Shapelets-Based Intrusion Detection for Protection Traffic Flooding Attacks

Yunbin Kim[1], Jaewon Sa[1], Sunwook Kim[2], and Sungju Lee[1](✉)

[1] Department of Computer Information Science,
Korea University, Sejong 30019, Korea
{kyb2629, sjwon92, peacfeel}@korea.ac.kr
[2] Department of SW Contents Research Laboratory,
ETRI, Daejeon, Korea
swkim@etri.re.kr

Abstract. The intrusion detection for the network traffic is a technique to detect abnormal traffic flow patterns in periodic network packets. The traffic flooding attacks can be detected by the abnormal intrusion detection techniques that detects well known attack patterns. In this paper, we propose an intrusion detection way to classify normal and abnormal traffic packet pattern by converting traffic into time series data and analyzing them, and apply the information gain technique to reduce the learning execution times. That is, the normal and abnormal packet patterns are classified by applying the shapelets technique to the time-series pattern between the normal traffic and the abnormal traffic packet patterns. The experimental results show that the proposed method classifies normal patterns and traffic flooding attacks into 95% accuracy.

Keywords: Bigdata analysis · Intrusion detection · Traffic flooding attack
Time-series analysis · Shapletes analysis

1 Introduction

The big data analysis and processing are an important issue with increased IoT devices, and many researches have been studied by using machine learning, parallel processing, cloud computing, and sensors [1–5]. Recently, as IT technology has become more popular, personalized service is provided to users through various technologies including prediction model design in various fields, and reliability of personal information including network security has been highlighted accordingly [2]. In particular, big data, including sensitive data such as personal information, personally identifiable information, and intellectual property are likely to be exposed to cyber attacks and hacking. Therefore, it is very important and difficult to establish effective security for Big Data systems and services. In the network intrusion detection technology, the intrusion model can be divided into two types of normal and abnormal traffics. The anomaly based detection technique can generate a profile of a user's general pattern and analyzes patterns [6]. Therefore, the intrusion detection system can classify a pattern obtained from past intrusions [6].

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 227–238, 2018.
https://doi.org/10.1007/978-3-319-91455-8_20

DDoS (Distributed Denial of Service) attack, which is one kind of abnormal infiltration analyzed by abnormal-based intrusion detection technology, creates a large number of zombie PC remotely, and uses it to increase the traffic exponentially. In addition, the cases of traffic flooding attacks are continuously increasing, and thus an efficient detection technique for such abnormal intrusion attacks is required.

In this paper, we propose an intrusion detection way to detect and classify abnormal intrusion attacks by transforming normal and abnormal traffic packet patterns into time-series pattern, and apply shapelets analysis technique to transformed time-series data. The intrusion detection refers to analyzing and classifying network traffic data and structure abnormally. Also, we apply the information gain technique to features selection to reduce the learning execution times. Based on the experimental result, we confirmed that, proposed approach can provide 95% accuracy by using information gain and shapelets analysis.

The rest of this paper is structured as follows. Section 2 describes the related works on researches for the intrusion detection systems and the time-series data analysis methods. Section 3 describes packet characteristics of traffic flooding attacks and measurement methods, and how to apply the shapelets analysis and reduce the features by using information gain technique. Sections 4 and 5 describe the experimental results and conclusions.

2 Background

2.1 Intrusion Detection System

The intrusion detection systems protect the computer and mobile devices. Static analysis methods run the file and examine the contents of the file. Moser et al. [7] uses a number of virus/malware approaches such as bus conversion, noxiousness and variant techniques. Siddiqui et al. [8] protected the file system using the file function, which N-grams are sequences of bytes of a certain length, and contain bytes adjacent to each other [9]. Wavelet transform [10] is another source of file functionality. Bilar [11] proposed the mnemonic of the instruction using the predictor of the malicious program. Statistical machine learning and data science methods [12] have been increasingly used for malware detection, including approaches based on support vector machines, logistic regression, Naïve Bayes, neural networks, deep learning, wavelet trans-forms, decision trees and k-nearest neighbors [8, 10, 13–19].

