Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 4102 – 4117
© TÜBİTAK
doi:10.3906/elk-1812-18

A hybrid feature-selection approach for finding the digital evidence of web application attacks

Mohammed BABIKER[1*], **Enis KARAARSLAN**[2], **Yaşar HOŞCAN**[1]
[1]Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Muğla Sıtkı Koçman University, Muğla, Turkey

Received: 04.12.2018 • **Accepted/Published Online:** 05.05.2019 • **Final Version:** 26.11.2019

Abstract: The most critical challenge of web attack forensic investigations is the sheer amount of data and level of complexity. Machine learning technology might be an efficient solution for web attack analysis and investigation. Consequently, machine learning applications have been applied in various areas of information security and digital forensics, and have improved over time. Moreover, feature selection is a crucial step in machine learning; in fact, selecting an optimal feature subset could enhance the accuracy and performance of the predictive model. To date, there has not been an adequate approach to select optimal features for the evidence of web attack. In this study, a hybrid approach that selects the relevant web attack features by combining the filter and wrapper methods is proposed. This approach has been validated by experimental measurements on 3 web attack datasets. The results show that our proposed approach can find the evidence with high recall, high accuracy, and low error rates. We believe that the results presented herein may help us to improve accuracy and recall of machine learning techniques; particularly, in the field of web attack investigation. The tools that use this approach may help digital forensic professionals and law enforcement in finding the evidence much more efficiently and faster.

Key words: Web application attacks, machine learning, feature selection, digital evidence

1. Introduction

Finding digital evidence in the investigation of a web attack is fundamental. Web application attacks have become more sophisticated, and the data seized for analysis has increased drastically. Consequently, investigators must deal with a vast amount of data. It is difficult to rely on traditional methods to obtain digital evidence in an acceptable amount of time and with the least amount of effort.

Several definitions of digital forensics have been proposed. The wide definitions from UK forensic science regulator [1] and the digital forensics research workshop [2] could be combined and summarized, for the use of scientifically derived and proven methods toward obtaining intelligence from digital evidence for use in investigations, or in criminal proceedings. Overall, these definitions are comprehensive. Therefore, many scientific and engineering methods may occur to achieve a high degree of efficiency, but strict and agreed rules exist in the methods of extracting and preserving digital evidence. However, there is a wide controversy between the use of conventional methods and automation methods in analysis. Depending on the digital investigator's experience alone, the analysis could take a lot of time and effort, especially in the presence of a vast amount of data. Automation methods can be successful in reducing the analysis time and effort, but they also raise

[*]Correspondence: mohammedbabiker@eskisehir.edu.tr

new challenges, such as a knowledge gap and low quality results. There is a need for better methods that are applied in the digital forensic phase to lead to better results [3].

In recent years, the use of machine learning and data mining algorithms has emerged as common practice in information security [4]. It has been well established in a variety of studies that machine learning and data mining are effective in cyber-attack detection [5–11]. Scientific methods may contribute in identifying, analyzing, and reconstructing digital evidence. However, not enough attention has been paid to the role of machine learning and data mining in digital forensics so far. These techniques can be a good choice in the analysis phase of digital forensics. They can play a vital role in the detection of attacks, as they meet the high demands of automated techniques. Machine learning deals with massive amounts of data in an intelligent way; thus, it is potentially a factor in analyzing complex attacks. Most of the machine learning models are based on features to predict outcomes. However, a large number of features may lead to irrelevant results. Similarly, regarding the large number of features, redundant features can cause noise, a high error rate, and deterioration in performance. Furthermore, selecting the optimal feature subset is a major challenge, particularly, a feature subset that can lead to high accuracy results in an acceptable performance. The aim of this study is to propose an approach to find the optimal feature subset that predicts accurate and interesting outcomes.

This paper first gives a brief overview and explanation of the key terms used in digital forensics and machine learning. Next, an overview of the related work is given. Section 3 describes the approach and methods that are used to evaluate the accuracy of feature selection. Section 4 presents our results, and Section 5 gives the discussion. The study yielded a number of key findings:

- The proposed hybrid approach greatly increased the recall and accuracy of detection attacks.
- This approach was successful in reducing irrelevant features and finding optimal features to investigate web attacks. This could contribute to building machine learning models with more relevant results, low error rate, less noise, and better performance.
- The proposed hybrid approach fulfilled the potential requirements to be considered as a forensic technique in finding the digital evidence of web attacks.
- This approach could also be considered as an effective feature selection method in building machine learning and data mining models, which could be a useful aid for the intelligent automation system of the web attack forensics.

