SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **8**:2750–2767
Published online 29 August 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.603

RESEARCH ARTICLE

Combining expert knowledge with automatic feature extraction for reliable web attack detection

Carmen Torrano-Gimenez[1]*, Hai Thanh Nguyen[2], Gonzalo Alvarez[1] and Katrin Franke[2]

[1] Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas, Madrid, Spain
[2] Norwegian Information Security Laboratory, Gjøvik University College, Gjøvik, Norway

ABSTRACT

In the detection of web attacks, it is necessary that Web Application Firewalls (WAFs) are effective, at the same time than efficient. In this paper, we propose a new methodology for web attack detection that enhances these two aspects of WAFs. It involves both feature construction and feature selection. For the feature construction phase, many professionals rely on their expert knowledge to define a set of important features, what normally leads to high and reliable attack detection rates. Nevertheless, it is a manual process and not quickly adaptive to the changing network environments. Alternatively, automatic feature construction methods (such as *n*-grams) overcome this drawback, but they provide unreliable results. Therefore, in this paper, we propose to combine expert knowledge with *n*-gram feature construction method for reliable and efficient web attack detection. However, the number of *n*-grams grows exponentially with *n*, which usually leads to high dimensionality problems. Hence, we propose to apply feature selection to reduce the number of redundant and irrelevant features. In particular, we study the recently proposed Generic Feature Selection (GeFS) measure, which has been successfully tested in intrusion detection systems. Additionally, we use several decision tree algorithms as classifiers of WAFs. The experiments are conducted on the publicly available ECML/PKDD 2007 dataset. The results show that the combination of expert knowledge and *n*-grams outperforms each separate technique and that the GeFS measure can greatly reduce the number of features, thus enhancing both the effectiveness and efficiency of WAFs. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS
Web attack detection; web application firewall; intrusion detection systems; machine learning algorithms; feature selection; *n*-grams

***Correspondence**
Carmen Torrano-Gimenez, Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas, Madrid.
E-mail: carmen.torrano@iec.csic.es

1. INTRODUCTION

Web applications are becoming increasingly popular as part of our daily life. They include for example online shopping, bank operations, people communication, embedded devices configuration, and so on. As a consequence, web applications are very attractive for attackers, and they are exposed to a wide range of attacks, such as Cross-Site Scripting (XSS), SQL injection, or web defacement, which might have dramatic consequences such as impersonation, revelation of private data stored in the database, or modifications in the web page. Furthermore, new threats appear every day, hence, it is necessary to adopt adequate security mechanisms to protect web applications.

Unfortunately, conventional firewalls, that operate at network and transport layers, are usually not enough to detect web-specific attacks. To be effective, the detection has to take place at the application layer. Web application firewalls (WAFs) [1] are systems that work at the application layer and can help to detect web attacks. In fact, these systems analyze the HTTP traffic in order to detect malicious actions and behavior that might compromise the security of web applications. When designing WAFs, it is important to consider both effectiveness (the capacity of detecting attacks while not raising false alarms) and efficiency (capacity of consuming low resources and achieving low computational complexity). Efficiency is critical, for WAFs operating in real-time environments for instance or in scenarios with resource constraints.

In the aim of web attack detection, it is usual to define a set of features considered important to distinguish between web attacks and normal traffic. This can be carried out using different techniques. On the one hand, many professionals rely on their expert knowledge to define this set of features. This approach often leads to high and reliable attack detection rates; however, it is a manual process and not quickly adaptive to the changing network environments. On the other hand, automatic methods, such as *n*-grams, overcome this drawback, nevertheless they provide unreliable results. Therefore, in this paper, we propose to combine expert knowledge with *n*-gram feature construction method for reliable and efficient web attack detection. Our hypothesis is that the combination would improve the results of expert knowledge and *n*-grams separately. Because *n*-grams are subsequences of *n* items from a given sequence [2,3], the number of *n*-grams increases exponentially with the value of *n*, which usually leads to the so called "curse of dimensionality" and computational complexity problem. In order to solve this problem, we propose to apply feature selection, which reduces the number of redundant and irrelevant features while not negatively affecting the detection accuracy. In particular, we use the recently proposed generic feature selection (GeFS) measure [4], which has been successfully tested in network intrusion detection system [4,5] and WAFs [6,7].

For the classification phase, several decision tree algorithms are used as classifiers of WAFs. This family of algorithms is one of the most popular [8] and experimentally successful in machine learning. Additionally, decision trees are widely used in intrusion detection; in fact, the winner of the famous DARPA intrusion detection contest [9] was an algorithm based on decision trees [10]. Due to these reasons, decision trees are chosen in this paper for the classification phase.

The experiments are conducted on the publicly available ECML/PKDD 2007 dataset [11], that contains labeled HTTP traffic.

The experimental results show that indeed, the combination provides better results than expert knowledge and *n*-grams separately, then confirming the hypothesis. Additionally, it is also shown that the GeFS measure can greatly reduce the number of features. Therefore, our methodology can enhance both the effectiveness and the efficiency of WAFs.

In summary, the paper presents the following contributions:

- We propose a methodology for the detection for web attacks designed to improve both the effectiveness and the efficiency of WAFs. For the construction of features, we propose to combine expert knowledge with *n*-gram feature construction method. And for the dimensionality problems, we propose to apply feature selection, in particular the GeFS measure.
- We apply the methodology to the ECML/PKDD 2007 dataset. The experimental results confirm our hypothesis and the success of our methodology.

The paper is organized as follows. Section 2 introduces the background concepts of the different techniques and methods used in this paper. Section 3 explains in detail all the phases of the methodology proposed. Then, Section 4 describes the particularities of the ECML/PKDD 2007 dataset, used to conduct the testing experiments of our methodology. The experimental settings are provided in Section 5, and the results are shown in Section 6. Finally, Section 7 presents the conclusions of this work.

2. BACKGROUND

2.1. Expert knowledge feature construction

The main goal of feature construction is to obtain the appropriate features that represent the regularities of the original dataset [12]. Choosing representative features of a dataset is crucial for the success of the classification algorithms [13].

In web attack detection, many authors apply their expert knowledge to determine the important features that will help in the detection of web attacks. For example, this is the case of the multi-model system of Kruegel *et al.* [14] that detects web attacks in HTTP queries that contain parameters. In this work, the features considered relevant for detecting malicious activity are the following: attribute length, attribute character distribution, structure of the parameters (regularity of the non-printable characters), detection of anomalous values for an attribute (different values for a fixed-value attribute), attribute presence or absence, attribute order, access frequency, inter-request time delay, and invocation order of the component programs of the web-based application. To evaluate the previous features, the system creates different models that are firstly trained to learn the normal values. Then, based on the model outputs, the test queries are reported as either a potential attack or as normal.

Other WAFs proposed by Torrano *et al.* [15,16] also apply expert knowledge for web attack detection. Like the system by Kruegel *et al.*, these WAFs follow the anomaly-approach, i.e. the normal behavior of the web application is defined and any action that deviates from that is tagged as intrusive. These WAFs divide the HTTP requests into tokens, therefore they work at token level and not at request level. These systems utilize two kinds of features to differentiate normal and anomalous traffic: features related to the length of the tokens and features related to the structure of the tokens of the HTTP request. The features concerning the structure count the number of letters, digits, and the rest of printable ASCII characters present in the tokens of the requests. The normal values of the features are learnt from the training web traffic and, then, the incoming requests are classified according to the models and the correct values learnt.

2.2. *N*-gram feature construction

N-gram-based automatic methods have been successfully used in intrusion detection for feature construction [17–19]. Models based on *n*-grams originate from the fields of information retrieval and natural language processing. An *n*-gram is a "language-independent" statistical model [17]. In the case of intrusion detection, if a payload is considered a string, then an *n*-gram is a substring of *n* characters. When working with byte sequences, the space S of all possible *n*-grams ($n \geq 1$) has the size of 2^{8n} (considering 8-bit representation for each character):

$$S = \left\{ n - \text{grams}_i \middle| i = 1, \ldots, 2^{8n} \right\}$$

Therefore, the number of *n*-grams increases exponentially with *n*. For example, for $n = 2$, the number of features is already 65 536. The lower the number of features, the more efficient the detection system is; thus, it is necessary to look for solutions to deal with the "curse of dimensionality" and computational complexity problem. In fact, in the following, we review some of the most significant web attack detection systems and the different solutions that they apply in order to address this problem.

McPAD [20] is a system working with high-order ($n > 1$) *n*-grams. It consists of an ensemble of multiple one-class support vector machine classifiers. After combining the results from the multiple models, the payload is classified as normal if the probability of the payload being normal exceeds a given threshold. The experiments are conducted on network traffic, including HTTP traffic. In order to reduce the dimensionality of the feature space for payload anomaly detection, a feature clustering algorithm is applied. Then, the dimensionality is reduced to *k*, being *k* the number of desired clusters.

Spectrogram [21] is a system that focuses on web traffic. It uses a mixture of multiple Markov chains to obtain the final likelihood score of the request being normal. The inference model tracks the *n*-gram level transitions within a string, where the likelihood of an *n*-gram is the likelihood of x_n (where x_i denotes the *i*th character within a string) and it is conditioned on the $n - 1$ preceding characters. It reduces the problem from exponential to linear complexity and takes advantages of the overlapping nature of the *n*-grams within an input string.

Rieck and Laskov [19] propose to use words (*n*-grams of variable length) instead of using fixed-length *n*-grams. They propose to analyze network protocols as a language: with words (of variable length) and boundary symbols. In the paper, they analyze different protocols that include HTTP, FTP, and SMTP traffic. The work presents a novel trie representation of *n*-grams, which is applied to transform the normal and malicious connection payloads. Then, they propose a comparison method that is applied to classify the connections.