The entropy analysis [10, 16, 20–22] is an effective technique for abnormal data detection by pointing to the possible 6080 presence of deception techniques. Despite polymorphism or obfuscation [23], files with high entropy are more likely to have encrypted sections in them. When an abnormal data switches between content regimes (*i.e.*, native code, encrypted section, compressed section, text, padding), there are corresponding shifts in its entropy time series [10]. In general, entropy analysis of data for intrusion detection, either the mean entropy of the entire data, or the entropy of chunks of code in sections of the file are computed. This simplistic entropy statistics approach may not be sufficient to detect expertly hidden malware, which for instance, may have additional padding (zero entropy chunks) to pass through high entropy filters.

2.2 Time-Series Clustering of Network Traffic

According to Keogh [24], clustering of time series can be categorized into two categories: full clustering and sub-clustering. Full clustering refers to grouping many individual time series into similar clusters or classes. Subsequence clustering refers to the use of sliding windows to extract subsequences from a single time series, and clustering is applied to the extracted subsequences. One of the most widely used approaches is hierarchical clustering. A similarity measure (*i.e.*, Euclidean distance) is applied to generate a pairwise distance matrix of primitive data. This approach generally applies to time series with the same length, but dynamic time warping (DTW) can be applied as a similarity measure to handle variable length time [25, 26]. The generation of the distance matrix is typically a computationally expensive operation for long time series [27]. Other widely used clustering algorithms such as *K*-means can also be applied to raw data [28]. By applying transformations to reduce the dimensionality of the data, as opposed to performing clustering on raw information, you can reduce the complexity of time series clustering. The purpose of the transformation is to first extract a specific function from the data, then apply a similarity measure and use the result as input to the clustering algorithm. In [27], the authors propose global feature extraction from individual time series (*i.e.*, trend, periodicity, and kurtosis, etc.). Time-series with similar global characteristics are clustered together. In [29], clustering is performed on the histogram representation of the data. Other transforms such as DFT (Discrete Fourier Transform) [30, 31], SVD (Singular Value Decomposition) [32] and APCA (Adaptive Piecewise Constant Approximation) [33] have also been proposed.

These transformations can extract the global properties of the time series. The main disadvantage of these approaches is the fact that when the local shape similarity is fundamental (*i.e.*, a sequence of signal strength measurements), the overall characteristics are not sufficient to adequately distinguish the time series. The use of wavelets has been discussed in the literature as a dimensional reduction technique that enables the extraction of localized shape features in the time domain [31, 34–37]. Transformations such as DFT can determine all spectral components in a time series, but cannot determine when these spectral components are present in the data (*i.e.*, time localization of features is not possible). The wavelet decomposition is provided to solve this problem. Recently, shape transformations have been proposed as approaches to cluster time series according to the shape [38–40, 44]. With this approach, a series of shapes (*i.e.*, subsequences with high discrimination power) are extracted from the time series collection.

2.3 Shapelets-Based on Time-Series Analysis

The shapelet is defined as a subsequence of one-time series in [38]. The subsequence S of length L is defined as a subset of one continuous value from the time series. The shapelets are selected by capturing unique shape features that are com-mon in time series classes. The shapelets can be found in L through a search of all possible subsequences of each time series as candidates for the shapelets [44]. However, this process is time consuming and a more efficient technique for shapelets generation has been proposed [38, 39]. The process of discovering shapelets for time-series clustering involves three main steps: creating candidates, measuring similarities between candidates and time series, and finally evaluating the quality of candidates. Regarding the generation of a shapelets candidates, it is first necessary to define the length of the candidate subsequence. Generally, a subsequence with a length between the predefined values l_{min} and l_{max} is considered. Using a generic search to generate the shapelets candidates, all possible subsequences with lengths between l_{min} and l_{max} are extracted from the time series of L. This process is slow and inefficient for large time series sets with long lengths. Rather than applying exhaustive search to create shapelets, we apply the algorithm proposed by Zakaria et al. In [39], we have made some modifications to accommodate the fact that we deal with time series of different lengths. [39], the authors proposed the use of unchecked it to collect time series.

3 Proposed Methods

In this paper, we use the NSL-KDD dataset, a quantified version of KDD CUP'99. The NSL-KDD dataset needs to extract useful features because it contains irrelevant data, redundant data, and noise data.

For this reason, data dimension reduction is performed through the information gain technique of feature selection. As a result of feature selection, the top 10 features with high weight are extracted and classified into normal and abnormal data by applying it to Shapelets, a machine learning technique.

