Hindawi
Complexity
Volume 2019, Article ID 5826737, 9 pages
https://doi.org/10.1155/2019/5826737

Research Article

Practical Employment of Granular Computing to Complex Application Layer Cyberattack Detection

Rafał Kozik,[1] Marek Pawlicki,[1] Michał Choraś,[1] and Witold Pedrycz[2]

[1] *Institute of Telecommunications and Computer Science, UTP University of Science and Technology, Bydgoszcz, Poland*
[2] *Department Electrical and Computer Engineering, University of Alberta, Canada*

Correspondence should be addressed to Rafał Kozik; rkozik@utp.edu.pl

Received 12 October 2018; Revised 28 December 2018; Accepted 6 January 2019; Published 16 January 2019

Guest Editor: Fernando Sánchez Lasheras

Network and information security are regarded as some of the most pressing problems of contemporary economy, affecting both individual citizens and entire societies, making them a highlight for homeland security. Innovative approaches to handle this challenge are undertaken by the scientific community, proposing the utilization of the emerging, advanced machine learning methods. This very paper puts forward a novel approach to the detection of cyberattacks taking inventory of the practical application of information granules. The feasibility of utilizing Granular Computing (GC) as a solution to the most current challenges in cybersecurity is researched. To the best of our knowledge, granular computing has not yet been widely examined or used for cybersecurity application purposes. The major contribution of this work is a method for constructing information granules from network data. We then report promising results on a benchmark dataset.

1. Introduction and Rationale

In the lifecycle of any technological advancement, the application domain and the target user base grow, shift and alter. However, the very technological advancements that contribute to the users' well-being can be twisted into submission by malevolent agents. Apart from the obvious uses, which are beneficial to the society, they can also be both the origin and the objective of a cybercrime.

It does not require exhaustive research to notice that the number and the severity of attacks targeted at the application layer is on the rise.

This situation is echoed by the rankings of cyberattacks, e.g., by OWASP [1], where SQLIA (SQL Injection Attack) and XSS (Cross Site Scripting) are on top of the list. Selected examples of critical systems which were successfully attacked through the application layer are the flight management system in Poland or the energy grid in Ukraine [2]. Therefore, we aim at proposing novel methods to effectively detect and counter the cyberattacks in the application layer. The research we are conducting aims at investigating the ability of the emerging methods of machine learning to counter these attacks. Several attempts have been made to utilize innovative machine learning solutions to tackle the cyberattack problem. Those solutions include lifelong learning intelligent systems or deep learning [3, 4].

We hereby offer, in this paper, the implementation in the domain of cybersecurity of yet another emerging concept, namely Granular Computing.

Therefore, the major contribution of this paper is the application of a granular computing paradigm for cybersecurity.

We propose to extract information granules (which can also be known as tokens) in order to better understand the structure, semantics, and meaning of HTTP requests. Such an approach allows for effective anomaly detection, as is proven by the results we report.

In particular, the main contributions of this paper are

(i) an innovative anomaly detection algorithm based on extracted information granules;

(ii) a description of packet structure;

(iii) an efficient method for request sequence encoding and classification; and

(iv) evaluation of the proposed approach.

The remainder of the paper is structured as follows: Section 2 is focused on current trends and challenges related to cybersecurity and anomaly detection approaches, Section 3 discusses the application of granular computing in cybersecurity, and Section 4 presents an approach for extracting information granules from network data, while Section 5 presents the results of the conducted experiments. Conclusions are given thereafter.

2. Cybersecurity: A Quick Primer on Challenges and Trends

The rising numbers of attacks aimed at citizens, societies and even seemingly secure systems built for critical infrastructures are easily observable [2, 5]. The inadequacy of signature-based systems in cyberattack detection is one of the primary causes of this situation. Whenever new attacks are created, or even slightly modified families of malware are utilized, those systems, which constitute an industry standard, fall short of properly handling the hazard until new signatures are added. Anomaly detectors, which could tackle either new, or obfuscated attacks, are likely to generate false positives. There is a plethora of solutions that aim at countering attacks targeting the application layer. Many of those products use the signature-based approach. The signature-based category of cyber-attack detection includes Intrusion Prevention and Detection Systems (IDS and IPS) which use a predefined set of patterns (called signatures) to identify an attack. IPS and IDS are deployed to improve the security of computer systems and networks through detection (in the case of IDS) and blocking (in the case of IPS) of the cyberattacks.

Another class of solutions is called WAF (Web Application Firewall [6–8]). Those solutions are based on black and white listing approach. Classification of requests sent from client to server is performed. WAFs work on the basis of regular expressions, patterns, and signatures to detect cyberattacks. The predefined patterns (or rules) are typically compared to the content of incoming requests (either the header or the payload). A very popular IDS/IPS open product is called SNORT [9]. It is an open source project with a community of users who can freely modify the software, providing new rules to the Snort engine.