Naiman [22] addresses the problem of intrusion detection by modeling sequences of system calls using *n*-grams. The system analyzes contiguous sequences of *n* system calls of the processes generated by an HTTP daemon. The motivation is that occurrences of sufficiently many new *n*-grams in some localized time frame constitute evidence of innovative behavior and thus of an anomaly. The detection rules are inferred by analyzing a tree of system calls.

The previous systems use the following types of datasets for their experiments:

- Its own dataset.
- The DARPA dataset.
- Unlabeled datasets.

The problem of the first option is that these datasets are normally not publicly available. The DARPA dataset is not enough for testing current WAFs (as will be carefully explained in Section 4) and unlabeled datasets are not useful for training and testing supervised learning WAFs. Because of these, it is not possible to compare other WAFs (like ours) with these systems. A difference of our system with the previously mentioned ones is that our solution applies feature selection to reduce the number of features and to deal with the high dimensionality problem of *n*-grams.

2.3. Feature selection

A feature selection method finds the smallest number of features that maximize the performance of the classification algorithm. By reducing the number of features without negative effect on detection accuracy, feature selection greatly increases the available processing time and reduces the required system resources, improving the efficiency of WAFs.

Typically, in machine learning, the feature selection methods are classified into three categories, depending on how they interact with the classifier: wrapper, filter, and hybrid models [23,24]. The wrapper model uses the performance of learning algorithms in assessing and selecting features [23,24]. The filter model considers statistical characteristics of a dataset directly without involving any learning algorithm [23,24]. When the number of features becomes very large, the filter model is more appropriate, given that it requires less computational resources; hence, it is the approach selected in the present paper.

A major challenge in the feature selection process is to choose appropriate measures that can precisely determine the relevance and the relationship between features of a given dataset. Because the relevance and the relationship are usually characterized in terms of correlation or mutual information [23,24], in this paper, we use the GeFS measure for intrusion detection [4]. This measure has been successfully tested for network [4,5] and web traffic [6,7]. An overview of the GeFS measure is explained next, and the details of the two instances used in this paper can be found in APPENDIX.

Definition 1. A generic feature selection measure used in the filter model is a function *GeFS*(*x*), which has the following form with $x = (x_1, \ldots, x_n)$:

$$GeFS(x) = \frac{a_0 + \sum_{i=1}^{n} A_i(x) x_i}{b_0 + \sum_{i=1}^{n} B_i(x) x_i}, x \in \{0, 1\}^n \tag{1}$$

In this definition, binary values of the variable x_i indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature f_i; a_0 and b_0 are constants; $A_i(x)$, $B_i(x)$ are linear functions of variables $x_1, \ldots, x_n$; *n* is number of features.

Definition 2. The feature selection problem is to find *x* $\{0, 1\}^n$ that maximizes the function *GeFS*(*x*).

$$\max_{x \in \{0,1\}^n} GeFS(x) = \frac{a_0 + \sum_{i=1}^{n} A_i(x) x_i}{b_0 + \sum_{i=1}^{n} B_i(x) x_i} \tag{2}$$

This generic feature selection contains several feature selection measures, such as the correlation feature selection (CFS) and the minimal redundancy maximal relevance (mRMR). The detail of these two measures can be found in the Appendix.

2.4. Decision trees

Classification algorithms from machine learning can help when massive volume of data has to be analyzed, reducing efficiently the amount of time and effort that would be required by manual analysis. Decision tree algorithms are predictive models that can be used as classifiers. This family of algorithms is one of the most popular [8] and experimentally successful machine learning algorithms. Additionally, this kind of algorithms are extensively used in intrusion detection; in fact, the winner of the famous DARPA intrusion detection contest [9] was an algorithm based on decision trees [10]. Because of these reasons, in this paper, we choose decision trees as classification algorithms. Because there is no standard classification algorithm for WAFs, we use four decision tree algorithms in this work: C4.5, Classification And Regression Tree (CART), Random Forest and Random Tree. In particular, the implementation versions of the algorithms used are the ones provided by the WEKA software (University of Waikato, Hamilton, New Zealand) [25]. Following, a brief explanation of each of the algorithms is presented. Further details about the algorithms can be found in [26,8].

- C4.5 was introduced by Ross Quinlan [27]. C4.5 is an algorithm used to generate decision trees that are built from a set of training data using the concept of information entropy.
  At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or another. Its criterion is choosing the attribute with the highest normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The C4.5 algorithm then recurs on the smaller sublists. The initial tree is then pruned, to avoid overfitting, with a single-pass algorithm derived from binomial confidence limits.
- CART is a recursive partitioning method that builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification). The classic CART algorithm was popularized by Breiman *et al*. [28].
  CART is a non-parametric algorithm. It generates a binary decision tree that is constructed by splitting the node that best differentiate the target variable into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. Deciding when the tree is complete and assigning a class to each terminal node are other of the key elements in the CART methodology.
- Random forest is an ensemble classifier that consists of many decision trees, and its output class is the mode of the class's output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman [29] and Adele Cutler. The method combines Breiman's "bagging" idea and the random selection of features.
- Random trees include the idea of the random selection of features. This idea was introduced independently by Ho [30,31] and Amit and Geman [32] in order to construct a collection of decision trees with controlled variation. In Ho's formulation, it is a way to implement stochastic discrimination [33]. This algorithm does not perform pruning to reduce the size of the decision tree. Some versions also provide backfitting so that unbiased probability estimates can be obtained by a hold-out set. The backfitting algorithm was introduced in 1985 by Leo Breiman and Jerome Friedman [34], and it is an iterative procedure used to fit a generalized regression additive model.

3. METHODOLOGY

In this section, our methodology for web attack detection is presented. Figure 1 shows the steps of the methodology proposed. The first step is feature construction, using the combination of expert knowledge and *n*-grams. Then, the GeFS measure is applied for dealing with the dimensionality problem. Finally, the classification phase, by means of decision trees, takes place. In this section, the details of each phase are carefully explained.

3.1. Feature construction

The web detection methodology proposed in this paper includes a new approach for feature construction that involves the combination of expert knowledge and *n*-grams. Our hypothesis is that the combination of expert knowledge features and *n*-grams would obtain better results than the basic alternatives separately and thus would enhance the effectiveness of WAFs.

The combination of the expert knowledge and *n*-grams features can be carried out in different manners. In this paper, we consider the following three alternatives for the combination approach:

- *Combine–select*. This alternative firstly merges all features extracted by expert knowledge and *n*-grams. As the number of *n*-grams leads to dimensionality problems, afterwards, feature selection is applied in order to reduce the number of features and thus to optimize the efficiency of the WAF (feature selection is explained in detail in the next section).
- *Select–combine*. Differently to the previous option, this alternative merges the expert features that are

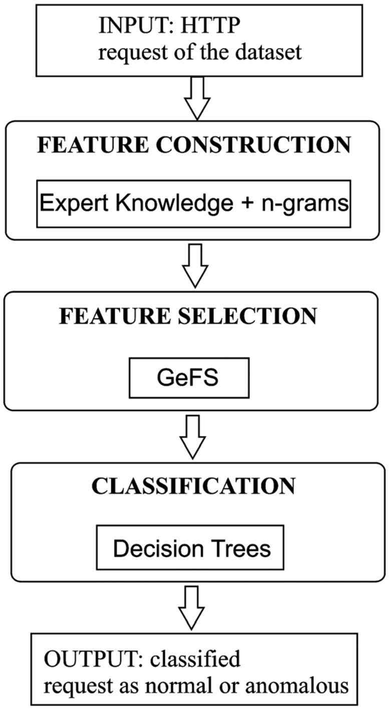

Figure 1. Steps of the methodology proposed.

already selected, that is, feature selection is performed firstly and the resulting features are merged subsequently.

- *Select-n-gram-combine*. This alternative follows the idea of the second alternative; however, it only takes the selected values of *n*-grams. The expert knowledge features are not selected, as they come from the knowledge of experts in the field. Therefore, this subset is composed of the expert knowledge features and those *n*-gram features selected by the GeFS measure. Similarly to the second alternative, feature selection is applied before the combination.

Figure 2 shows the scheme of these three alternatives. Each alternative generates different subsets of features as will be explained in Sec. 5.3.1.

3.2. Feature selection

The next step in our methodology is feature selection, which reduces the dimensionality of the features constructed in the previous step and thus enhances the efficiency of WAFs. In order to do that, in this paper we apply the GeFS measure. The search strategy for obtaining relevant features with GeFS is shown in APPENDIX. According to the first Step of this algorithm, we analyze the statistical properties of the different subsets of features in order to check whether there are linear or non-linear relations between features. The result of this analysis is used to select the appropriate instance of the GeFS measure ($GeFS_{\mathrm{CFS}}$ is recommended in the case that linear relations exist, and the $GeFS_{\mathrm{mRMR}}$ otherwise).

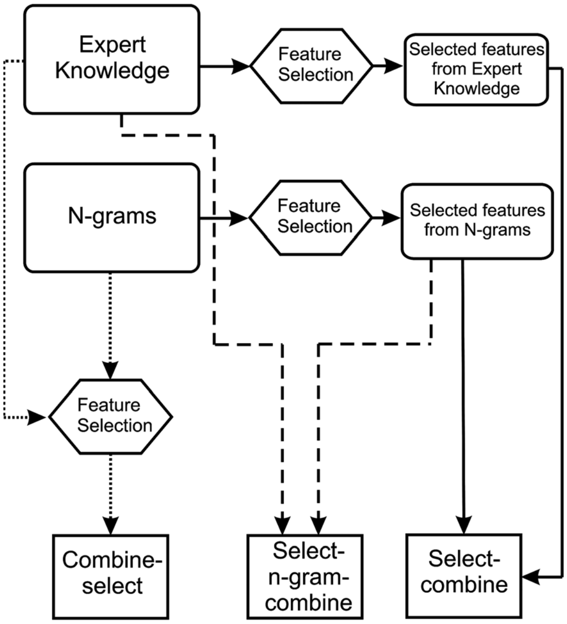

Figure 2. Scheme of the structure of the three combination alternatives.