Figure 1 shows the overall configuration of the proposed method.

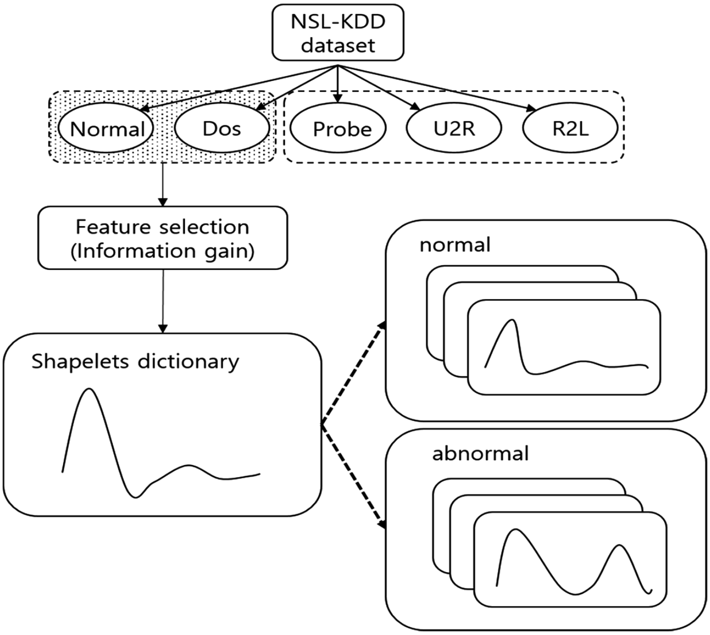


Fig. 1. The overall structure of the proposed method.

3.1 DoS Attack Detection Using Time Series Pattern

NSL-KDD [43] is a refined version of the data set KDD CUP'99, which is part of the DARPA scheme. It has four types of attacks: normal traffic packets and abnormal packets on the real network. Each attack consists of 41 features. In addition, the four attack types consist of DoS, Probe, U2R, and R2L. It also supports learning and test datasets, and the dataset features include protocol type, service, src_byte, and dst_byte, which are the contents of the network header, and access details such as host and guest logins. In this paper, we focus on traffic flooding attack using shapelets based time series analysis and use DoS attack traffic dataset among four attack types.

Table 1 shows the distribution of packets by four attack types (*i.e.*, DoS, Probe, U2R, R2L) provided by the NSL-KDD dataset.

Table 1. Distribution of packets by attack type.

Types of attack	# of packets	Ratio distribution of packet (%)
DoS	50,943	71
Probe	18,216	25
U2R	72	1
R2L	2,231	3
Sum of total	71,462	100

The four types of attacks provided by NSL-KDD are as follow.

- DoS: Denial of Service Attack, it is an attack that maliciously attacks a system and causes the system to run out of resources, thereby preventing its intended use.
- Probe: An attack that collects system vulnerabilities before attempting an actual attack.
- U2R: User to Root, it is an attack that attempts to gain administrator privileges.
- R2L: Remote to Local, an attack where an unauthorized user gains access from outside.

3.2 Reducing the Feature by Using Information Gain

The training data set of NSL-KDD consists of 4,756,832 packets. In the case of intrusion detection, it is necessary to learn about the types of attacks added periodically in order to detect new attack types. Therefore, a method for reducing the execution time of learning is needed. That is, when all the data sets are used, the accuracy of the overfitting may be reduced and the learning time may be increased.

The entropy is used for numerical operations to find the best conditions for separating data. This means the traffic flooding packets generated from the data set. If a given data set contains a lot of different types of values, the entropy is high, and if it is distributed over the same types of values, the entropy is set low. The entropy is used for numerical operations to find the best conditions for separating data. If there are many different kinds of results in a given data set, entropy is high, and entropy is low if the same kind of results exist. The entropy has a value between 0 and 1, that is, when entropy is 0, only the same types of data exists, and when entropy is 1, it is not separated at all.

$$E(S) = -\sum\nolimits_{x \in X} p(x)\log_2 p(x) \tag{1}$$

$$p(x) = \frac{freq(S_x)}{|S|} \tag{2}$$

The entropy calculated using Eq. (1) can be used to calculate a value of information gain that can distinguish data with high discrimination power.

$$Information\,Gain(S) = E_{high_{level}}(S) - \sum\nolimits_{t \in T} p(t)E_{low_{level}}(t) \tag{3}$$

$E_{high\_level}(S)$ is the entropy of a parent node, so that the information gain is the entropy of the parent node minus the entropy of the child node, taking into account the weights proportional to the number of records in the lower node.