1.1. Digital evidence

Digital evidence has a pivotal role in solving forensic cases [12]. According to a definition provided by the national institute of justice, “digital evidence tends to be used to refer to information and data of value to an investigation that is stored on, received, or transmitted by an electronic device” [13]. In web application forensics, log files provide detailed events and a trace of security attacks [14]. For this reason, investigators always search for malicious entries in the log files [15, 16]. Logs as digital evidence provide extremely useful information that can help in the detection of various kinds of attacks. Likewise, knowing their types and places in web application architecture is important to get the desired result. Log files are located in different places in web architecture, as shown in Figure 1. Logs could contain evidence of web application attacks in various components of web application architecture; thus, it is necessary here to clarify the contents of those log files. For instance, application logs contain detailed information about business logic [17]. Web server logs contain important information about HTTP requests. Likewise, the access log has data to trace back user activities [18]. Furthermore, web application firewall and application level intrusion detection system logs are a rich source for attack investigations. In fact, those logs contain captured network traffic and attacks. On the other hand, firewall logs, in most cases, have a lack of application layer data, but could help in finding information related to the network layers [19]. As a result of the huge amount of web traffic, the analysis and examination of log files becomes a tedious task [20]. Consequently, there is an urgent need for automated techniques such as machine learning and data mining.

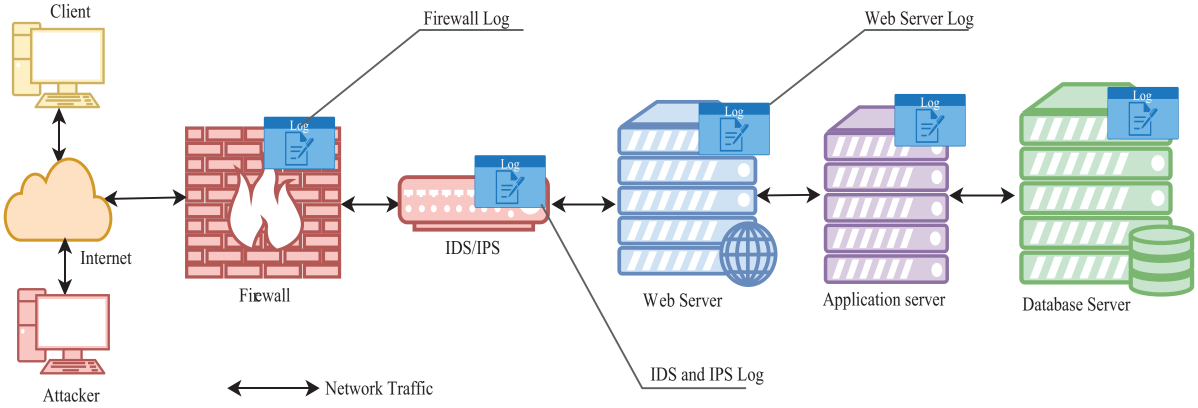

Figure 1. The main evidence log files in the web application architecture.

Forensic analysis of web attack evidence differs from the other kinds of analyses. First, web attack forensics requires that the rules and laws of digital evidence be considered in all steps, and the evidence must contain data that has a value for the proof or rejection of the incident. Second, the presence of digital evidence should play a role in the probability of the attack facts. Finally, all of the steps in the web attack analysis should be simple and easy to re-test by an expert witness. All of these reasons make the selection of features that are associated with the event of attack an important task. Accuracy here is more important than performance, especially because most of the log file analyses are done off-line, unlike the detection techniques where analysis relies more on real-time traffic. Turning now to the feature selection, it is the first step to build a model in machine learning and data mining.

1.2. Feature selection

Feature selection is a major area of interest within the field of data mining and machine learning. The goal of feature selection is to identify the relevant and important features to build predictive models. Features could be redundant and contain noise that may cause overfitting and negatively impact the performance of algorithms [21]. Optimal feature subset enhances the accuracy and performance of the predictive model and could reduce the complexity of the underlying process [22, 23]. For this reason, an automated process to obtain a subset of relevant features plays a critical role in preprocessing a dataset.