We follow two steps to analyze the statistical properties of each alternative:

- First, the corresponding subset of features is visualized in the two-dimensional space to get a plot matrix. In the matrix, each element represents the distribution of the data points depending on, either the values of a feature and the class label, or the values of two features.
- With the aim of verifying our observations from the graphics, the next step is to calculate the correlation coefficients between the features.

With the analysis earlier, the appropriate instance of the GeFS measure is chosen. We then apply the optimization algorithm mentioned in APPENDIX to find globally optimal feature subsets by means of the GeFS instance selected for each case. Moreover, the non-selected instance is also applied to the ECML/PKDD 2007 dataset to see how the wrong choice of the GeFS instance would negatively affect the results.

3.3. Classification algorithm

Finally, the classification algorithms are run in order to classify the requests (as normal or anomalous) and test the detection results.

The experiments are conducted with both the basic cases (expert knowledge and *n*-grams) and with the combination subsets in order to test our hypothesis: that the combination would obtain better results than the basic alternatives individually.

In the classification phase, four different decision trees are applied in this paper in order to evaluate the performance of the system, namely C4.5, CART, Random Forest, and Random Tree [26].

4. DATASET DESCRIPTION

Using a good dataset is critical for testing a system. Unfortunately, in the area of web intrusion detection, the task of obtaining appropriate datasets is not easy, and it usually faces several problems, such as datasets not publicly available, datasets not labeled, or datasets not-realistic (a good analysis of this problem can be found in [35]). Because, in this paper, we focus on web attack detection, a dataset containing HTTP traffic is needed. In intrusion detection, the DARPA dataset [9,36] presented in 1998 and 1999 by the Massachusetts Institute of Technology (MIT), has been widely used for testing systems. This dataset contains network traffic, including HTTP samples. However, the DARPA dataset has been criticized by the intrusion detection system community [37]. Moreover, its HTTP traffic is not appropriate for web attack detection given that it is out of date and it does not include many of the actual attacks (note how much web applications and web attacks have changed during the last decade). Therefore, this dataset cannot be used to test our methodology.

Due to these reasons, we choose the ECML/PKDD 2007 dataset [11] for the experiments, a labeled dataset that is publicly available. Additionally, the dataset contains only HTTP traffic; therefore, it is ideal for testing WAFs. This dataset was generated for the ECML/PKDD 2007 Discovery Challenge [11], and it is divided into two parts: the training dataset and the test one. In particular, in this paper, the training dataset is used, which is composed of 50 000 samples, where 20% are attacks and the rest are normal requests. The requests of this dataset are independent from each other, and it includes different types of modern attacks. Concretely, the classes of attacks included are the following: XSS, SQL Injection, LDAP Injection, XPATH Injection, Path Traversal, Command Execution, and Server-Side Include attacks. Each request is labeled with specifications of normal traffic or the corresponding class of attack.

5. EXPERIMENTAL SETTINGS

In this section, the feature-construction and feature selection settings are explained for the basic cases (expert knowledge and *n*-grams) and for the three combination cases.

5.1. Expert knowledge

This section contains the settings of the feature construction and feature selection phases for the expert knowledge case.

5.1.1. Feature construction.

Using our expert knowledge, we construct 30 features that are considered relevant for the detection of web attacks. These features are shown in Table I. Some of the features refer to the length of different parts of the request because length is an important element to be considered in the detection of attacks such as buffer overflow. From our knowledge about web attacks, we have observed that the non-alphanumeric characters are present in many injection attacks; therefore, we consider four kinds of characters:

- Letters.
- Digits.
- Non-alphanumeric characters that have special meaning in a set of programming languages (referred in Table I as “special” chars).
- Other characters (non-alphanumeric characters that are not included in the third category).

Several features in the list refer to the number of appearances of these four types of characters in both the path and the argument’s values. Another feature is built studying the entropy of the bytes composing the requests. Additionally, we have collected the keywords of several programming languages that are often used in the injection attacks, and other features are built by counting their appearances in different parts of the request.

5.1.2. Feature selection.

As explained in Sec. 3.2, we follow two steps to choose the appropriate instance of the GeFS measure. In this section, we show how to do this for the case of expert knowledge. Firstly, the data points are visualized in the two-dimensional space, getting a plot matrix where each element represents the distribution of data points depending on, either the values of a feature and the class label, or the values of two features. The distribution of all the possibilities (representation of two features and representation of every feature versus the class label) are studied; however, for brevity, here, only two significant examples are shown (Figures 3 and 4). These figures are examples of the datapoint distribution of the expert knowledge subset. In the first example, the length of the path is plotted versus the length of the arguments and, in the second example, the length of the path versus the number of letters in the arguments. Black asterisks represent the normal requests, and magenta circles represent the anomalous ones. As can be observed in the figures, there is non-linear relationship between the features extracted bymeans of expert knowledge.

The next step is to calculate the correlation coefficients between the features in order to verify the observations from the graphics. The calculation reveals that the non-linear relations between features are more representative. Indeed, more than the 83% of the correlation coefficients are lower than 0.09. Therefore, in this case, the $GeFS_{\mathrm{mRMR}}$ measure is chosen for feature selection.

Table I. Names of 30 features that are considered relevant for the detection of web attacks.

Feature name	Feature name
Length of the request	Length of the path
Length of the arguments ⋆	Length of the header "Accept" †
Length of the header "Accept-Encoding"	Length of the header "Accept-Charset"
Length of the header "Accept-Language"	Length of the header "Cookie"
Length of the header "Content-Length"	Length of the header "Content-Type"
Length of the Host	Length of the header "Referer"
Length of the header "User-Agent"	Method identifier
Number of arguments	Number of letters in the arguments
Number of digits in the arguments	Number of "special" char in the arguments • ⋆
Number of other char in the arguments • ⋆	Number of letters char in the path
Number of digits in the path	Number of "special" char in the path
Number of other char in path	Number of cookies
Minimum byte value in the request	Maximum byte value in the request
Number of distinct bytes † ⋆	Entropy
Number of keywords in the path ⋆	Number of keywords in the arguments

The symbol • refers to features selected by the $GeFS_{\mathrm{CFS}}$ for the expert knowledge subset, to features selected by the $GeFS_{\mathrm{mRMR}}$ for the expert knowledge subset, † to the characters selected by $GeFS_{\mathrm{CFS}}$ for the combine–select subset, and ⋆ to the characters selected by $GeFS_{\mathrm{mRMR}}$ for the combine–select subset.

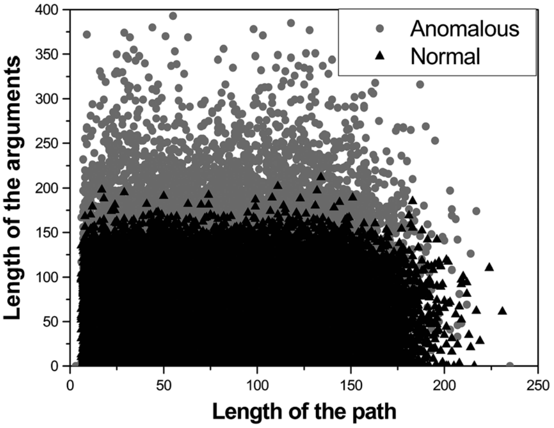

Figure 3. Sample distribution of the expert knowledge datapoints.

In Figure 5, the reduction in the number of features after applying feature selection can be observed. The figure represents the number of features of the full subset (before feature selection) and the number of features after applying the $GeFS_{\mathrm{CFS}}$ and $GeFS_{\mathrm{mRMR}}$ instances. In this case, the measure selected ($GeFS_{\mathrm{mRMR}}$) reduces significantly (80%) the number of features, but it is not the one that gets the highest reduction. However, apart from reducing the number of features, in a WAF, it is also important to consider the detection accuracy. Section 6 takes care of this aspect. The features selected by each instance of the GeFS measure can be found in Table I. The symbol is used to point out the expert knowledge features that are selected by the $GeFS_{\mathrm{CFS}}$ measure and the ones selected by the $GeFS_{\mathrm{mRMR}}$ measure.

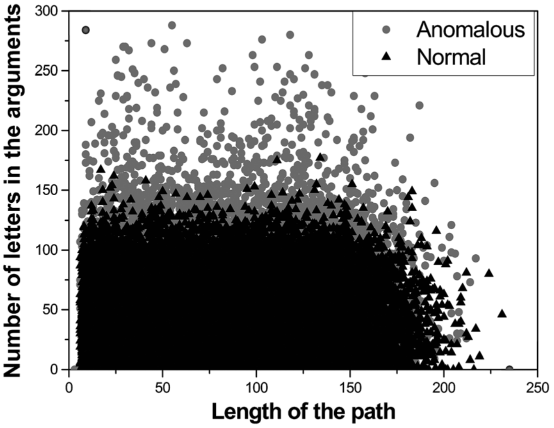

Figure 4. Sample distribution of the expert knowledge datapoints.