In this paper, to solve the problem of decreasing the accuracy and increasing the learning time according to over-sum, we applied the information gain method [46] to select the top 10 features among the 41 features and apply it to the time-series analysis method. Table 2 lists the top 10 information gains used in this paper.

Table 2. Information gain-based feature selection.

No.	Features	Information gain score (Priority score)
1	Src_bytes	1.345491
2	Service	1.097208
3	Flag	0.962485
4	Diff_srv_rate	0.947248
5	Dst_host_diff_srv_rate	0.886182
6	Same_srv_rate	0.878879
7	Count	0.820505
8	Dst_host_same_srv_rate	0.800392
9	Dst_host_srv_count	0.777514
10	Dst_bytes	0.722053

4 Experimental Results

The shapelets technique is used to classify time series patterns and classifies them into several classes using subsequences extracted between time series patterns [41, 42, 44]. That is, the Euclidean distance is calculated for each traffic time series pattern using the extracted representative subsequence by learning several traffic time series patterns. Then, normal and abnormal binary classification is performed according to the criterion of the threshold value with respect to the calculated distance.

To effectively reduce the learning and test execution time of the shapelets technique, Fast-shapelets technique [41, 42, 44, 45] was applied and the shapelets were created to maximize the distance between normal and abnormal pattern classes.

After generating the shape through the Euclidean distance method in the normal traffic class and the abnormal traffic class, binary classification was performed based on the threshold values of the abnormal traffic class and the generated shapelets. We used labeled training and testing data sets to distinguish between normal and abnormal data. The training set consisted of 67,343 normal and 43,281 abnormal data, and the test set consisted of 9,710 normal and 5,076 abnormal data. In Fig. 1, each feature is set on the x-axis, and the packet value is shown on the y-axis (Fig. 2).

To classify the two labeled classes, a subsequence (i.e., Shapelet) was extracted through a learning process. It is confirmed that they are classified by mapping to normal and abnormal data using the extracted Shapelet. In this paper, we classify traffic flooding attack (*i.e.*, DoS) packet and normal data packet using shapelets algorithm, and confirm the classification process through the following decision tree.

In order to show the accuracy performance of applying the proposed approach to NSL-KDD packet data, we define the following three accuracy metrics (Fig. 3).