Automated feature selection methods can be categorized into filters, wrappers, and embedded methods as shown in Figure 2. The fastest method is the filter method, because it applies a statistical measure to the features independently from the prediction model. As a consequence, the filter method has a low computational cost and low risk of overfitting [24]. In contrast to the filter method, the wrapper method finds the feature subset by repetitive evaluation of subset combinations regarding the accuracy of the prediction model [25]. Therefore, the wrapper method is more computationally expensive and risky regarding overfitting, but it has

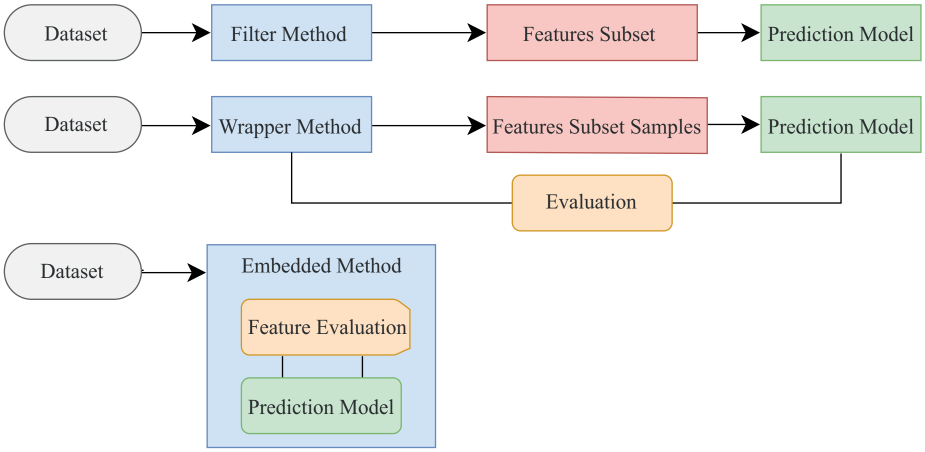

Figure 2. The main feature selection methods for machine learning.

high accuracy. Another feature selection method is the embedded method, which learns which feature subset has the best accuracy in creating time. Similar to the wrapper method, the embedded method suffers from overfitting.

2. Related works

Awareness of anomaly detection in web application attacks is not recent, having possibly first been described in [26] by Kruegel et al. In their first study, weblog files were the main source of analysis. They extracted client queries that contained successful GET requests from the log. According to their study, the main features to detect web attacks were as follows: the length of a query attribute, character distribution, structural inference parameter, attribute presence or absence, attribute order, access frequency, interrequest time delay, and invocation order. On the basis of the probability theory and statistics, they built multidetector models using the Markov model, Pearson's statistical chi-squared test, and statistical correlation. Conversely, their system was not stable due to the high number of alerts: more than 5000 alerts per day. Thus, there is a need for manual efforts. Moreover, the system suffers from false positive in detection. A new approach is therefore needed for those limitations. Robertson et al. employed generalization and characterization techniques in their second paper [27]. They automatically converted suspicious web requests into anomaly signatures, and then, similar anomalous requests were grouped. Moreover, a heuristics-based technique was used to find the attack that generated the anomalies. Unfortunately, a large number of similar alerts were manually handled, owing to false positive and false negative detection. However, the greatest research effort in their study was extracting effective features to detect web application attacks. Over time, an extensive method was developed based on their features.

The features in the detection of web attacks were studied extensively by Nguyen et al. [28]. They conducted a series of experiments in HTTP-traffic and provided a CSIC dataset that contained 30 features. In their analysis, it was shown that if there were many features linearly correlated to each other, then the correlation-based feature selection (CFS) measure could achieve better results. The CFS removed more than 63% of irrelevant and redundant features. However, the detection accuracy was reduced. Unfortunately, the feature construction methods are rather controversial, and there is no general agreement about them. This view was supported by Atienza et al. [29], who pointed out that many features in the CSIC dataset have a single value that does not provide any information to discriminate between normal and anomalous requests. For this reason, Atienza et al. removed those single value features, which led to several neural models failing in the detection of web attacks. In another study [30], the same researchers performed a similar series of experiments, but this time,they used 11 relevant features. Those features were extracted by n-grams and succeed in removing more than 95% of the irrelevant and redundant features. However, the results showed that there was no increase in accuracy. Instead, the accuracy was reduced by 6% with unreliable outcomes. In order to fill the gaps in their paper, a broader perspective that combined expert knowledge with n-gram feature construction was adopted in [31]. In their experiments, the authors identified 3 sets of combinations. They found that the combination that merged features extracted by n-grams and expert knowledge and then applied feature selection was suitable when the number of features was important. However, when the important factor was accuracy, they recommended merging the expert knowledge features with the selected features from the n-grams.