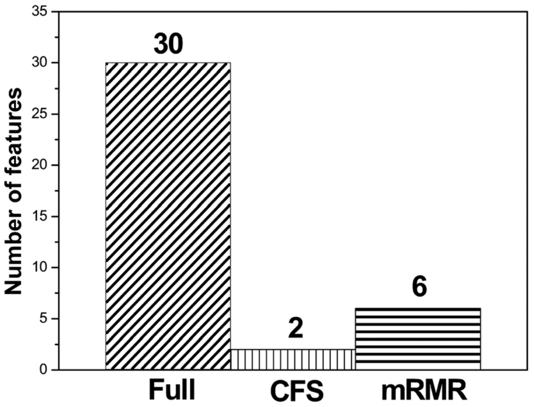

Figure 5. Number of features for the expert knowledge subset.

5.2. *N*-grams

In this section the processes of feature construction and feature selection for the *n*-gram case are described.

5.2.1. Feature construction.

In this paper, we set $n = 1$, thus the number of 1-gram is 256. We have chosen $n = 1$ because it is the simplest case. Cases with $n > 1$ require high cost in time and computational complexity, what is not appropriate for WAFs operating in real environments or scenarios with resource constraints. Furthermore, it would be expected that the results would improve as n increases; however, other papers using *n*-grams in HTTP traffic [21,20] show that it is not necessary the case. Therefore, we have not considered the case $n > 1$ in this paper.

With the assumption that payloads of normal traffic are different from payloads of attack traffic, we use the following automatic feature construction method, which is based on *n*-grams extraction.

Given an HTTP request p, a feature vector of p is constructed as follows:

$$x_p = \left(x_1, x_2, \ldots, x_2^{8n}\right),$$

where x_i is the number of appearances of $n\text{-gram}_i$ in p. Therefore, in our case, the vector constructed for every request represents the number of appearances of each character (1-gram) in the HTTP request.

In contrast to Spectrogram [21], which examines only a small part of the HTTP request for feature construction (the parameter names and their respective values), our system considers the whole HTTP request for feature construction. In our previous work [7], we also used some parts of the request; however, from the security point of view, we consider that it is more appropriate to analyze the whole request because it allows detecting attacks embedded in any part of the request, such as cookie tampering attacks, which are included in the HTTP headers.

The result of the 1-gram extraction was that only 96 features (37.5% of the 256) appear (at least once) in the ECML/PKDD 2007 dataset. These 96 features are listed in Table II.

5.2.2. Feature selection.

In this subsection, we show how to choose the appropriate instance of the GeFS measure to select important features extracted by using *n*-grams. Firstly, the data points of this subset are visualized in the two-dimensional space. In Figures 6 and 7, two sample distributions of the *n*-gram subset are represented. Figure 6 shows the number of appearances of the character "a" versus the class label, and Figure 7 shows the number of appearances of the character "a" versus the number of appearances of the character ">". As can be observed when looking at the figures, there are linear relations between the *n*-gram features extracted from the ECML/PKDD dataset.

Regarding the correlation coefficients of the *n*-gram subset, they show that the subset has many features that are linearly correlated to each other. In fact, more than 52% of the correlation coefficients are greater than 0.1. Hence, the selected measure for the *n*-gram subset is $GeFS_{\mathrm{CFS}}$.

Figure 8 shows the number of features of the full-set and the number of features after feature selection, for the two instances of the GeFS measure. The GeFS instances dismiss many of the features of the full set that are considered irrelevant or redundant for detecting web attacks. The GeFS instances dismiss many of the features of the full-set that are considered irrelevant or redundant for detecting web attacks. In this case, $GeFS_{\mathrm{mRMR}}$ provides the smallest number of selected features. However, it is also important to consider the detection accuracy as will be seen in Section 6. The *n*-grams selected by both instances can be seen in Table II, which contains the symbol to refer the *n*-gram features that are selected by $GeFS_{\mathrm{CFS}}$ and for the features selected by the $GeFS_{\mathrm{mRMR}}$ measure. It is remarkable that indeed, some of the 1-grams selected are critical for the detection of

Table II. Ninety-six characters appearing in the ECML/PKDD 2007 dataset at least once.

Features							
k ⋆	5	g	LF •	a ⋆	3	=	;
_	:	″ ⋆	l	9	0	D	h
z	x	$ •	Q	W	f	H	E
R	4	Y	.	7	,	6	q
p	e	r	L	i	b	&	m
@	C	Space •	s ⋆	2	Z	A	u
c	F	v	1	y	LF	?	I
M	j	8	n	d	- •	t †	P
*	K	/	G	V	U	w	T
S	o	N	J	+ •	B	′ •	]
O	X	% •	) •	| ⋆	(•	\ • ⋆	~
< •	>	[•	! •	# •	{	‘• †	}

The symbol • refers to the 15 characters selected by the $GeFS_{\mathrm{CFS}}$ measure for *n*-grams, ◊ to the five characters selected by the $GeFS_{\mathrm{mRMR}}$ measure for *n*-grams, † to the characters selected by $GeFS_{\mathrm{CFS}}$ for the combine–select alternative, and ⋆ to the characters selected by $GeFS_{\mathrm{mRMR}}$ for the combine–select alternative.

DOI: 10.1002/sec

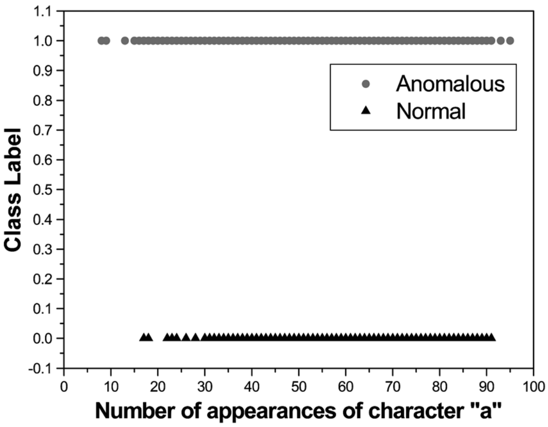

Figure 6. Sample distribution of the *n*-gram datapoints.

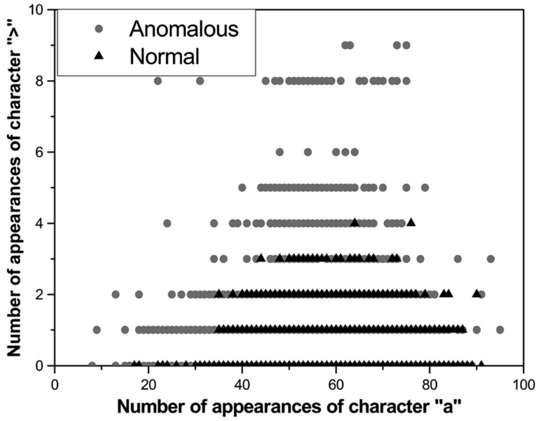

Figure 7. Sample distribution of the *n*-gram datapoints.

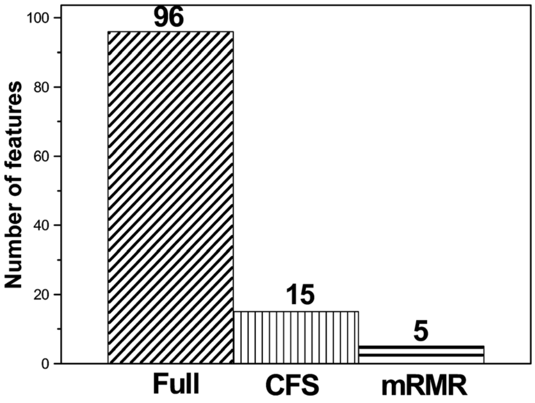

Figure 8. Number of features for the *n*-gram subset.

web attacks: for instance, the quotation mark (') is included in many SQL injection attacks, and the character "<" is typically appearing in scripts such as the ones used in XSS attacks.

5.3. Combination

This subsection explains the settings for the combination of expert knowledge with *n*-grams.

5.3.1. Feature construction.

In the combination case, the three alternatives described in Section 3.1 are used for feature construction. Each alternative generates different subsets of features, explained as follows:

- *Combine-select*. With this alternative, the resulting subset of features is composed of 126 features in total, corresponding to the combination of the 30 features from expert knowledge and the 96 features from *n*-grams. Then, feature selection is applied.
- *Select–combine*. In this case, there are four options for generating the subsets of features, corresponding to the two instances of the GeFS measure for selecting features from expert knowledge and *n*-grams. However, what makes sense is to choose the subset composed of the features selected by the appropriate instances of the GeFS measure ($GeFS_{\mathrm{mRMR}}$ in the case of expert knowledge and $GeFS_{\mathrm{CFS}}$ for *n*-grams). That is, the subset composed of 15 *n*-gram features selected by $GeFS_{\mathrm{CFS}}$ and six features selected by $GeFS_{\mathrm{mRMR}}$ from expert knowledge, which is denoted by *mRMR + CFS*. Even though, we also show another subset with features selected by different GeFS instances, to see how the wrong choice of feature selection methods would negatively affect the detection performance. As an example, we create a second subset combining six features selected by mMRM from expert knowledge (as in the previous *select–combine* subset), and in this case, five features selected by $GeFS_{\mathrm{mRMR}}$ from *n*-grams (the opposite selection than in the previous *select–combine* subset). This last subset is called *mRMR + mRMR*. Because the features that compose these two subsets are already selected, it is not necessary to apply feature selection again.
- *Select–n-gram–combine*. As only *n*-grams are selected with this alternative, two subsets can be considered, corresponding to the different instances of the GeFS measure. The subset that should be used in this case is the one using the $GeFS_{\mathrm{CFS}}$ instance for *n*-grams; however, the other subset is also included to show the importance of the right selection of the GeFS instance.

Table III summarizes the structure of the subsets corresponding to each combination alternative.

All these subsets, together with the expert knowledge and *n*-gram subsets, are used in the experiments and will be explained in Section 6.