Injection attacks such as SQL or XSS occur when an improperly validated request containing malicious code is sent to an interpreter as part of a command or query. XSS flaws might occur when the application processes malicious data and sends it to a web browser without proper validation or escaping (the reformatting of ambiguous characters). By its nature, XSS allows attackers to execute scripts in the victim's browser. This leads to hijacking user sessions, defacing web sites, or redirecting the user to malicious sites.

The problem of developing effective signatures for such attacks is a highly complex one, as these attack vectors and patterns lend themselves to obfuscation. Another drawback of many WAF solutions is the fact that those are based on the preliminary (or previously learnt) assumptions regarding the request's structure (e.g., [6, 7]). However, different protocols utilizing HTTP as the transportation protocol are characterized by different payload structures. For example, the structure sent via a plain HTML form is different from that of a GWT-RPC or a SOAP call. In such cases a number of pre-prepared signatures will not match a differently-structured payload, and in consequence, those attacks will not effectively be detected.

Another approach to HTTP traffic anomaly detection was presented in [10]. The authors applied a DFA (Deterministic Finite Automaton) to compare the requests described by means of tokens. The method based on the *n-grams* applied for anomaly detection and cyber-attacks detection has been presented in [11]. Other systems have also implemented algorithms to analyze the n-grams, for example with the use of statistical analysis [12], Self-Organizing Maps [13], Bloom filters [14], and a wide variety of different machine learning classifiers [15]. However, depending on the analyzed protocols and the methodology used to analyze n-grams, researchers report very diverse results for n-grams techniques. For instance, in [11] authors reported a high number of false positives for various state of the art methods. In contrast, authors in [16] reported a recognition rate of over 85%, while having quite low false positive rate of only 1%.

3. Information Granules in Cybersecurity

One of the most serious challenges of the methods and algorithms used in cybersecurity is being able to reach a correct understanding of the network data. Undeniably, cyberecosystems are quickly changing, as are the characteristics of the data. This fluctuation of properties produces uncertainty and difficulties in data partitioning/clustering. It is profoundly challenging to construct correct generalizations, rules and thresholds, and substandard choices greatly decrease the efficiency of typical pattern recognition and anomaly detection algorithms. In addition, many of the used pattern recognition techniques do not try to incorporate or even take into account the semantics of the analyzed network data.

We propose the utilization of the practical elements of Granular Computing for anomaly detection as a solution of the preceding problems.

Granular computing refers to a general data analysis and recognition framework, incorporating data partitioning into so called information granules.

Granular Computing emerged as a general structure of data processing and knowledge discovery utilizing items called information granules. The very concept of granulation appeared independently in an array of fields, including fuzzy and rough sets or cluster analysis [17]. Granules are alignments of elements drawn together by their similarity, closeness or functionality [18]. A granule which occurs at a particular granularity level conveys a certain aspect of the modelled issue [19]. In situations with a certain degree of uncertainty granules can provide a convenient solution. This property can be translated into a certain economy when dealing with intricate problems. The tolerance for uncertainty bears a degree of resemblance to human thinking itself [18]. What follows is the utilization of Granular Computing in designing real-life smart systems. The hierarchical nature of Granular Computing in conjunction with the basics of human reasoning conveyed in its premise makes it a perfect match for meaningful abstraction on various levels of detail [20].

Granules are essentially tiny parts of a larger construct, which describe a particular facet of that construct, when viewed from a particular level of granulation [21]. As an illustration of this principle one can consider how in cluster analysis objects can be grouped together based on similarity or distance functions. Since objects grouped in one cluster should exemplify a strong degree of similarity, clusters can be considered as granules [20]. Granules can be, thus, amassed into larger collections, which are then perceived as new, larger granules or divided into smaller pieces, which are more specific [21].

Ideally, the extracted information granules should comply with the Principle of Justifiable Granularity (PJG) [22]. PJG is a guideline for information granules to best comply with two competing requirements: justifiability and specificity. It stipulates that the constructed granules cover the relevant portion of the data, but should not be highly dispersed across the dataset. This can be achieved by selecting granules that resemble relevant semantics describing the data. Typical practical methods of granular computing are fuzzy sets [23], rough sets [24–26], and intuitionistic sets [27].

Granular computing allows for better data understanding through the incorporation of semantic aspects, similarities and uncertainties. Granular computing has been used recently for the analysis of spatiotemporal data [28], to concept-cognitive learning from large and multi-source data in formal concept analysis [29].

Granular Computing has been used to estimate a power plant's electrical power output [30]. The principles of granular computing are utilized to cluster the data with regard to the distance between granules, and the density of the newly constructed granules. Data prepared this way is fed into an Adaptive Neuro-Fuzzy Inference System (ANFIS). The procedure has been tested and proven to be a close fit [30].

Another application of granular computing allows for the recognition of faces that were surgically altered. Multiple levels of granules are created, some granules contain information about the whole face, some about specific regions, and some about the fine detail of specific features. Those granulation levels are then directed to classifiers [31].