In the same vein, a number of studies have examined n-gram in the detection of web attacks. For example, Zhang et al. [32] found that a high-order n-gram, 5-gram, works well in HTTP payload when combined with Bayesian probability. On average, false positive and false negative rates were 16.604% and 18.712%, respectively. Therefore, this finding was in line with those of [30, 31, 33], However, if there is a lot of attack syntax, this could decrease the performance of the detection. Moreover, Wressnegger et al. [34] set up a series of experiments by utilizing high-order n-grams of bytes with a combination of learning schemes. Their methods perform well in the detection of web attacks with a successive identification of 81.5% attacks and 0.01% false alarms. Together, these results provide an important insight into using n-grams for the detection of web attacks, and this view was supported by Nascimento and Correia [35], who found that n-gram modeling is very accurate with a low false positive rate in detection of web attacks.

3. The hybrid feature selection for forensics (HFSF) approach

The hybrid approach is utilized, along with the integrating filter and wrapper methods, to select the optimal features. The approach described herein is geared towards solving the performance problems related to the wrapper, by reducing features through the filter. Thus, the results were ultimately higher in accuracy and computation efficiency. To highlight the advantages of the approach, Figure 3 shows the hybrid feature selection approach, which combined the filter and wrapper methods. As shown in Figure 3, the key components of the hybrid approach can be listed as follows:

- Feature construction and preprocessing: This is the first phase that preprocessed the datasets and produced the complete features. Algorithm 1 is proposed to conduct the construction of the new features. The design of the algorithm is based on the calculation of the static characteristics of the string attribute, such as the length, and total number of digits, letters, and special characters. As the string attribute in most web attack datasets is problematic, all of the string attributes are split into unigram characters as a set of numeric features representing character occurrences. Multiinterval discretization was adapted to preprocess the numerical features to partition the numerical features into subranges.

- Feature selection methods: This is the second phase, which selects the features with the hybrid feature selector. The feature selector had selection criteria to evaluate features based on various measures, such as information measures, distance measures, and dependence measures. The filtered feature subset is evaluated using the wrapper method. Accuracy and error rate statistical validation is used by applying search strategies, such as forward and backward approach, in the evaluation of the subset using the wrapper method.

- The evaluation is based on prediction models, such as Bayesian classifiers or the decision tree. The prediction model can be applied with the forwarding searching strategy, where features are added in each iteration until adding a new feature did not enhance the performance of the model. Applying the prediction model with a backward strategy is started with all of the subset features, and the feature that had less effect on the performance model is eliminated in each iteration. The searching strategy is chosen according to computation expensive in the evaluation of subsets. Finally, the feature that performed the best is chosen as one of the optimal features subsets.

```
Input: The string features F= $\langle f_1, f_2, \ldots, f_n \rangle$
Output: New constructed features FDigits[], FLetters[], FChar[],FLength[], 1-gram sequence
        FUnigram[]
$i \leftarrow 1$
while $i \leq n$ do
    Length, number of digits, number of letters, number of special characters calculated and
     constructed as new numeric features.
    $FDigits[i] \leftarrow$CountDigit(fi);
    $FLetters[i] \leftarrow$CountLetters(fi);
    $FChar[i] \leftarrow$CountSpecialChar(fi);
    $FLength[i] \leftarrow$CountLength(fi);
    Splits a string feature into unigram with max 1 gram.
    FiUnigram = CharacterNGramTokenizer(Fi,min=1,max=1)
    $i \leftarrow i + 1$
end
return  FDigits[], FLetters[], FChar[],FLength[], FUnigram[]
```

Algorithm 1: Feature construction.