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

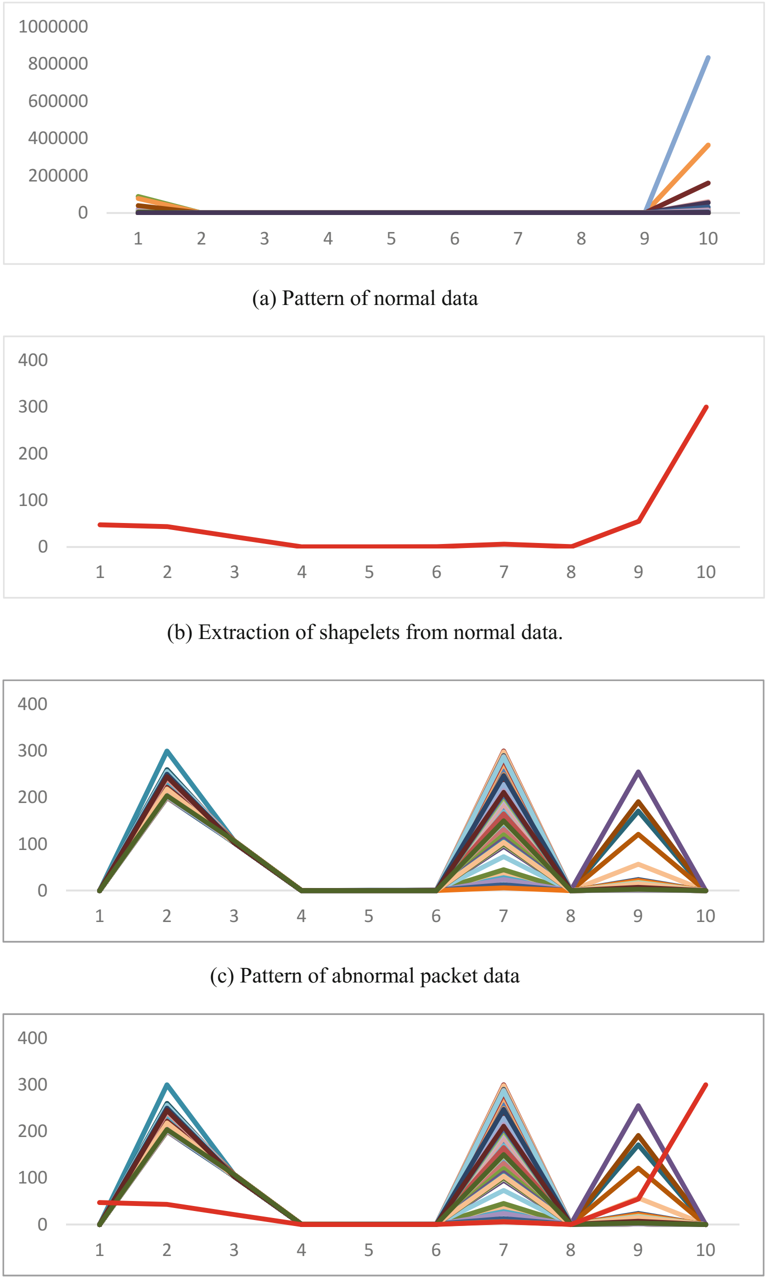
$$acurracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Experiments in this paper were performed on Intel Core i5-4690 3.5 GHz, 8 GB RAM environment. The data used in the experiment are the NSL-KDD data set, and the classifier for packet classification is shapelets.

In order to verify the validity of the proposed method, we conducted a comparative experiment with SVM, which is a typical technique in machine learning algorithms. The ratio of training and test data was constructed in the same way as the proposed method, and the kernel function used RBF (*i.e.*, Radial basis function).

Table 3 shows the metrics for classification of normal and abnormal packets for the SVM and the proposed method. SVM and the proposed method were verified through the *precision*, *recall*, and *accuracy*, which are measures to judge classification accuracy. The *precision* is the ratio of the number of the normal packets detected to the actual number of packets, and the *recall* is the ratio of the number of normal packets detected by the algorithm among the actual packets. And *accuracy* means total accuracy.

Table 4 shows the execution time of the learning by reducing the number of features through Information Gain. As a result of reducing the number of features, it is confirmed that the performance improvement is about 25 times higher than the execution time using all 41 features.



(a) Pattern of normal data

(b) Extraction of shapelets from normal data.

(c) Pattern of abnormal packet data

(d) Abnormal packet data and the shapelets.

Fig. 2. Pattern of the normal data and abnormal attack packet. (a) shows the normal data packet that keeps a low value and records a high value in the tenth feature (Dst_byte), and (b) shows an extraction of shapelets from normal data, (c) shows abnormal packet data that records a high value in for a second and seventh features, and (d) shows that the shapelets extracted from the normal data is applied to the abnormal packet data.

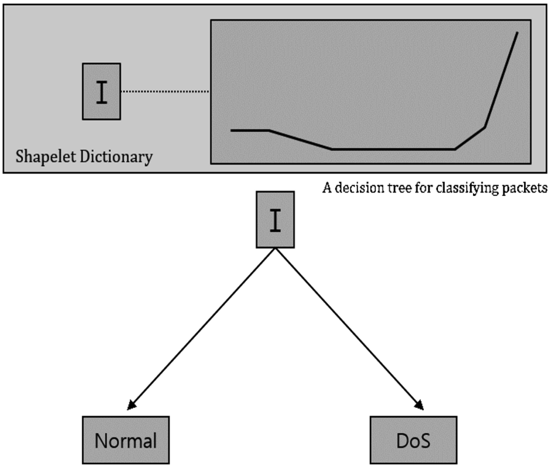


Fig. 3. The shapelets dictionary and decision tree for classifying packets: the representative subsequence shapelets is extracted from the learned normal data to classify the two classes (Normal, and traffic flooding packets, *i.e.*, DoS).