The utilization of granular computing allowed for the creation of more accurate medical classifiers, which has an appreciable value in medical procedures. Data is granulated on multiple levels with regards to the Euclidean distance of the data points to the centroids. This distance constitutes the level of granulation, with higher granulation achieved through enlarging the distance. Data on multiple levels is then served to classifiers, which return a value between 0 and 1. The values are then introduced to a final stage classifier [32].

One of the recent research papers proposed a system introducing granular computing to financial markets. The proposed method of time series forecasting bested the state-of -the-art benchmark algorithms. Clustering is performed with an adaptation of the possibilistic fuzzy c-means algorithm, supplemented with the ability to process intervals. The algorithm recursively gauges cluster centers as it brings in novel data [33].

Granular computing can be used to increase the calculating speed when solving the economic dispatch problem in order to reach the minimal running cost of a power grid. The described method breaks down a large power grid into smaller components which are treated as equivalent power stations. The process can be repeated, obtaining a finer level of granulation with each iteration. Determining optimal values on each level, before trying to reach a global optimum, makes the procedure both quicker and easier [34].

We apply our own methods to construct a practical solution based on information granules obtained from the network data.

To the best of our knowledge, granular computing has not yet been widely examined nor adapted for cybersecurity application purposes. One of the rare published papers is authored by Napoles et al. [35]. The authors addressed the problem of modeling and classification for network intrusion detection by utilizing a recently proposed granular model named Rough Cognitive Networks (RCN). The authors both proposed and defined RCN for detection of atypical (abnormal and potentially dangerous) patterns in the network traffic. RCN has been delineated as a sigmoid FCM (Fuzzy Cognitive Map). Map concepts denote information granules corresponding to the RST (Rough Set Theory) -based positive, boundary and negative regions of decision classes. Learning mechanisms for RCN are based on a self-adaptive Harmony Search algorithm. The proposed model has been evaluated with the NSL-KDD dataset (https://github.com/defcom17/NSL_KDD) and is shown to be a suitable and promising approach for detecting abnormal traffic in computer networks. Future work will address validation and further evaluation of the model based on real network traffic.

In the upcoming section we present our innovative methods for extracting information granules to counter cyber-attacks in the application layer.

4. Proposed Approach for Extracting Information Granules from Network Data in Cybersecurity

Hereby we propose a new method for the clustering of multiple HTTP sequences utilizing a machine-learning classifier and granular computing approach.

As a way of grasping the semantics and the granularity, we use the information about the request structure and the statistical measurements of the structure content to detect anomalous behavior of untrusted sessions between client and server.

An overview of the proposed algorithm is presented in Figure 1, while a general overview of the granulation procedure utilized in our approach is presented in Figure 2.

As can be seen later in Section 5, the conducted experiments confirm the promising results and we can report that the proposed approach and method is competitive with other state-of-the-art solutions.

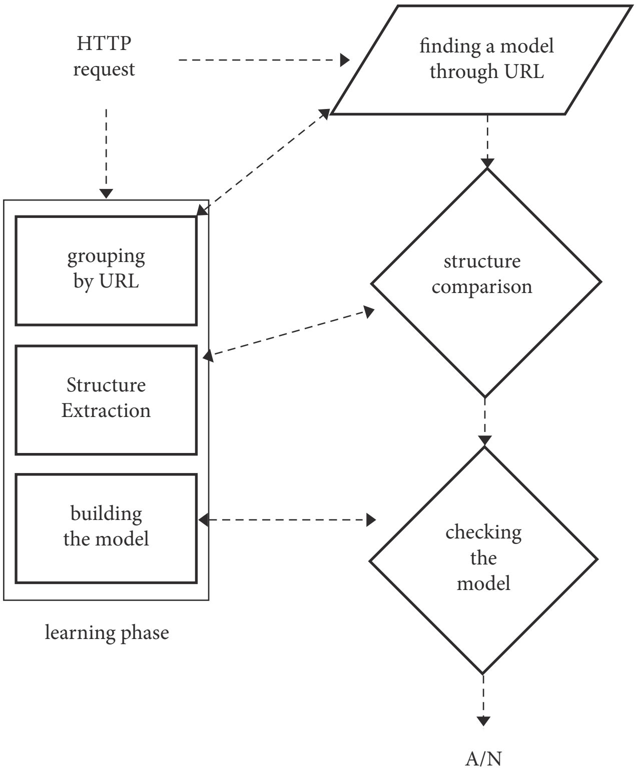

Figure 1: High-level overview of the proposed algorithm.