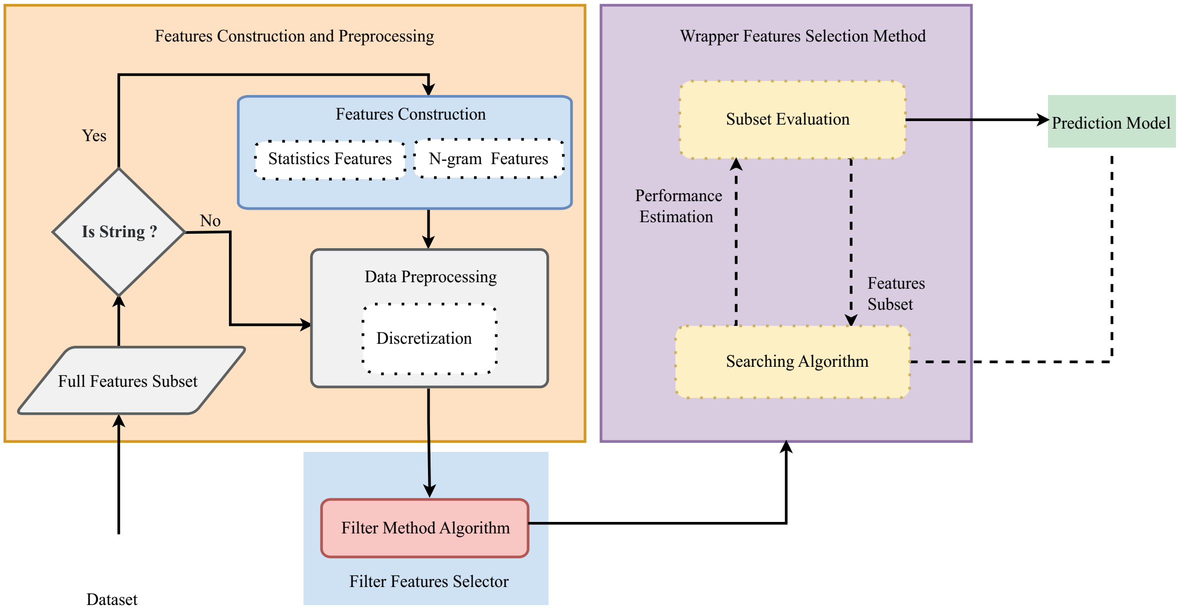

Figure 3. The Hybrid Feature Selection for Forensics Approach.

4. Experimental evaluation

The design of the experiments was based on the multilayer web application architecture, where integration technologies, such as client-side, server-side applications, business logic, and database back-end hosting were segregated into layers. The web application architecture could contain 2 types of logs. The first log is associated with network level traffic; however, the other log is related to web application layer protocol. A major advantage of using multiple datasets is that it simulates the collection of evidence and various types of logs, just as forensic investigators do in the real-world scenarios. Another advantage of using different types of logs is that it allows the evaluation of the feature selection methods in terms of stability.

In this experiment, 3 datasets were chosen for both web traffic and network traffic. First, in order to experiment on the web traffic, a single class dataset CSIC 2010 [36] and multiclass dataset ECML/PKDD'2007 [37] were employed. These 2 datasets contained labeled HTTP requests and various types of attacks, such as cross-site scripting, LDAP injection, SQL injection, buffer overflow, information gathering, files disclosure, CRLF injection, XPATH injection, SSI attacks, path traversal, and command execution. Second, to examine the network traffic, CICID2017 [38] was chosen. It is particularly useful in studying intrusion detection system (IDS) traffic and intrusion prevention system (IPS) traffic. The CICID2017 contains benign and the most up-to-date common attacks. Those attacks are labeled based on the timestamp, source IP, destination IP, and source and destination ports and protocols. Indeed, the 3 datasets can be more useful for identifying the optimal feature selection methods if they are well preprocessed. This process will be implemented in the preprocessing phase in this study.

A major problem with CSIC 2010 is the irrelevant and single value features. As a result, overfitting problems could occur. Those features were preprocessed according to the procedure used by Atienza et al. [29]. Payload and URL features contain traffic and information related to the attacks; therefore, they should not be discarded. Instead, new features were constructed from them as follows: The nominal attributes URL and payload was combined and then converted into a string attribute in order to apply a unigram on them. The new string attribute was split into characters with the frequency of occurrence as a value for each character and, as a result, new numeric attributes were generated. All of the numeric attributes were discretized with respect to class, to nominal. The discretization was done in order to partition the numeric features into subranges by adapting the multiinterval discretization proposed by Fayyad and Irani [39]. The discretization was applied using Weka supervised attribute discretization with 6 decimal places for cut points. The unigram of the text features was based on Weka's character n-gram tokenizer, where the string features were split into a set of numeric features representing character occurrence.

The same procedures were also applied in the ECML/PKDD'2007 dataset, and for CICID2017, the timestamp attribute was removed and all of the numeric attributes were discretized. Table 1 shows the number of instances and constructed features of the selected datasets. for the tests had an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz, 2801 Mhz, 4 Core(s), 8 Logical Processor(s) and 16 GB of RAM with 64-bit Windows 10 Pro Operating System. The following 3 filter feature selection methods were chosen:

- Information gain: The information gain of attribute with respect to the class is measured. Here H(Class) is Shannon's entropy for the class.

$$Infogain(Class, Attribute) = H(Class) - H(Class|Attribute). \tag{1}$$

- Chi-squared: The value of the chi-squared statistic with respect to the class is computed.

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{2}$$

Here O = the frequencies observed, E = the frequencies expected.

- CFS: The prediction and redundancy between the features are considered. Next, the highly correlated features with class and less correlated between each other are selected.