Table 3. Classification accuracy of SVM and proposed methods.

Rating scale	Result (%)	
	SVM	Proposed methods
Precision	98.4	96
Recall	95.6	97
Accuracy	96.3	95

Table 4. Computational time according to number of features.

	# of features	Learning time (sec.)
Non-feature selection	41	428
Feature selection	10	17

5 Conclusions

The intrusion detection system is an important technique that sets the criteria for reliability and validity for hosts that support the network services. In this paper, we proposed a shapelets technique for detecting abnormal traffic based on traffic flooding attack and confirmed that the classification accuracy was about 95%. Also, we confirmed that there is a 25 times improvement in the performance time by reducing the number of features with information gain technique. In the future works, we will conduct research on real-time attack detection by classifying each attack technique and reducing the execution time.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2015R1C1A1A02037688).

References

1. Chung, Y., Lee, S., Jeon, T., Park, D.: Fast video encryption using the H.264 error propagation property for smart mobile devices. Sensors **15**(4), 7953–7968 (2015)
2. Lee, S., Jeong, T.: Forecasting purpose data analysis and methodology comparison of neural model perspective. Symmetry **9**(7), 108 (2017)
3. Lee, S., Kim, H., Chung, Y., Park, D.: Energy efficient image/video data transmission on commercial multi-core processors. Sensors **12**(11), 14647–14670 (2012)
4. Lee, S., Kim, H., Sa, J., Park, B., Chung, Y.: Real-time processing for intelligent-surveillance applications. IEICE Electr. Express **14**(8), 20170227 (2017)
5. Lee, S., Jeong, T.: Cloud-based parameter-driven statistical services and resource allocation in a heterogeneous platform on enterprise environment. Symmetry **8**(10), 103 (2016)
6. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst. Appl. **29**(4), 713–722 (2005)
7. Moser, A., Kruegel, C., Kirda, E.: Limits of static analysis for malware detection. In: 23rd Computer Security Applications Conference, ACSAC 2007, pp. 421–430. IEEE, Miami Beach (2007)
8. Siddiqui, M., Wang, M.C., Lee, J.: A survey of data mining techniques for malware detection using file features. In: 46th Conference Proceedings on xx, pp. 509–510. ACM, Alabama (2008)
9. Tahan, G., Rokach, L., Shahar, Y.: Mal-ID: Automatic malware detection using common segment analysis and meta-features. J. Mach. Learn. Res. **13**, 949–979 (2012)
10. Wojnowicz, M., Chisholm, G., Wolff, M., Zhao, X.: Wavelet decomposition of software entropy reveals symptoms of malicious code. J. Innovation Digit. Ecosyst. **3**(2), 130–140 (2016)
11. Bilar, D.: Opcodes as predictor for malware. Int. J. Electr. Secur. Digit. Forensics **1**(2), 156–168 (2007)
12. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 1, pp. 337–387. Springer, New York (2001). https://doi.org/10.1007/978-0-387-21606-5
13. Alazab, M., Venkatraman, S., Watters, P., Alazab, M.: Zero-day malware detection based on supervised learning algorithms of API call signatures. In: 9th International Conference Proceedings on Australasian Data Mining, vol. 121, pp. 171–182. Australian Computer Society, Ballarat (2011)
14. Davis, A., Wolff, M.: Deep Learning on Disassembly Data. In: Black Hat, USA (2015)
15. Kolter, J.Z., Maloof, M.A.: Learning to detect malicious executables in the wild. In: 10th ACM SIGKDD International Conference Proceedings on Knowledge Discovery and Data Mining, pp. 470–478. ACM (2004)
16. Lyda, R., Hamrock, J.: Using entropy analysis to find encrypted and packed malware. IEEE Secur. Priv. **5**(2), 40–45 (2007)
17. Schultz, M.G., Eskin, E., Zadok, F., Stolfo, S.J.: Data mining methods for detection of new malicious executables. In: Conference Proceedings on Security and Privacy, 2001 IEEE Symposium, pp. 38–49. IEEE, Oakland (2001)