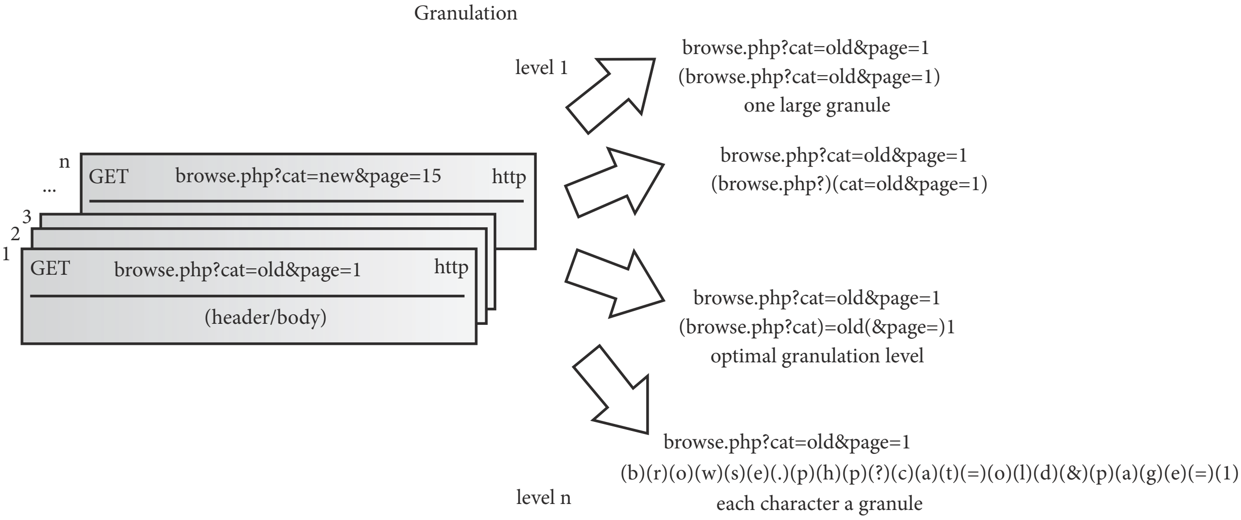
Granulation
level 1
browse.php?cat=old&page=1
(browse.php?cat=old&page=1)
one large granule
browse.php?cat=old&page=1
(browse.php?)(cat=old&page=1)
browse.php?cat=old&page=1
(browse.php?cat)=old(&page=)1
optimal granulation level
browse.php?cat=old&page=1
(b)(r)(o)(w)(s)(e)(.)(p)(h)(p)(?)(c)(a)(t)(=)(o)(l)(d)(&)(p)(a)(g)(e)(=)(1)
each character a granule
level n
n
...
3
2
1
GET
browse.php?cat=new&page=15
http
GET
browse.php?cat=old&page=1
http
(header/body)

Figure 2: The overview of the granulation algorithm. The HTTP requests undergo a textual semantic segmentation of similar requests. The optimal level of granulation reveals data that can be used to calculate the feature vector.

The crucial advantage of the method proposed in this paper is that it is invariant to the underlying protocol stack (the method is protocol agnostic). In other words, it does not need to be tuned to any of the used protocols or application interfaces using HTTP for transport (e.g., the RESTful API, and SOAP). Hypertext Transfer Protocol (HTTP) is now frequently used due to its simplicity and reliability in assuring communication between computers in distributed networks and allowing for increased usage of the web applications.

In our method, we apply a granular computing approach and extract information granules from HTTP requests.

An information granule in this approach is defined as a recurring sequence of information, which shares semantics for all the requests sent to the same resource or server (in Figure 2).