5.3.2. Feature selection.

In this work, feature selection is utilized to deal with the high dimensionality problem and thus to guarantee the

Table III. Description of the subsets corresponding to each combination alternative.

Alternative	Total Number of Features	Expert Knowledge Features	*N*-gram Features	Subset Name
Combine–select	126	30	96	Combine–select
Select–combine	21	6 (*GeFS*$_{\mathrm{mRMR}}$)	15 (*GeFS*$_{\mathrm{CFS}}$)	mRMR + CFS
	11	6 (*GeFS*$_{\mathrm{mRMR}}$)	5 (*GeFS*$_{\mathrm{mRMR}}$)	mRMR + mRMR
Select–*n*-gram–combine	45	30	15 (*GeFS*$_{\mathrm{CFS}}$)	Expert + CFS
	35	30	5 (*GeFS*$_{\mathrm{mRMR}}$)	Expert + mRMR

effectiveness of the WAF. In the case of the combination, it is especially useful as the number of features is higher. In this section, the steps for the selection of the proper GeFS instance are performed for the three alternatives of the combination case. Regarding the *combine-select* subset, Figure 9 is a sample of its data point distribution that plots the length of the header "Accept-Charset" versus the number of digits in the path. And the sample in Figure 10 represents the length of the header "Accept-Charset" versus the length of the header "Accept-Language".

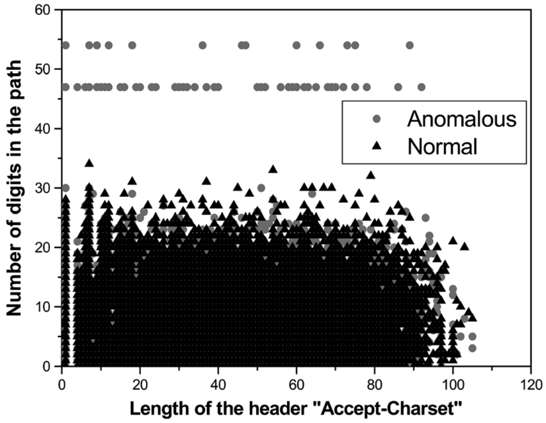

Figure 9. Sample distribution of the combine–select datapoints.

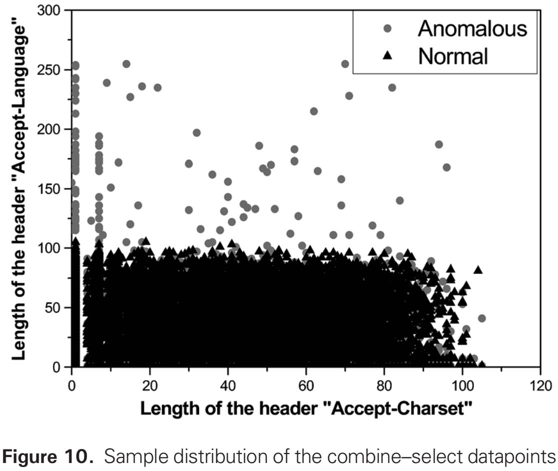

Figure 10. Sample distribution of the combine–select datapoints.

In the case of the *select-combine* and *select-n-gram-combine* alternatives, it is not necessary to study their statistical properties as the features in these subsets are already selected.

Regarding the correlation coefficients, when analyzing the calculation corresponding to the *combine-select* subset, we observed that more than 76% of the coefficients are less than 0.09, which means that there are non-linear relations between the features of this subset, and therefore, the *GeFS*$_{\mathrm{mRMR}}$ measure is the proper one for this case. It is not necessary to calculate the correlation coefficients for the *select-combine* and *select-n-gram-combine* subsets because the features are already selected.

For the case of the *combine-select* subset, Figure 11 shows the number of features of the full subset and for the subsets after feature selection. The reduction is large: 91.27% for the *GeFS*$_{\mathrm{mRMR}}$ instance, which is the selected one but not the one that gets the bigger reduction (see Section 6 for details of the performance).

The features selected from the *combine-select* subset, are represented in Tables I and II with *y* for *GeFS*$_{\mathrm{CFS}}$ and for *GeFS*$_{\mathrm{mRMR}}$. Note that many features selected by the *GeFS*$_{\mathrm{CFS}}$ and *GeFS*$_{\mathrm{mRMR}}$ measures are the same in the case of *n*-grams, expert knowledge as well as in the selection-combination case, which indicates that they are important features for detecting web attacks.

The number of features of the *select-combine* and *select-n-gram-combine* subsets can be seen in Figure 12. In summary, Table IV shows the GeFS instances chosen for every subset of the ECML/PKDD 2007 dataset.

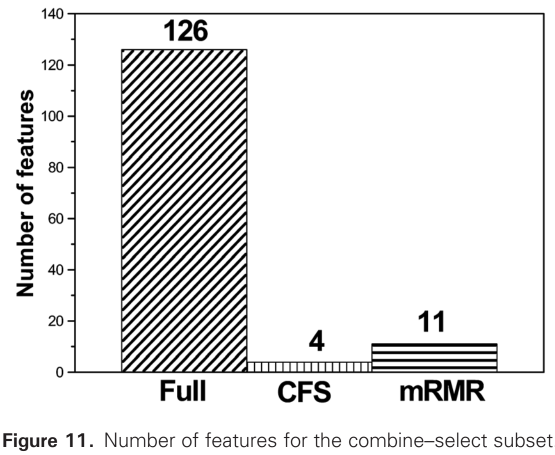

Figure 11. Number of features for the combine–select subset.

DOI: 10.1002/sec

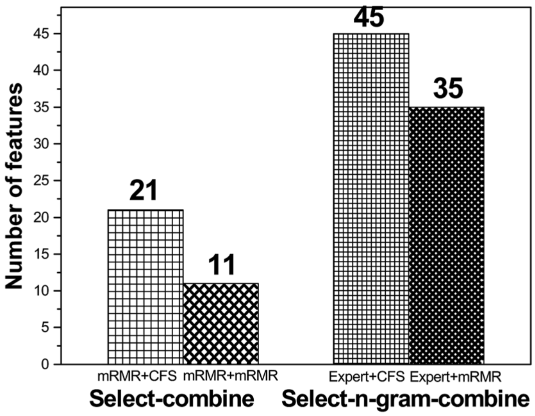

Figure 12. Number of features of the select–combine and select–*n*-gram–combine subsets.

Table IV. GeFS instance chosen for every subset of the ECML/PKDD 2007 dataset.

Subset	GeFS instance
Expert knowledge	$GeFS_{\mathrm{mRMR}}$
n-gram	$GeFS_{\mathrm{CFS}}$
Combine–select	$GeFS_{\mathrm{mRMR}}$
Select–combine	Nothing
Select–*n*-gram–combine	Nothing

5.4. Classification algorithm

The experiments are conducted with all the previously mentioned subsets of features, corresponding to the basic and the three combination cases, in order to compare their results and conclude which is the best alternative.

For the classification phase, four decision tree algorithms are applied to evaluate the performance of the WAF: C4.5, CART, Random Forest and Random Tree [26]. Concretely, we have used the implemented versions of these decision trees provided by the WEKA software [25]. For the C4.5 algorithm, the WEKA tool uses the J48 implementation. Regarding Random Tree, new versions of the WEKA tool also provide an option to perform backfitting allowing that unbiased probability estimates can be obtained by a hold-out set.

The experiments are conducted with 10-fold cross validation, and the rest of the setting values are the ones set by default in the WEKA software.

The next section shows and analyzes the results obtained from the experiments.

6. EXPERIMENTAL RESULTS

Effectiveness and efficiency are important aspects to evaluate WAFs. Therefore, in this section, the results are analyzed considering not only the number of features shown before but also the detection performance. This section shows, firstly, the accuracy results obtained with decision trees and, then, a discussion examining and analyzing the results.

The detection performance of the WAF is measured in terms of detection rate and false positive rate. The detection performance of a WAF is better as the detection rate (attacks detected) is higher and the false positive rate (false alarms) is lower.

The experiments of the basic subsets of features (expert knowledge and *n*-grams) and of all the combination subsets are compared in order to test our hypothesis: that the combination would obtain better results than the basic alternatives individually.

6.1. Basic cases

As mentioned before, the experiments are conducted with four decision trees from machine learning. In this section, the accuracy results for the basic cases (expert knowledge and *n*-grams) are presented.

Table V shows the performance of the decision trees (in terms of accuracy and false positive rate) on the expert knowledge subset. The column full-set shows the results before feature selection, and the $GeFS_{\mathrm{CFS}}$ and $GeFS_{\mathrm{mRMR}}$ columns correspond to the results after feature selection. In the table, our selection of the GeFS instance is highlighted with bold numbers ($GeFS_{\mathrm{mRMR}}$ in the case of expert knowledge). As can be observed, the results are better for the $GeFS_{\mathrm{mRMR}}$ instance than for $GeFS_{\mathrm{CFS}}$, what confirms our selection of the GeFS instance.

Table V. Accuracy and false positive rate of four decision trees performed on the expert knowledge subset.

	Accuracy			False Positive Rate		
Classifiers	Full-set	$GeFS_{\mathrm{CFS}}$	$GeFS_{\mathrm{mRMR}}$	Full-set	$GeFS_{\mathrm{CFS}}$	$GeFS_{\mathrm{mRMR}}$
C4.5	95.42	89.07	**92.30**	7.8	23.2	**15.1**
CART	95.51	89.12	**92.23**	7.9	23.1	**14.7**
Random Forest	95.80	89.10	**91.53**	8.1	23.2	**15.1**
Random Tree	92.43	89.11	**88.81**	10.6	23.2	**16.3**
Average	94.79	89.10	91.22	8.6	23.18	15.3

Bold letters are used to highlight the subset selected, corresponding to the correct GeFS instance.

