data visualisation tool.

**Acknowledgements.** This work was supported by Comunidad de Madrid (Spain) under the project S2013/ICE-3095-CM (CIBERDINE) and the Spanish Government projects MINECO/FEDER DPI2015-65833-P and TIN2014-54580-R.

## References

1. Gartner: Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015. http://www.gartner.com/newsroom/id/3165317. Accessed 1 Apr 2017
2. Ziv, A.: The Ethereal Perimeter. https://www.linkedin.com/pulse/ethereal-perimeter-avishai-ziv. Accessed 1 Apr 2017
3. Curry, S., Kirda, E., Schwartz, E., Stewart, W.H., Yoran, A.: Big data fuels intelligence-driven security. RSA Security Brief (2013)
4. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
5. Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., Li, K.: $AI^2$: Training a big data machine to defend. In: 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 49–54. IEEE (2016)
6. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards parameter-free data mining. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 206–215. ACM (2004)

7. Cilibrasi, R., Vitanyi, P.: Automatic extraction of meaning from the web. In: 2006 IEEE International Symposium on Information Theory, pp. 2309–2313. IEEE (2006)
8. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**(1–2), 18–28 (2009)
9. Kruegel, C., Valeur, F., Vigna, G.: Intrusion Detection and Correlation: Challenges and Solutions, vol. 14. Springer Science & Business Media, Heidelberg (2004)
10. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 1–72 (2009)
11. Chaurasia, M.A.: Comparative study of data mining techniques in intrusion dectection. Int. J. Curr. Eng. Sci. Res. (IJCESR) **3**(9), 107–112 (2016)
12. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2014)
13. Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., Robert, A.: Shallow and deep networks intrusion detection system: a taxonomy and survey, pp. 1–43 (2017). ArXiv e-prints
14. Deepa, G., Thilagam, P.S.: Securing web applications from injection and logic vulnerabilities: approaches and challenges. Inf. Softw. Technol. **74**, 160–180 (2016)
15. Ingham, K.L., Inoue, H.: Comparing anomaly detection techniques for HTTP. In: Raid, vol. 4637, pp. 42–62 (2007)
16. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 251–261. ACM, New York (2003)
17. Dong, Y., Zhang, Y.: Adaptively Detecting Malicious Queries in Web Attacks. ArXiv e-prints (2017)
18. Bronte, R., Shahriar, H., Haddad, H.: Information theoretic anomaly detection framework for web application. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), pp. 394–399 (2016)
19. Kozik, R., Choraś, M., Renk, R., Hołubowicz, W.: Patterns extraction method for anomaly detection in HTTP traffic. In: International Joint Conference. Advances in Intelligent Systems and Computing, vol. 369, pp. 227–236. Springer, Cham (2015)
20. Corona, I., Giacinto, G.: Detection of server-side web attacks. J. Mach. Learn. Res. **11**, 160–166 (2010)
21. Juvonen, A., Sipola, T., Hämäläinen, T.: Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Comput. Netw. **91**, 46–56 (2015)
22. Pillai, T.R., Palaniappan, S., Abdullah, A.: Predictive modeling for intrusions in communication systems using GARMA and ARMA models. In: 2015 5th National Symposium on Information Technology: Towards New Smart World, NSITNSW 2015, pp. 1–6. IEEE (2015)
23. Shahriar, H., Zulkernine, M.: Information-theoretic detection of SQL injection attacks. In: Proceedings of IEEE International Symposium on High Assurance Systems Engineering, pp. 40–47. IEEE (2012)
24. Ashfaq, A.B., Javed, M., Khayam, S.A., Radha, H.: An information-theoretic combining method for multi-classifier anomaly detection systems. In: IEEE International Conference on Communications, pp. 1–5. IEEE (2010)
25. Zolotukhin, M., Hamalainen, T., Kokkonen, T., Siltanen, J.: Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic. In: 2016 23rd International Conference on Telecommunications, ICT 2016 (2016)