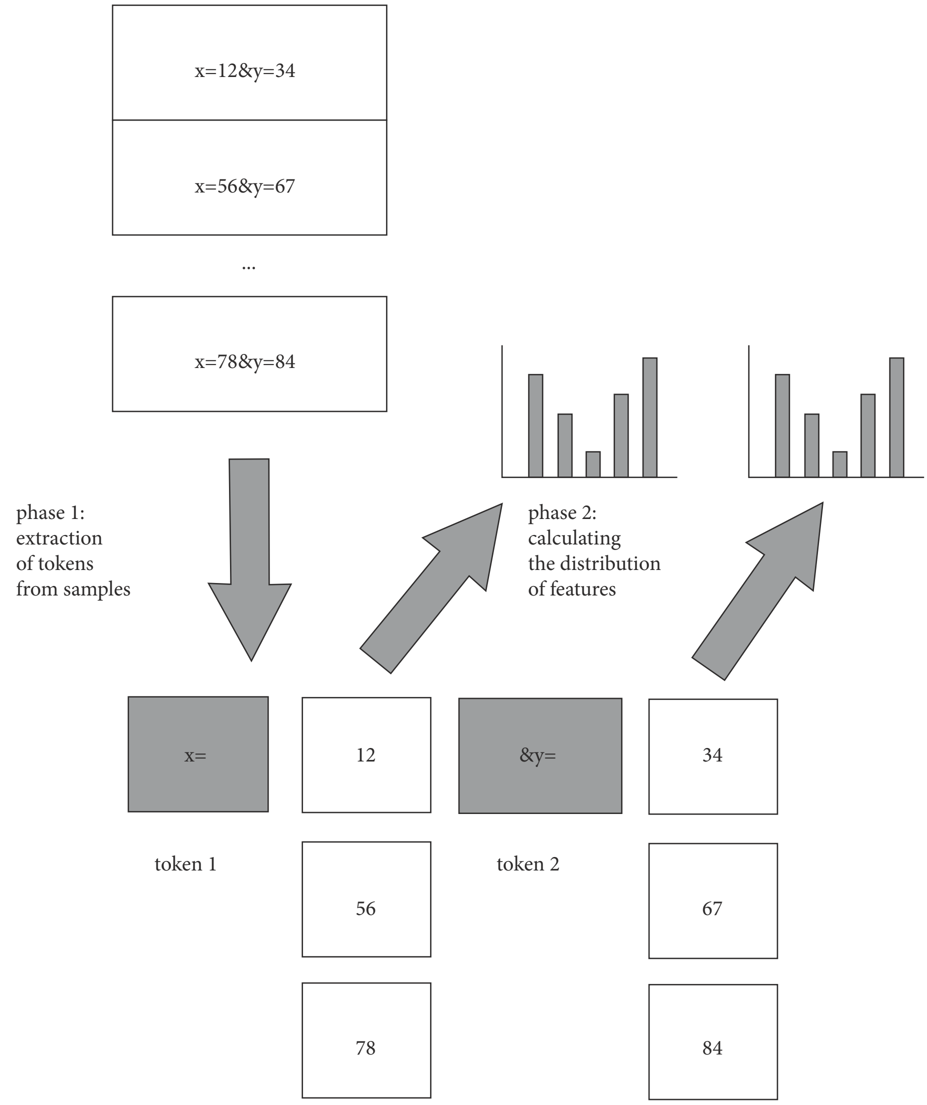

Figure 3: The parameter encoding approach.

The analysis of each single request can result in several granules (tokens). Our goal is to extract granules that will identify in the analyzed requests the delimiters of regions representing positions injected with cyberattacks or malware.

After these tokens are found, we have to describe the sequences between them by calculating their statistical properties (these tokens are represented on the optimal granulation level in Figure 2). We propose a two-step segmentation of HTTP requests. In the first step we divide the request space into smaller subsets to perform further calculations in the parallelized manner. In the second step we perform the actual segmentation of requests in order to identify their structure. The structure is described and represented by the extracted information granules. Those information granules (shaded boxes in Figure 3) allow us to identify the clusters in the feature space consisting of the feature vectors extracted from the granularized data.

Afterwards, when feature vectors are assigned to the appropriate clusters, we apply a machine-learning classifier. The classification task is to assign them either to the normal or anomalous class. In our experiments we have used various supervised methods. These have been explained in the experiments section.

In order to extract the granules we implemented the known LZW compression method (Lempel-Ziv-Welch [36, 37]). Firstly, we create the LZW dictionary D which allows us to transform the textual input (e.g., logs) into natural numbers (see the following):

$$D : word \longrightarrow \{i : i€N\} \qquad (1)$$

The algorithm performs scanning within the set S in order to find the successively longer subsequences. This step is performed until the algorithm finds a sequence that does not yet belong to the dictionary. The found substring is added to the dictionary unless it is already represented. The described

```
Data: Set of HTTP payloads $S$
Result: Dictionary $D$
$s$ = empty string
while there is still data to be read in S do
    $ch \leftarrow$ read a character
    if $(s + ch) \in D$ then
        $s \leftarrow s+ch$;
    else
        $D \leftarrow D \cup (s+ch)$;
        $s \leftarrow ch$;
    end
end
```

FIGURE 4: The algorithm for creating the dictionary D.

procedure repeats until the entire dataset is scanned. The described algorithm is shown in Figure 4.

At the end of the processing a set S of HTTP payloads, the dictionary D, containing a list of sequences is created.

This dictionary has the form of an unordered list. Positions in the list can be taken only by one word. The dictionary D is implemented as a hash-table to achieve *O(1)* lookup time.

Of course, as in LZW method, creating the dictionary D, also allows for compressing the data. The algorithm replaces words by numbers corresponding to the position of the sequence in the dictionary.

It is worth noticing that even after the single scan of the data, we can extract a reasonable number of candidates for appropriate information granules. Of course, it is not a trivial task, given the need to achieve balance between the specificity and justifiability. Another advantage is the compression of the data.

Still we have to further process the dictionary in order to obtain the collection of information granules. First condition is that we remove all the candidates that do not appear in all the samples used for structure extraction. In the next step we remove the sequences that also appear elsewhere as the subsequences of others.

The HTTP requests have the form of character sequences, whose lengths vary.

Moreover, single granule can appear at different positions in consecutive HTTP requests. Also the distance between granules may vary and, additionally, it happens that sometimes one granule is a subset of another information granule.