When looking at the number of features (Figure 5) and the detection results of expert knowledge in Table V, it is noticeable that the GeFS measure greatly reduces the number of features (from 30 to 2) while keeping almost the same detection results. $GeFS_{\text{CFS}}$ selects less features; however, its detection accuracy is also lower, and its false detection rate is much higher than the $GeFS_{\text{mRMR}}$ one.

Regarding *n*-grams, Table VI summarizes the performance of the four decision trees over this subset. The results show that our selection of the GeFS instance is correct (the $GeFS_{\text{CFS}}$ column is highlighted in bold numbers in the table). It can be observed that when the instance is chosen wrongly, the results are negatively affected, what shows the importance of the good selection of the GeFS instance. By observing the number of features in Figure 8 as well as the performance results in Table VI, it can be observed that the $GeFS_{\text{CFS}}$ measure greatly reduces the number of features (from 96 to 15) while not only keeping, but even improving the detection results (from 92.99 to 93.47). This implies that the application of the $GeFS_{\text{CFS}}$ measure improves at the same time the effectiveness and efficiency of the WAF.

6.2. Combination cases

In this section, the results corresponding to the three combination alternatives are shown.

Table VII shows the accuracy and false positive rate obtained by the decision trees for the *combine–select* subset.

Once again, the results confirm our selection (highlighted with bold numbers) of the appropriate GeFS instance, which is $GeFS_{\text{mRMR}}$ in this case. Note that when choosing the wrong instance, the results become worse.

The $GeFS_{\text{mRMR}}$ measure reduces highly the number of features while the detection results are only slightly altered, as it can be perceived when looking at both the Figure 11 and Table VII.

Regarding the *select–combine* alternative, Table VIII indicates the detection performance results for both the *mRMR + CFS* and *mRMR + mRMR* subsets. Remember that the *mRMR+CFS* subset (with bold numbers in

Table VI. Accuracy and false positive rate of four decision trees performed on the *n*-gram subset.

	Accuracy			False Positive Rate		
Classifiers	Full-set	$GeFS_{\text{CFS}}$	$GeFS_{\text{mRMR}}$	Full-set	$GeFS_{\text{CFS}}$	$GeFS_{\text{mRMR}}$
C4.5	94.12	**94.12**	89.95	9.2	**10.4**	18.7
CART	94.92	**94.16**	90.03	8.7	**10.2**	18.3
Random Forest	94.41	**93.70**	89.12	11	**10.7**	20.7
Random Tree	88.55	**91.90**	88.73	15.7	**11.9**	22
Average	92.99	93.47	89.45	11.15	10.8	19.9

Table VII. Accuracy and false positive rate of four decision trees performed on the *combine-select* subset.

	Accuracy			False Positive Rate		
Classifiers	Full-set	$GeFS_{\text{CFS}}$	$GeFS_{\text{mRMR}}$	Full-set	$GeFS_{\text{CFS}}$	$GeFS_{\text{mRMR}}$
C4.5	97.34	76.06	**92.42**	4	46.4	**14.2**
CART	97.41	76.10	**92.52**	4.2	46.2	**14.3**
Random Forest	96.98	72.32	**92.41**	5.6	47.3	**14.5**
Random Tree	92.78	72.52	**88.18**	10	48.9	**15.9**
Average	96.13	74.25	91.38	5.95	47.2	14.72

Table VIII. Accuracy and false positive rate of four decision trees performed on the *select-combine* subsets.

	Accuracy		False Positive Rate	
Classifiers	mRMR + CFS	mRMR + mRMR	mRMR + CFS	mRMR + mRMR
C4.5	**97.09**	94.52	**4.8**	10.2
CART	**96.94**	94.55	**4.9**	10.1
Random Forest	**97.56**	94.34	**4.4**	10.3
Random Tree	**95.81**	91.45	**5.8**	11.6
Average	96.88	93.71	4.98	10.55

the table) is created with the features selected by the appropriate instances of the GeFS measure and that the *mRMR+mRMR* subset is created in the aim of comparison, choosing wrongly the GeFS instances to see how this fact would affect the results. As expected, Table VIII shows that the results are better when choosing the appropriate instances of the feature selection measure. In the next section, the results are carefully discussed.

Finally, Table IX presents the detection results for the *select-n-gram-combine* case. The results of the expert + CFS subset are presented in bold numbers. The performance of the expert + mRMR subset is also shown to compare the results when not choosing the correct instance of the GeFS measure. In fact, the results in the table confirm that the good selection of the GeFS instance leads to better results. The next section discusses accurately the results, considering the accuracy and the number of features for the evaluation of each alternative.

6.3. Discussion

In this section, the results shown before are discussed. Firstly, the combination alternatives are compared with the basic cases, to see if the hypothesis is confirmed and indeed the combination outperforms the basic cases separately. Then, the combination alternatives are compared with each other, to conclude which of them is the most appropriate in each case.

In this discussion, the number of features and detection performance criteria are considered to evaluate each alternative. In order to simplify the comparison, in the following, we show several figures. Figure 13 shows a summary of the number of features for all the subsets. Also, Figures 14 and 15 represent graphically a summary of the average values of the accuracy and false positive rate shown in the previous tables (however, the comparison can also be done looking at the values in the tables). For simplicity, in some comparisons, only the detection rate values are cited, but the same comments apply also to the false positive rate.

The starting point is the comparison of the *combine–select* alternative with the basic cases. Looking at the detection results of Figures 13 and 14 corresponding to the *combine–select*, expert knowledge, and *n*-grams subsets, it can be seen that for the full subset, the results of the *combine–select* alternative are better than the results obtained by both expert knowledge and *n*-grams separately (96.13 vs 94.79 and 92.99). In this case, also the number of features of the combination is higher than the number of features of the basic cases (126 features vs 30 and 96). Nevertheless, the combination does not obtain the best

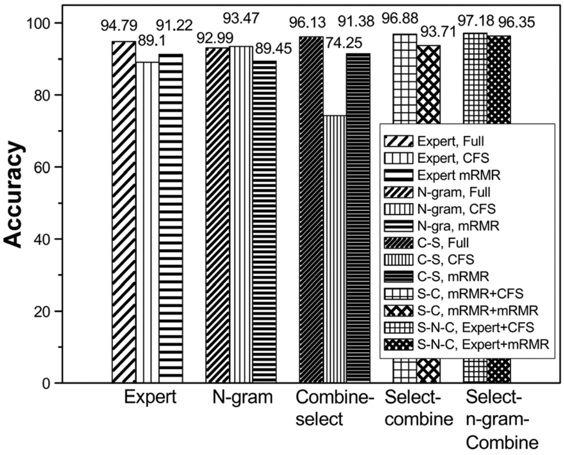

Figure 13. Accuracy in average for the expert knowledge, *n*-gram, combine–select, select–combine and, select–*n*-gram–combine subsets.

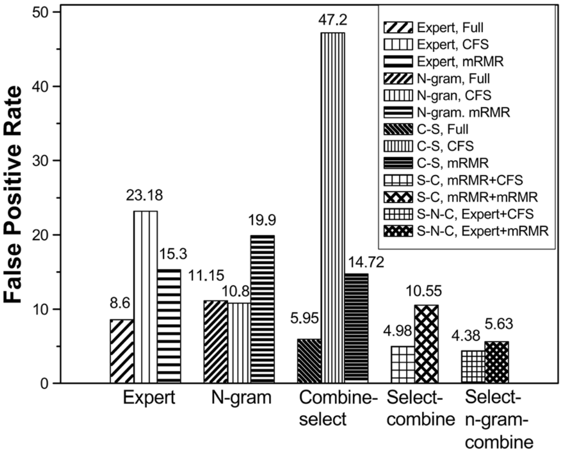

Figure 14. False positive rate in average for the expert knowledge, *n*-gram, combine–select, select–combine, and select–*n*-gram–combine subsets.

Table IX. Accuracy and false positive rate of four decision trees performed on the *select-n-gram-combine* subsets.

	Accuracy		False positive rate	
Classifiers	Expert + CFS	Expert + mRMR	Expert + CFS	Expert + mRMR
C4.5	**97.63**	96.69	**3.8**	5.3
CART	**97.58**	96.82	**4**	5
Random Forest	**97.84**	97.16	**3.8**	4.8
Random Tree	**95.69**	94.74	**5.9**	7.4
Average	**97.18**	**96.35**	**4.38**	**5.63**

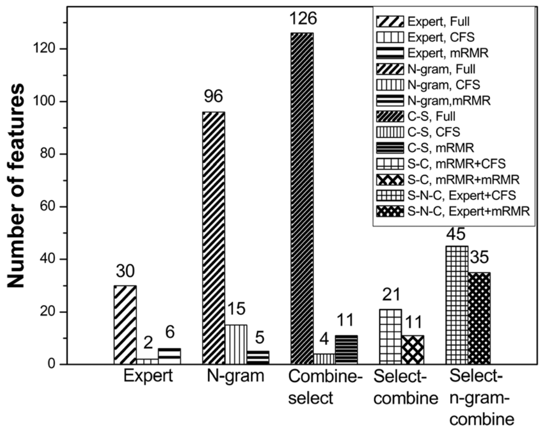

Figure 15. Number of features of the expert knowledge, *n*-gram, *combine-select*, *select-combine* and *select-n-gram-combine* subsets. In the legend, C-S denotes the *combine-select* alternative, S-C means *select-combine* and S-N-C represents the *select-n-gram-combine* case.

results after feature selection. The *n*-gram subset gets better results; however, it also uses more features (15 vs 11). The combination performs better than expert knowledge (91.38 vs 91.22) using five features more.