18. Shabtai, A., Moskovitch, R., Elovici, Y., Glezer, C.: Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey. Elsevier **14**(1), 16–29 (2009)
19. Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq, M.: PE-Miner: mining structural information to detect malicious executables in realtime. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 121–141. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04342-0_7
20. Baysa, D., Low, R.M., Stamp, M.: Structural entropy and metamorphic malware. J. Comput. Virol. Hacking Tech. **9**(4), 179–192 (2013)
21. Sorokin, I.: Comparing files using structural entropy. J. Comput. Virol. **7**(4), 259 (2011)
22. Wojnowicz, M., Chisholm, G., Wolff, M.: Suspiciously structured entropy: wavelet decomposition of software entropy reveals symptoms of malware in the energy spectrum. In: International Conference Proceedings on FLAIRS, pp. 294–298 (2016)
23. O'Kane, P., Sezer, S., McLaughlin, K.: Obfuscation: the hidden malware. IEEE Secur. Priv. **9**(5), 41–47 (2011)
24. Keogh, E., Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowl. Inf. Syst. **8**(2), 154–177 (2005)
25. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: International Conference Proceedings on Discovery Data Mining, vol. 10, pp. 359–370 (1994)
26. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
27. Wang, X., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. Data. Min. Knowl. Discov. **13**(3), 335–364 (2006)
28. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Proceedings on Berkeley symposium, vol. 1(14), pp. 281–297 (1967)
29. Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. J. Intell. Inf. Syst. **39**(2), 287–315 (2012)
30. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57301-1_5
31. Lin, J., Vlachos, M., Keogh, E., Gunopulos, D.: Iterative incremental clustering of time series. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 106–122. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24741-8_8
32. Korn, F., Jagadish, H.V., Faloutsos, C.: Efficiently supporting ad hoc queries in large datasets of time sequences. In: International Conference Proceeding on Management of data, vol. 26(2), pp. 289–300. ACM, Tucson (1997)
33. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: International Conference Proceeding on Management of data, vol. 30(2), pp. 151–162. ACM, Santa Barbara (2001)
34. Chan, K.P., Fu, A.W.C.: Efficient time series matching by wavelets. In: 15th International Conference Proceedings on Data Engineering, pp. 126–133. IEEE, Sydney (1999)
35. Popivanov, I., Miller, R.J.: Similarity search over time-series data using wavelets. In: 18th International Conference Proceeding on Data Engineering, pp. 212–221. IEEE, San Jose (2002)
36. Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: Proceedings Workshop on Clustering High Dimensionality Data and its Applications, pp. 23–30 (2003)

37. Antoniadis, A., Brossat, X., Cugliari, J., Poggi, J.M.: Clustering functional data using wavelets. Int. J. Wavelets **11**(1), 1350003 (2013)
38. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data. Min. Knowl. Discov. **28**(4), 851–881 (2014)
39. Zakaria, J., Mueen, A., Keogh, E.: Clustering time series using unsupervised-shapelets. In: 12th International Conference Proceedings on Data Mining (ICDM), pp. 785–794. IEEE, Brussels (2012)
40. Zakaria, J., Mueen, A., Keogh, E., Young, N.: Accelerating the discovery of unsupervised-shapelets. Data. Min. Knowl. Discov. **30**(1), 243–281 (2016)
41. Patri, O., Wojnowicz, M., and Wolff, M.: Discovering malware with time series shapelets. In: 50th International Conference Proceedings on System Science, Hawaii (2017)
42. Castro-Hernandez, D., Paranjape, R.: Classification of user trajectories in LTE HetNets using unsupervised shapelets and multiresolution wavelet decomposition. IEEE Trans. Veh. Technol. **66**(9), 7934–7946 (2017)
43. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 1999 data set. In: Computational Intelligence for Security and Defense Applications, CISDA 2009, pp. 1–6. IEEE, Ottawa (2009)
44. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: 15th ACM SIGKDD International Conference Proceedings on Knowledge discovery and data mining, pp. 947–956. ACM, Paris (2009)
45. Rakthanmanon, T., Keogh, E.: Fast shapelets: a scalable algorithm for discovering time series shapelets. In: International Conference Proceedings on Data Mining, pp. 668–676. Society for Industrial and Applied Mathematics (2013)
46. Gao, Y., Feng, Y., Tan, J.: Exploratory study on cognitive information gain modeling and optimization of personalized recommendations for knowledge reuse. J. Manuf. Syst. **43**, 400–408 (2017)