Wrapper feature selection methods were applied using 2 approaches: a forward search strategy with the Naïve Bayes and a backward strategy with the decision tree. Following that, the best filter feature selection method was chosen to be combined with the wrapper methods. Comparison of the feature selection methods was based on 2 machine learning algorithms, C4.5 decision tree algorithm was selected for its reliability and validity on web attacks. It outperforms other tree-based machine learning algorithms namely random forest, CART, and random tree on CSIC 2010 as in [28]. To represent a different approach in machine learning; the Naïve Bayes algorithm was selected because it implements Bayes' theorem for classification problems. The Naïve Bayes algorithm has the best performance in CICIDS2017 compared to other 6 machine learning algorithms as specified in [40]. The same two machine learning models have been used for comparison of feature selection methods as in [41]. One advantage of the C4.5 and Naïve Bayes is that both could work in a binary classification problem and multiclass problem. The experiments were run using cross-validation, and the accuracy, sensitivity, and specificity were

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{3}$$

$$Recall(Sensitivity) = \frac{TP}{TP+FN}, \tag{4}$$

$$Rootmeansquareerror(RMSE) = \sqrt{\frac{\sum_{i=1}^{n}(Predictedvalue - Actualvalue)^2}{n}}, \tag{5}$$

where:
TP = true positives: number of attacks predicted positive that are actually positive.
FP = false positives: number of attacks predicted positive that are actually negative.
TN = true negatives: number of attacks predicted negative that are actually negative.
FN = false negatives: number of attacks predicted negative that are actually positive.

Table 1. The number of features and instances after preprocessing the datasets.

Dataset	Original		Preprocessed	
	Instances	Features	Instances	Features
CSIC 2010	223585	17	37539	81
ECML/PKDD 2007	68269	14	50107	96
CICID 2017	170366	84	170366	83

5. Results

The first set of results, as can be seen in Table 2, examined the accuracy impact of the feature selection methods on 3 web attack datasets. On observation of the CICID2017 dataset, as shown in Table 2, shows that it is apparent that the CFS and information gain methods used were effective on the growth accuracy. Equally important, the 2 methods reduced the number of features. While CFS tended to be the most effective filter method, as it chose only 3 features, with an average accuracy of 99.5401%. On average, The hybrid feature selection with naive bayes (HFSN) was shown to have a better accuracy of 99.5759% with the reduction of the features into 2 features. On the other hand, the hybrid feature selection with decision tree (HFST) approach did not show any differences in accuracy from the CFS filter method.

Comparing the 3 filter methods and 2 hybrid approaches, as can be seen in Figure 4a, the result of recall and precision were the same in the information gain and chi-squared; however, the recall was higher in CFS. Interestingly, the recall value was observed to be the best in the HFSN method. In terms of precision, the hybrid approach did not show any significant differences. What stands out in Figure 4b is that the root mean squared error (RMSE) was in constant decline with each of the feature selection methods, and we can see that the HFSN approach gave the lowest RMSE value. For performance, the graph in Figure 4c shows that there was a marked decrease in the CPU time with the HFSN approach, as well as a decrease in the CPU time in the HFST approach and CFS methods; however, there was a slight decrease of time in the information gain and chi-squared methods.

Table 2. Accuracy of the feature selection methods in the 3 web attack datasets.

Datasets	No FS	Info.gain	Chi-squared	CFS	HFST	HFSN
	%	%	%	%	%	%
CICID2017	98.4968	98.4918	98.4918	99.5401	99.5401	99.5760
ECML/PKDD 2007	78.4162	78.4162	78.4162	83.0812	82.9275	83.07625
CSIC 2010	65.88345	65.87815	65.8768	65.43065	65.42395	65.9101

Turning now to the experimental results on the ECML/PKDD 2007 dataset, Figure 5 and Table 2 show the intercorrelations among the accuracy and RMSE measures. On the question of recall and precision, Figure 5a shows that there is a clear trend of decreasing precision in the CFS method and the hybrid methods at a precision of 0.92 for each. However, no significant decrease in precision was evident in the information gain and chi-squared methods. The hybrid approach and CFS resulted in the highest value of recall, where the recall was 0.98. It can be seen that by far, the hybrid and CfS approaches resulted in the greatest accuracy and the lowest error. Conversely, the information gain and chi-squared methods did not show any significant increase in accuracy, and no differences in the error rate were observed as shown in Figure 5b. What stands out in Figure 5c is a significant decrease in the CPU time in the hybrid approaches, reaching 0.02 s in the case of the Naïve Bayes algorithm and 0.24 s in the C4.5 algorithm.