Next, the results of the *mRMR + CFS* subset are contrasted with the results obtained by *n*-grams and expert knowledge. Comparing the corresponding columns in Figure 13 and Figure 14, it can be observed that the *select–combine* alternative notably improves the results of the two basic cases separately, both before feature selection (96.88 vs 94.79 and 92.99) and after (96.88 vs 91.22 and 93.47). Meanwhile, the number of features of the *mRMR + CFS* subset is lower in the full-set case (21 vs 30 and 96) but not than after feature selection (21 vs 6 and 15). Therefore, this combination alternative obtains better results (higher detection accuracy and lower false positive rate) than the expert knowledge and *n*-grams cases, thus enhancing the effectiveness of WAFs and confirming our hypothesis. Additionally, it reduces the number of features before feature selection (not after).

Now, the results of the *select–n-gram–combine* alternative are analyzed. When comparing the results of the *Expert + CFS* subset with the expert knowledge and *n*-gram ones, it can be seen that this combination gets higher detection accuracy (Figure 13) and lower false positive rate (Figure 14) than the two basic alternatives separately, both before (97.18 vs 94.79 and 92.99) and after (97.18 vs 91.22 and 93.47) feature selection, which confirms our hypothesis. This combination alternative uses more features (45) than the other alternatives (30, 6, 15) except for the full-set of *n*-grams (96). Similar to the previous combination alternative, this combination improves the performance of both expert knowledge and *n*-grams before and after feature selection; meanwhile, it only reduces the number of features for the full-set of *n*-grams.

Now, the results of all combination alternatives are analyzed with the aim of comparison.

Looking at Figures 13 and 14, it is noticeable that the best detection results are the ones corresponding the *select–n-gram–combine* alternative (higher detection rate and lower false positive). However, it is also the alternative using a higher number of features. Figure 15 plots the number of features vs the detection accuracy of the three combination alternatives. In the figure, it can be seen that for the combination alternatives, the higher the number of features, the best the accuracy results.

Finally, a summary graph is shown in the aim of comparison. Figure 17 represents the accuracy and false positive rate of all the subsets of the ECML/PKDD 2007 dataset. In the figure, the size of the circles is used to represent the number of features, being the coordinates of the alternative the point in the center of the circles. Figure 17 shows in detail the left-upper area of Figure 16, showing the results with a bigger size to facilitate the observation of the results. In the figures, colors are used to distinguish between the alternatives: red circles refer to expert knowledge subset, violet to *n*-grams, blue to *combine-select*, black to *select-combine* and finally green to *select-n-gram-combine*.

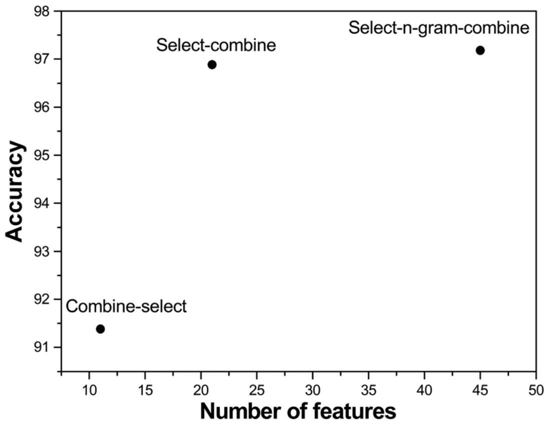

Figure 16. Accuracy and number of features of the three combination alternatives.

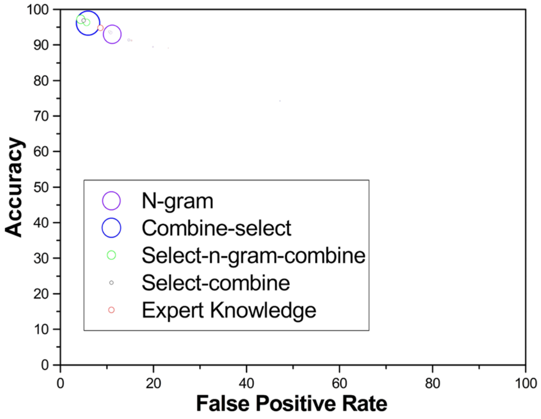

Figure 17. Accuracy and false positive rate for all the subsets of the ECML/PKDD 2007 dataset. The size of the symbols represents the number of features.

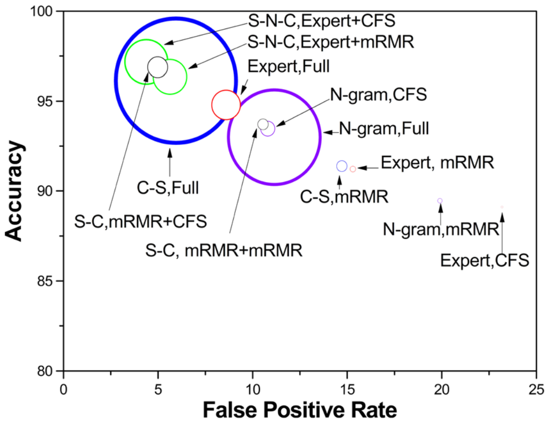

Figure 18. Detail of the accuracy and false positive rate for the subsets.

As can be seen in Figures 17 and 18, the best results are reached by the three combination alternatives, which are closer to the (0,1) point, and some of them do not use a high number of features.

7. CONCLUSIONS

In this paper, we propose a new methodology for web attack detection, which is designed to enhance both the effectiveness and efficiency of WAFs. Typically, on the design of WAFs, professionals define a set of important features that aid in the differentiation between normal traffic and attacks. On the one hand, using expert knowledge for defining these features leads to reliable results, yet it has to be performed manually. On the other hand, the *n*-gram feature construction method is automatic, but their results are unreliable. Therefore, in this paper, we propose to combine both approaches for the feature construction phase. Our hypothesis is that the combination would improve the results of expert knowledge and *n*-grams separately. However, the number of *n*-grams increases exponentially with *n*, which leads to dimensionality problems. To deal with this problem, we propose to use feature selection, in particular the GeFS measure. In our methodology, we propose three alternatives for the combination of *n*-grams and expert knowledge:

- The first alternative, called *combine–select*, merges all the features extracted by *n*-grams and expert knowledge and applies feature selection afterwards.
- In contrast, the second alternative (*select–combine*) merges the features already selected from expert knowledge and *n*-grams.
- Meanwhile, the third alternative, called *select-n-gram-combine*, is a variation of the second one and it merges the expert knowledge features (not selected) with the selected ones from *n*-grams.

Then, four decision tree algorithms from machine learning are used as classifiers of WAFs. This methodology is tested experimentally on the publicly available ECML/PKDD 2007 dataset, which is publicly available and contains labeled HTTP traffic. The results show that GeFS can reduce the number of redundant and irrelevant features, while keeping the accuracy results (or even improving them as in the *n*-grams case), thus enhancing the efficiency of WAFs. Additionally, the results obtained also confirm that indeed the combination alternatives improve the detection accuracy of *n*-gram and expert knowledge separately, which enhances the effectiveness of WAFs and verifies our hypothesis. More concretely, the first alternative improves the results of the two separate techniques in some cases, but not in all of them. Meanwhile, the second and third alternatives improve the detection accuracy in all the cases. Regarding the number of features, the *combine-select* alternative reduces the number of features in almost all the cases. From the three combination alternatives, *select-n-gram-combine* is the one that obtains the higher detection accuracy; however, it is also the one that uses a higher number of features. The *combine-select* alternative obtains worse results; nevertheless, it is the combination that uses the lower number of features. And finally, *select-combine* is in the middle of the other two alternatives concerning the number of features used and the detection results obtained. Therefore, for each specific system to be protected, the selection of the best alternative varies depending on the characteristics of the target scenario and the importance of the detection accuracy and number of features in each case. When the number of features is important, the *combine-select* alternative is recommended; however, when the important factor is accuracy (or false positive rate), it is recommended to apply the *select-n-gram-combine* one. In any case, it is shown that the methodology can enhance both the effectiveness and efficiency of WAFs, to a greater or lesser extent depending on the alternative chosen and the target scenario. For future work, higher order *n*-gram ($n > 1$) will be studied.

ACKNOWLEDGEMENTS

We would like to thank the Ministerio de Economía y Competitividad, that sponsored the project “MARSOL” (TIN2011-29709-C0201), the Instituto de Seguridad de la Información at CSIC, the Gjøvik University College, and the Norwegian Information Security Laboratory.

APPENDIX. INSTANCES OF THE GeFS MEASURE

In this section, two instances of the GeFS measure are presented: the CFS measure and the mRMR measure. These measures can be represented by the form (2) shown in Section 2.3.

The mRMR feature selection measure

In 2005, Peng *et al.* [38] proposed a feature selection method, which is based on mutual information. In this method, relevant features and redundant features are considered simultaneously. In terms of mutual information, the relevance of a feature set S for the class c is defined by the average value of all mutual information values between the individual feature f_i and the class c as follows: $D(S,c) = \frac{1}{|S|}\sum_{f_i\in S} I(f_i;c)$. The redundancy of all features in the set S is the average value of all mutual information values between the feature f_i and the feature f_j: $R(S) = \frac{1}{|S|^2}\sum_{f_i,f_j\in S} I(f_i;f_j)$. The mRMR criterion is a combination of two measures given earlier and is defined as follows:

$$\max_{S}\Big[\frac{1}{|S|}\sum_{f_i\in S} I(f_i;c) - \frac{1}{|S|^2}\sum_{f_i,f_j\in S} I(f_i;f_j)\Big] \tag{A1}$$

Suppose that there are n full-set features. The binary values of the variable x_i are used in order to indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature f_i in the globally optimal feature set. The mutual information values, $I(f_i\,;c)$ and $I(f_i\,;f_j)$, are denoted by constants c_i and a_{ij}, respectively. Therefore, the problem (3) can be described as an optimization problem as follows:

$$\max_{x\in\{0,1\}^n}\left[\frac{\sum_{i=1}^{n} c_i x_i}{\sum_{i=1}^{n} x_i} - \frac{\sum_{i,j=1}^{n} a_{ij}x_i x_j}{\left(\sum_{i=1}^{n} x_i\right)^2}\right] \tag{A2}$$

It is obvious that the mRMR measure is an instance of the GeFS measure presented in Section 2.3. This measure is denoted by $GeFS_{\mathrm{mRMR}}$.