26. Medeiros, I., Neves, N., Correia, M.: Detecting and removing web application vulnerabilities with static analysis and data mining. IEEE Trans. Reliab. **65**(1), 54–69 (2016)
27. Yahalom, S.: URI anomaly detection using similarity metrics. Ph.D. thesis, Tel-Aviv (2008)
28. Zolotukhin, M., Hämäläinen, T., Kokkonen, T., Siltanen, J.: Analysis of HTTP requests for anomaly detection of web attacks. In: Proceedings - 2014 World Ubiquitous Science Congress: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, DASC 2014, pp. 406–411 (2014)
29. Moh, M., Pininti, S., Doddapaneni, S., Moh, T.S.: Detecting web attacks using multi-stage log analysis. In: Proceedings - 6th International Advanced Computing Conference, IACC 2016, pp. 733–738 (2016)
30. Song, Y., Keromytis, A.D., Stolfo, S.J.: Spectrogram: a mixture-of-markov-chains model for anomaly detection in web traffic. In: NDSS, p. 15 (2009)
31. Ye, N.: A markov chain model of temporal behavior for anomaly detection. In: Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, pp. 171–174 (2000)
32. Ariu, D., Tronci, R., Giacinto, G.: HMMPayl: an intrusion detection system based on Hidden Markov Models. Comput. Secur. **30**(4), 221–241 (2011)
33. Lampesberger, H., Winter, P., Zeilinger, M., Hermann, E.: An on-line learning statistical model to detect malicious web requests. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. LNICST, vol. 96, pp. 19–38 (2012)
34. Garcia-Teodoro, P., Diaz-Verdejo, J.E., Tapiador, J.E., Salazar-Hernandez, R.: Automatic generation of HTTP intrusion signatures by selective identification of anomalies. Comput. Secur. **55**, 159–174 (2015)
35. Jongsuebsuk, P., Wattanapongsakorn, N., Charnsripinyo, C.: Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In: International Conference on Information Networking, pp. 1–5. IEEE (2013)
36. Senthilnayaki, B., Venkatalakshmi, K., Kannan, A.: Intrusion detection using optimal genetic feature selection and SVM based classifier. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–4. IEEE (2015)
37. Akbar, S., Chandulal, J.A., Rao, K.N., Kumar, G.S.: Improving network security using machine learning techniques. In: 2012 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–5. IEEE (2012)
38. Enache, A.C., Sgârciu, V.: Anomaly intrusions detection based on support vector machines with an improved bat algorithm (2015)
39. Aburomman, A.A., Ibne Reaz, M.B.: A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl. Soft Comput. J. **38**, 360–372 (2016)
40. Lin, W.C., Ke, S.W., Tsai, C.F.: CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl. Based Syst. **78**(1), 13–21 (2015)
41. 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 Syst. Appl. **38**(1), 306–313 (2011)
42. Pan, Z., Lian, H., Hu, G., Ni, G.: An integrated model of intrusion detection based on neural network and expert system. In: 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005), vol. 2005, pp. 671–672 (2005)

43. Sheu, T.F., Huang, N.F., Lee, H.P.: NIS04-6: A time-and memory-efficient string matching algorithm for intrusion detection systems. In: Global Telecommunications Conference, GLOBECOM 2006, IEEE, pp. 1–5. IEEE (2006)
44. Fogla, P., Lee, W.: Evading network anomaly detection systems: formal reasoning and practical techniques. In: Proceedings of the 13th ACM conference on Computer and Communications Security, pp. 59–68 (2006)
45. Weller-Fahy, D.J., Borghetti, B.J., Sodemann, A.A.: A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Commun. Surv. Tutor. **17**(1), 70–91 (2015)
46. Vitányi, P.M.B., Balbach, F.J., Cilibrasi, R.L., Li, M.: Normalized information distance. In: Emmert-Streib, F., Dehmer, M. (eds.) Information Theory and Statistical Learning, pp. 45–82. Springer, Boston (2009)
47. GNU: gzip - GNU Project - Free Software Foundation. https://www.gnu.org/software/gzip/. Accessed 6 Apr 2017
48. bzip.org: bzip2: Home. http://www.bzip.org/. Accessed 6 Apr 2017
49. Bloom, C.: Charles Bloom's Page: Source Code: PPMZ. http://www.cbloom.com/src/ppmz.html. Accessed 6 Apr 2017
50. Cilibrasi, R., Vitányi, P.M.B.: Clustering by compression. IEEE Trans. Inf. Theory **51**(4), 1523–1545 (2005)
51. Bertacchini, M., Fierens, P.I.: Preliminary results on masquerader detection using compression based similarity metrics 2 previous work. Electron. J. SADIO **7**(1), 31–42 (2007)
52. Bertacchini, M., Benitez, C.E.: NCD based masquerader detection using enriched command lines. Innovation, vol. 4397 (2004)
53. CSIC-dataset: HTTP DATASET CSIC 2010. http://www.isi.csic.es/dataset/. Accessed 29 Mar 2017
54. Tukaani-project: A Quick Benchmark: Gzip vs. Bzip2 vs. LZMA. http://tukaani.org/lzma/benchmarks.html. Accessed 6 Apr 2017
55. Scikit learn: scikit-learn: machine learning in Python - scikit-learn 0.18.1 documentation. http://scikit-learn.org/stable/. Accessed 29 Mar 2017
56. Nguyen, H.T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., Franke, K.: Application of the generic feature selection measure in detection of web attacks. In: Computational Intelligence in Security for Information Systems, pp. 25–32. Springer (2011)
57. Marty, R.: Applied Security Visualization. Addison-Wesley, Upper Saddle River (2009)
58. Zhang, T.: Bridging the gap of network management and anomaly detection through interactive visualization. In: 2014 IEEE Pacific Visualization Symposium, pp. 253–257 (2014)