Contrary to the accuracy results for the previous datasets, Figure 6 and Table 2 present the information gain method with the highest accuracy, at 65.87815, compared to the other filter methods in the CSIC 2010 dataset. Regarding recall and precision, Figure 6a shows that there was a significant increase in recall with the hybrid approach compared to the filter methods, with a marked drop in precision. What is interesting in Figure 6a is the high rate of recall in the CFS method and a dramatic decline in precision compared to information gain. What stands out in Figure 6b is that the RMSE gradually decreased starting from the correlation filter method CFS. Although the accuracy was enhanced on the hybrid approaches compared to the filter methods, the accuracy and RMSE of the HFSN were significantly higher than those of the HFST. No decline in the RMSE was detected in the HFST. In the same way, information gain outperformed the correlation in accuracy and RMSE as in Table 2. With respect to the CPU time, as a performance measure, Figure 6c reveals that there was a sharp 90% decrease in the CPU time with the hybrid approach in the Naïve Bayes. Likewise, the CPU time reached a peak in HFST, with a 98.8% decrease of time. What can be clearly seen in Figure 7 is the dramatic decline in the number of features with the hybrid approach. The figure shows that there was a marked increase in the average accuracy with the hybrid approach.

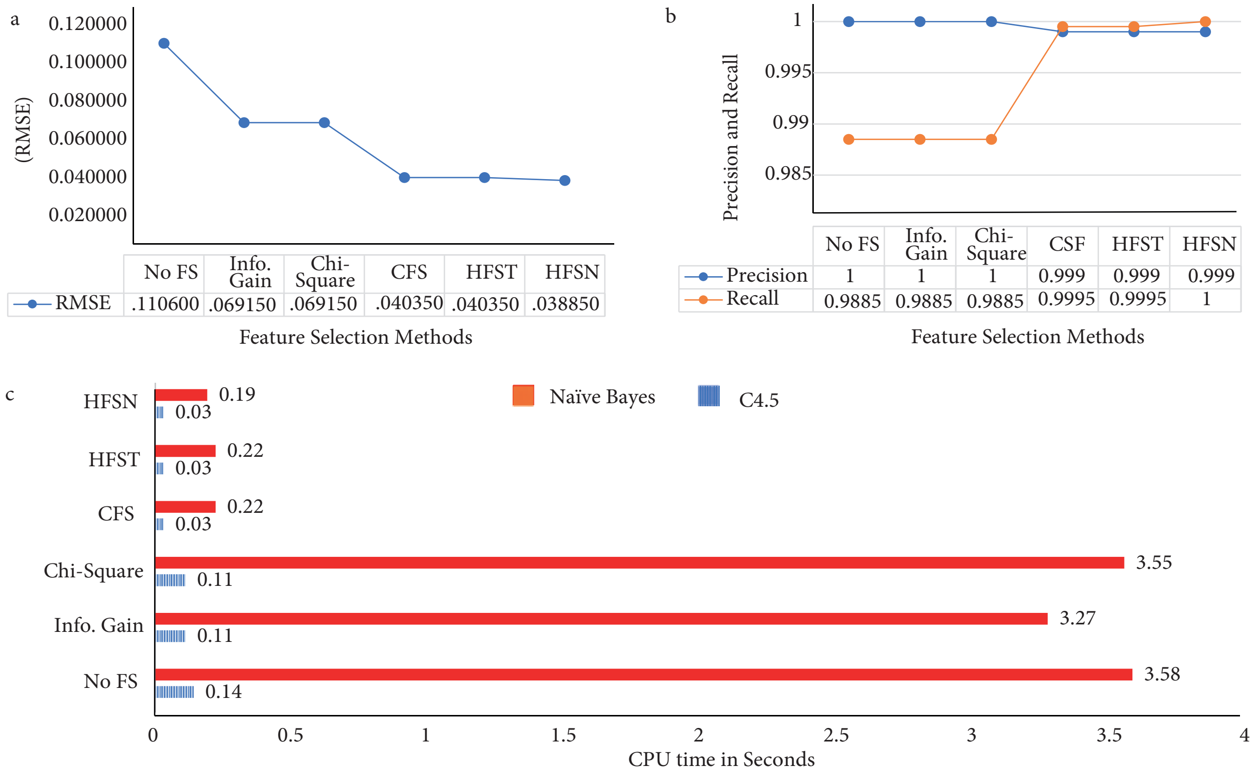

Figure 4. (a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the CICID2017 dataset.