In our approach, we propose to use IDC (Idealized Character Distribution) method. We calculate it in the training phase from normal requests sent to a web application (the assumption is that the requests are normal, and manual inspection is needed in this step). The IDC is calculated as the mean value of all the character distributions. During the detection phase, we calculate and evaluate the probability that the character distribution of a sequence is an actual sample drawn from its ICD. Hereby, we use the well-known Chi-Square metric.

Equation (2) is used for computing the value of the Chi-Square metric $D_{chisq}(Q)$ for a sequence Q:

$$D_{chisq}(\mathrm{Q}) = \sum_{n=0}^{N} \frac{\left(ICD_n - h_n(\mathrm{Q})\right)^2}{ICD_n} \tag{2}$$

where N indicates the number of bins in the *histogram(in our approach we used N=9)*, *ICD* the distribution established for all the samples, and *h()* the distribution of the sequence Q that is being tested.

In order to calculate the distributions we count the number of characters that fall into each of the range of the ASCII table. We use the following ranges for this distribution count: <0,31>, <32,47>, <48,57>, <58,64>, <65,90>, <91,96>, <97,122>, <123,127>, and <128,25>. The chosen ASCII ranges represent different types of signs such as numbers, quotes, letters, or special characters and in result represent well the distribution. The histogram that is used here will have 9 bins (due to 9 ranges).

5. Experiments

The CSIC10 benchmark dataset [38] was used for the experiments. It contains several thousand HTTP protocol requests which are organized in a form similar to the Apache Access Log. The dataset was developed at the Information Security Institute of CSIC (Spanish Research National Council) and it contains the generated traffic targeted to an e-Commerce web application. For convenience, the data was split into anomalous, training, and normal sets. The dataset contains approx. 36000 normal and 25000 anomalous requests. The anomalous requests are not always cyberattacks. They might refer to some anomalies (e.g., requesting unavailable resource), but more importantly they contain a wide range of application layer attacks, such as SQL injection, buffer overflow, information gathering, files disclosure, CRLF injection, XSS, server side include, and parameter tampering. To understand the results it is important to remember that the requests targeting hidden (or unavailable) resources are also considered anomalies. Some examples of such anomalies are requests for configuration files, default files, or session IDs in a URL (symptoms of an http session takeover attempt). Moreover, the requests with an appropriate format (e.g., a telephone number composed of letters) are also labeled as anomalies. As authors of the dataset explained, such requests may not have a malicious intent but nevertheless they do not follow the normal behavior of the web application. Still, there is no other appropriate, publicly available dataset for the web attack detection problem where we could reliably compare our results.

To verify our method based on information granule extraction we checked how our solution handles different quantities of learning data. As it is the typical case, in our experiments we also used the 10-fold approach.

The 10-fold cross-validation, also known as rotation estimation, is a model validation technique applied for evaluation of a machine learning model effectiveness in generalising the model to an unforseen dataset. The method is utilised in spotting problems like overfitting or selection bias. It provides an overview of how the model might perform. In general, k-fold cross-validation achieves results less biased than other methods based on splitting the dataset into training and testing data subsets (e.g., repeated random subsampling). Cross-validation averages the results of all the

Table 1: True positive rate and false positive rate for different learning algorithms.

Algorithm	True Positive Rate	False Positive Rate
ICD	97,8%	8,1%
DS AdaBoost	93,7%	0,1%
RepTree	93,1%	0,3%
Random Forest	91,9%	0,7%

folds to come up with a more accurate assessment of a model performance.

In our case, the data used in learning and evaluation purposes is divided randomly into 10 parts (folds). One part of the data (10% of the entire dataset) is used for evaluation while the remaining 90% is used for training (e.g., establishing model parameters). The whole procedure is repeated 10 times, so each time a different part of the dataset is used for evaluation and a different part is used for training. The results for all 10 folds are averaged to yield an overall error estimate.

5.1. Comparison of Different Classification Techniques. In this experiment we have compared the effectiveness of various machine learning methods. As it was explained earlier first the granules are extracted and data for each granule is encoded used histograms. These histograms are used to train the algorithms.

It must be noticed that ICD is purely anomaly detection method and it can be trained on normal data. Other algorithms require both normal and anomalous data. As it is shown in Table 1 the ICD algorithm achieves the highest anomaly detection (TPR). However, the number of false alarms in contrast to other methods is relatively high.

5.2. Assessment of Training Dataset Size Impact of Classification Effectiveness. For each fold we deliberately picked only a subset of data to train the classifier. In such approach, we still have the same number of testing samples (common baseline for comparison) even if we have used only a fraction of the available training data. The entire 10-fold cross validation is repeated for different proportions of the training data, namely, 1%, 10%, 20%, and 100%. Results are presented in Table 2.

To obtain a better overview of the effectiveness of our method we calculated and present the ROC curves. The ROC curve for 300 learning samples is presented in Figure 5, while the curve for 32400 samples is presented in Figure 6.