Correlation-feature selection measure

The CFS measure evaluates subsets of features on the basis of the following hypothesis: “Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other” [39]. The following equation gives the merit of a feature subset S consisting of k features:

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

Here, $\overline{r_{cf}}$ is the average value of all feature classification correlations, and $\overline{r_{ff}}$ is the average value of all feature–feature correlations. The CFS criterion is defined as follows:

$$\max_{S_k}\left[\frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_k}}{\sqrt{k + 2\left(r_{f_1f_2} + \cdots + r_{f_if_j} + \cdots + r_{f_kf_1}\right)}}\right] \tag{A3}$$

By using binary values of the variable x_i as in the case of the mRMR measure to indicate the appearance or the absence of the feature f_i, the problem (A3) can be rewritten as an optimization problem as follows:

$$\max_{x\in\{0,1\}^n}\left[\frac{\left(\sum_{i=1}^{n} a_i x_i\right)^2}{\sum_{i=1}^{n} x_i + \sum_{i\neq j} 2b_{ij}x_i x_j}\right] \tag{A4}$$

It is obvious that the CFS measure is an instance of the GeFS measure shown in Section 2.3. This measure is denoted by $GeFS_{\mathrm{CFS}}$.

The methodology for determining appropriate instances of the GeFS measure is described next, as well as a new search strategy for obtaining subsets of relevant features by means of this measure:

- *Step 1*: Analyze the statistical properties of the given dataset in order to choose the appropriate feature selection instance ($GeFS_{\mathrm{CFS}}$ or $GeFS_{\mathrm{mRMR}}$) from the generic feature selection measure GeFS. We choose the $GeFS_{\mathrm{CFS}}$ measure if the dataset has many features that are linearly correlated to the class label and to each other. Otherwise, the $GeFS_{\mathrm{mRMR}}$ measure is selected.
- *Step 2*: According to the choice from Step 1, construct the optimization problem (2) for the $GeFS_{\mathrm{CFS}}$ measure or for the $GeFS_{mRMR}$ measure. In this step, we can use expert knowledge by assigning the value 1 to the variable if the feature is relevant and the value 0 otherwise.
- *Step 3*: Transform the optimization problem of the GeFS measure to a mixed 0–1 linear programming (M01LP) problem, which is to be solved by means of the branch-and-bound algorithm. A non-zero integer value of x_i from the optimal solution x indicates the relevance of the feature f_i regarding the GeFS measure.

REFERENCES

1. Becher M. Web application firewalls. *VDM Verlag Dr. Mueller e.K.* 2007. ISBN-10: 383640446X, ISBN-13: 978-3836404464.
2. Cavnar W, Trenkle J. *N*-gram-based text categorization. *In Proc. of Symposium on Document Analysis and Information Retrieval (SDAIR)* 1994; 161–175.
3. Marceau C. Characterizing the behavior of a program using multiple-length *n*-grams. *In Proc. of New Security Paradigms Workshop* 2000; 101–110.
4. Nguyen H, Franke K, Petrovic S. Towards a generic feature-selection measure for intrusion detection. *In 20th International Conference on Pattern Recognition* 2010; 1529–1532. Istanbul, Turkey.
5. Nguyen H, Franke K, Petrovic S. Improving effectiveness of intrusion detection by correlation feature selection. *International Conference on Availability, Reliability and Security (ARES)* 2010; 17–24. Poland.
6. Nguyen H, Torrano-Gimenez C, Alvarez G, Franke K, Petrovic S. Application of the generic feature selection

measure in detection of web attacks. *In Proc. of International Conference on Computational Intelligence in Security for Information Systems (CISIS)* 2011; 25–32. Malaga, Spain.

7. Torrano-Gimenez C, Nguyen HT, Alvarez G, Petrović S, Franke S. Applying feature selection to payload-based web application firewalls. *In Proc. of the 3rd International Workshop on Security and Communication Networks at Gjøvik University College* 2011; 75–81.
8. Wu X, Kumar V, Quinlan JR, *et al*. Top 10 algorithms in data mining. *Knowledge and Information Systems* 2008; **14**(1): 1–37.
9. Lippmann RP, Fried DJ, Graf I, *et al*. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. *In Proc. of the DARPA Information Survivability Conference and Exposition (DISCEX)* 2000. Hilton Head, South Carolina.
10. Pfahringer B. Winning the KDD99 classification cup: bagged boosting. *SIGKDD Explor. Newsl.* 2000; **1**:65– 66.
11. Rassi C, Brissaud J, Dray G, Poncelet P, Roche M, Teisseire M. Web analyzing traffic challenge: description and results. *In Proc. of the Discovery Challenge ECML/PKDD* 2007; 47–52.
12. Otero F, Silva M, Freitas A, Nievola J. Genetic programming for attribute construction in data mining. *In Genetic Programming: Proc. 6th European Conference (EuroGP)* 2003.
13. Lim SH, Wang LL, DeJong G. Explanation-based feature construction. *In Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI)* January 2007: 931–936. Hyderabad, India.
14. Kruegel C, Vigna G, Robertson W. A multi-model approach to the detection of web-based attacks. *Computer Networks* 2005; **48**(5): 717–738.
15. Torrano-Gimenez C, Perez-Villegas A, Alvarez G. An anomaly-based web application firewall. In *International Conference on Security and Cryptography (SECRYPT).* INSTICC Press: Milan (Italy), 2009; 23–28.
16. Torrano C, Perez A, Alvarez G. A self-learning anomaly-based web application firewall. *Advances in Intelligent and Soft Computing* Springer-Verlag, 2009; **63**: 85–92.
17. Wang K, Parekh J, Stolfo S. Anagram: a content anomaly detector resistant to mimicry attack. *In Recent Advances in Intrusion Detection (RAID)* 2006; 226–248.
18. Wang K, Stolfo S. Anomalous payload-based network intrusion detection. *In Recent Advances in Intrusion Detection (RAID)* 2004; 203–222.
19. Rieck K, Laskov P. Detecting unknown network attacks using language models. *In Proc. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA) Conference* 2006; 74 –90.
20. Perdisci R, Ariu D, Fogla P, Giacinto G, Lee W. McPAD: a multiple classifier system for accurate payload-based anomaly detection. *Computer Networks* 2009; 864–881.
21. Song Y, Keromytis A, Stolfo S. Spectrogram: a mixture-of-markov-chains model for anomaly detection in web traffic. *In Proc. of Network and Distributed System Security Symposium (NDSS)* 2009.
22. Naiman DQ. Statistical anomaly detection via httpd data analysis. *Computational Statistics & Data Analysis* 2004; **45**: 51–67.
23. Guyon I, Gunn S, Nikravesh M, Zadeh LA. Feature extraction: foundations and applications. *Series Studies in Fuzziness and Soft Computing* 2005. Springer.
24. Liu H, Motoda H. Computational Methods of Feature Selection. *Chapman & Hall/CRC* 2008.
25. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* 2009; **11**: 10–18.
26. Duda RO, Hart PE, Stork DG. *Pattern Classification*. John Wiley & Sons: USA, 2001.
27. Quinlan JR. C4.5: programs for machine learning. *Machine Learning* Morgan Kaufmann Publishers 1994; **16**(3): 235–240. DOI: 10.1007/BF00993309.
28. Breiman L, Friedman JH, Olshen R, Stone CJ. *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software: Pacific California, 1984.
29. Breiman L. Random forests. *Machine Learning* 2001; **45**(1): 5–32. DOI:10.1023/A:1010933404324.
30. Ho T. Random decision forest. *In Proc. of the 3rd International Conference on Document Analysis and Recognition* 1995; 278–282.
31. Ho T. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1998; **20**(8): 832–844. DOI:10.1109/34.709601.
32. Amit Y, Geman D. Shape quantization and recognition with randomized trees. *Neural Computation* 1997; **9** (7): 1545–1588. DOI:10.1162/neco.1997.9.7.1545.
33. Kleinberg E. An overtraining-resistant stochastic modeling method for pattern recognition. *The Annals of Statistics* 1996; **24**(6): 2319–2349.
34. Breiman L, Friedman JH. Estimating optimal transformations for multiple regression and correlations (with discussion). *Journal of the American Statistical Association* 1985; **80**(391): 580–619. DOI:10.2307/2288473.
35. Ingham K, Inoue H. Comparing anomaly detection techniques for HTTP. *In Proc. of the 10th International Conference on Recent Advances in Intrusion Detection (RAID)* 2007; 42–62.
36. Lippmann R, Haines JW, Fried DJ, Korba J, Das K. The 1999 DARPA off-line intrusion detection

evaluation. In *Proc. Recent Advances in Intrusion Detection (RAID)*. Springer-Verlag: New York, 2000; 162–182.

37. McHugh J. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *Proc. ACM Transactions on Information and System Security (TISSEC)* 2000; **3**(4): 262–294.
38. Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2005; **27**(8): 1226–1238.
39. Hall M. Correlation based feature selection for machine learning. *Doctoral Dissertation, University of Waikato, Department of Computer Science* 1999.