6. Conclusions

In this paper we have proposed using the elegant theory of Granular Computing (GC) as the new approach to cybersecurity and network anomaly detection. The major contribution and innovation of this work is the first practical implementation of the method to extract information granules in order to detect cyberattacks. The proposed solution is designed to work with a typical HTTP-based, request-response web application. It can be described as an anomaly detection tool that receives HTTP requests, analyses their content, extracts information granules, and classifies those either as normal or as anomalies. We conducted the set of experiments on a standard benchmark dataset and typical evaluation scenarios. We report promising results, which demonstrate the efficiency of our approach and motivate our further research in applying Granular Computing to the cybersecurity domain (e.g., for other types of attacks in other layers).

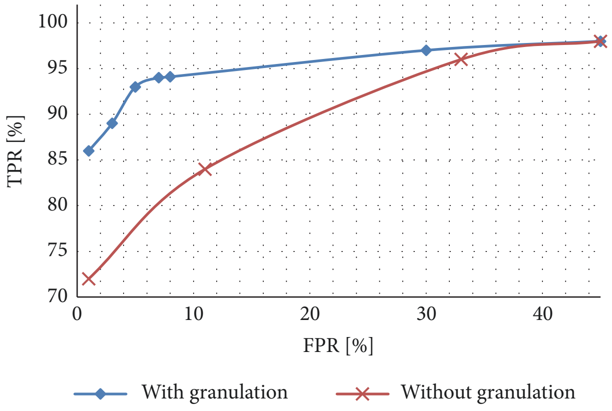

Figure 5: ROC curves for the Chi-Square metric comparing effectiveness of anomaly detection when granules for payload are extracted and otherwise. The experiment was conducted for an algorithm trained on 300 samples.

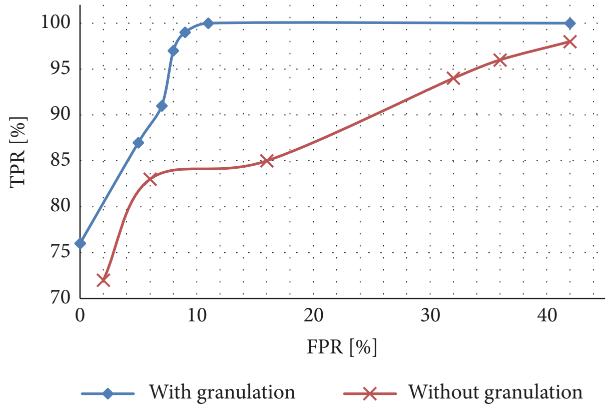

Figure 6: ROC curves for the Chi-Square metric comparing effectiveness of anomaly detection when granules for payload are extracted and otherwise. The experiment conducted for an algorithm trained on 32000 samples.

Data Availability

The data used to support the findings of this study are included within the article.

Table 2: True positive rate and false positive rate for different numbers of learning samples.

TP Rate [%]	FP Rate [%]	Data Set Size	Number of samples
86,6	1.8	1%	300
95,6	5,8	10%	3000
96,9	6,8	20%	6000
97,7	8,1	100%	32400

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] "OWASP Top 10, 2013, OWASP project homepage," 2018.

[2] M. Choraś, R. Kozik, A. Flizikowski, W. Hołubowicz, and R. Renk, "Cyber Threats Impacting Critical Infrastructures," in *Managing the Complexity of Critical Infrastructures*, R. Setola, V. Rosato, E. Kyriakides, and E. Rome, Eds., vol. 90, pp. 139–161, Springer International Publishing, Cham, Switzerland, 2016.

[3] M. Choraś, R. Kozik, R. Renk, and W. Hołubowicz, "The concept of applying lifelong learning paradigm to cybersecurity," *Intelligent Computing Methodologies*, pp. 663–671, 2017.

[4] D. Ariu, I. Corona, R. Tronci, and G. Giacinto, "Machine Learning in Security Applications," *Transactions on Machine Learning and Data Mining*, vol. 8, no. 1, 2015.

[5] R. Kozik, M. Choraś, A. Flizikowski, M. Theocharidou, V. Rosato, and E. Rome, "Advanced services for critical infrastructures protection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 6, pp. 783–795, 2015.

[6] "SCALP, Project homepage," http://code.google.com/p/apache-scalp/, 2018.

[7] "PHPIDS, Project homepage," 2018.

[8] "OWASP Stinger, Project homepage," https://www.owasp.org/index.php/Category:OWASPStingerProject, 2018.

[9] "SNORT, Project homepage," http://www.snort.org/, 2018.

[10] K. L. Ingham, A. Somayaji, J. Burge, and S. Forrest, "Learning DFA representations of HTTP for protecting web applications," *Computer Networks*, vol. 51, no. 5, pp. 1239–1255, 2007.

[11] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, "N-Gram against the Machine: On the Feasibility of the N-Gram Network Analysis for Binary Protocols," in *Research in Attacks, Intrusions, and Defenses*, pp. 354–373, 2012.

[12] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, vol. 3224, pp. 203–222, Springer, Berlin, Germany, 2004.

[13] D. Bolzoni, S. Etalle, P. Hartel, and E. Zambon, "POSEIDON: A 2-tier anomaly-based network intrusion detection system," in *Proceedings of the 4th IEEE International Workshop on Information Assurance, IWIA 2006*, pp. 144–156, UK, April 2006.

[14] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: a content anomaly detector resistant to mimicry attack," in *Recent Advances in Intrusion Detection*, vol. 4219, pp. 226–248, Springer, Berlin, Germany, 2006.

[15] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: a multiple classifier system for accurate payload-based anomaly detection," *Computer Networks*, vol. 53, no. 6, pp. 864–881, 2009.

[16] K. L. Ingham and H. Inoue, "Comparing Anomaly Detection Techniques for HTTP," *Recent Advances in Intrusion Detection*, pp. 42–62, 2007.

[17] T. Y. Lin and C. J. Liau, "Granular computing and rough sets," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., pp. 535–561, Springer, Boston, Mass, USA, 2005.

[18] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 111–127, 1997.

[19] A. Bargiela and W. Pedrycz, "The roots of granular computing," in *Proceedings of the IEEE International Conference on Granular Computing*, pp. 806–809, 2006.

[20] Y. Yao, "A partition model of granular computing," *Transactions on Rough Sets I*, vol. 1, pp. 232–253, 2004.

[21] Y. Y. Yao, "Granular Computing," in *Proceedings of the 4th Chinese National Conference on Rough Sets*, vol. 31, pp. 1–5, 2004.

[22] W. Pedrycz and W. Homenda, "Building the Fundamentals of Granular Computing: A Principle of Justifiable Granularity," *Applied Soft Computing*, vol. 13, no. 10, pp. 4209–4218, 2013.

[23] C. Wagner, S. Miller, J. M. Garibaldi, D. T. Anderson, and T. C. Havens, "From interval-valued data to general type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 2, pp. 248–269, 2015.

[24] F. Gong, M.-W. Shao, and G. Qiu, "Concept granular computing systems and their approximation operators," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 2, pp. 627–640, 2017.

[25] Y. H. Qian, J. Liang, and C. Y. Dang, "Incomplete multigranulation rough set," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 40, no. 2, pp. 420–431, 2010.

[26] M. I. Ali, B. Davvaz, and M. Shabir, "Some properties of generalized rough sets," *Information Sciences*, vol. 224, pp. 170–179, 2013.

[27] B. Huang, Y. Zhuang, and H. Li, "Information granulation and uncertainty measures in interval-valued intuitionistic fuzzy information systems," *European Journal of Operational Research*, vol. 231, no. 1, pp. 162–170, 2013.

[28] M. Song, W. Shang, L. Wang, and W. Pedrycz, "Analysis of spatiotemporal data relationship using information granules," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 5, pp. 1439–1446, 2017.

[29] J. Niu, C. Huang, J. Li, and M. Fan, "Parallel computing techniques for concept-cognitive learning based on granular computing," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 11, pp. 1785–1805, 2018.

[30] W. Sun, J. Zhang, and R. Wang, "Predicting electrical power output by using Granular Computing based Neuro-Fuzzy modeling method," in *Proceedings of the 27th Chinese Control and Decision Conference, CCDC 2015*, pp. 2865–2870, China, May 2015.

[31] K. Vimitha and M. Jayasree, “Recognizing faces from surgically altered face images using granular approach,” in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 463–466, Chennai, India, March 2017.

[32] M. Al-Shammaa and M. F. Abbod, “Granular computing approach for the design of medical data classification systems,” in *Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2015*, pp. 1–7, Canada, August 2015.

[33] L. Maciel, R. Ballini, and F. Gomide, “Evolving granular analytics for interval time series forecasting,” *Granular Computing*, vol. 1, no. 4, pp. 213–224, 2016.

[34] X. Li and L. Fang, “Research on economic dispatch of large power grid based on granular computing,” in *Proceedings of the 2016 IEEE PES Asia Pacific Power and Energy Engineering Conference, APPEEC 2016*, pp. 1130–1133, China, October 2016.

[35] G. Nápoles, I. Grau, R. Falcon, R. Bello, and K. Vanhoof, “A Granular Intrusion Detection System Using Rough Cognitive Networks,” *Recent Advances in Computational Intelligence in Defense and Security*, vol. 621, pp. 169–191, 2016.

[36] T. A. Welch, “A Technique for high-performance data compression,” *The Computer Journal*, vol. 17, no. 6, pp. 8–19, 1984.

[37] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[38] C. Torrano-Gimnez, A. Prez-Villegas, and G. Alvarez, “The HTTP dataset CSIC 2010,” http://users.aber.ac.uk/pds7/csic-dataset/csic2010http.html.