6. Discussion

One of the main goals of those experiments was to attempt to find a way to select relevant features for web attack evidence, and indeed, to result in high accuracy and high performance in prediction models. These results support the outperformance of our hybrid feature selection approach against the other feature selection methods. The hybrid approach selected features that achieved greater accuracy and high performance with a lower error rate. The hybrid approach enhanced the recall by approximately 8% for ECML/PKDD 2007 and 20% for CISC2010 until it reached the peak value in CICID2017. This result of recall may be explained as the proportion of detected attacks from cases that were actually attacks, although another explanation for this is that the most relevant results were returned with a low false negative. Another important finding in this result is that increasing in recall was associated with the decreasing of precision. Thus, this could lead to less missing evidence in digital forensic investigations, as confirmed by [39, 41].

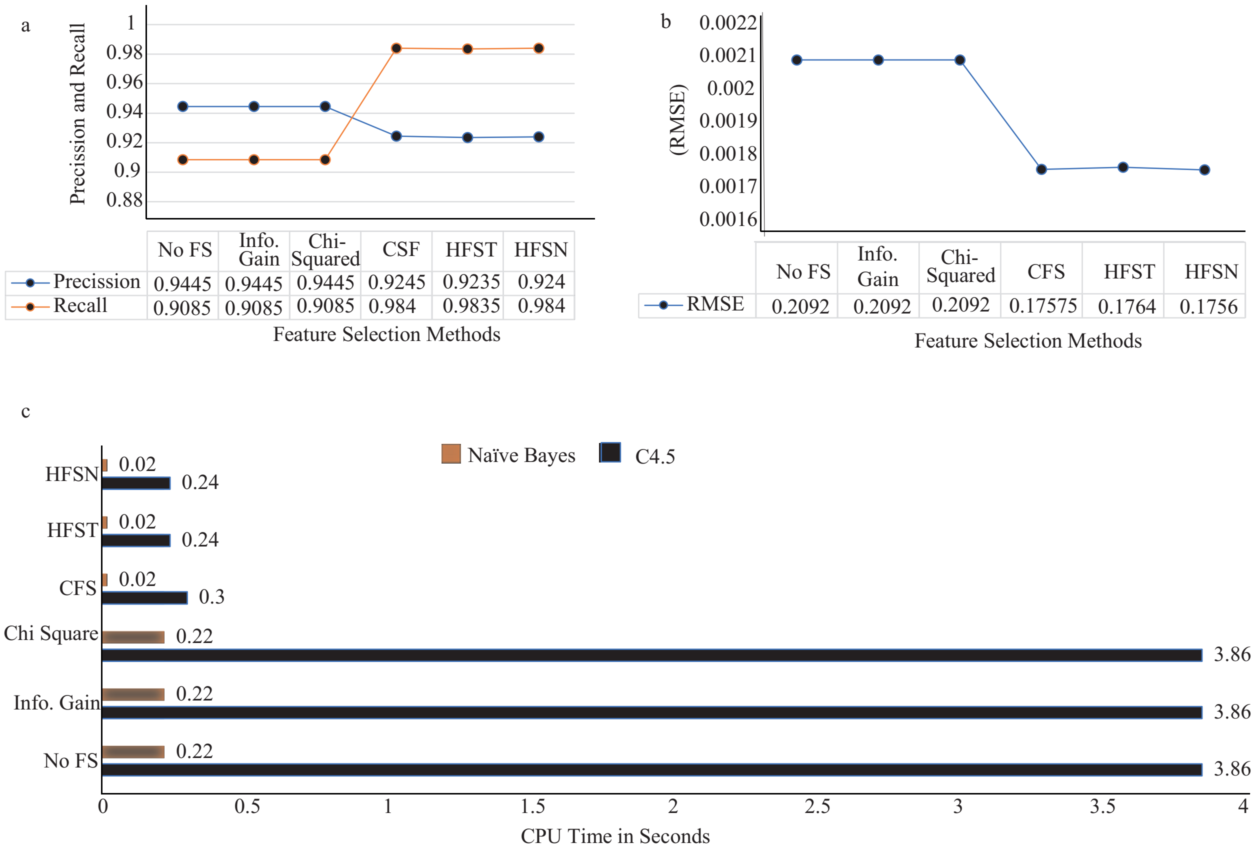

Figure 5. (a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the ECML/PKDD 2007 dataset.

Table 3 compares the proposed study with the related works which use the same datasets in terms of accuracy and the techniques they are based on. Nguyen et al. [42] showed that minimal-redundancy-maximal-relevance (mRMR) was successful in removing 80% of the irrelevant features from the ECML/PKDD 2007 dataset with a 4.2% reduction in accuracy. This differs from the findings presented here; in particular, our hybrid approach removed almost 90% of the irrelevant features. At the same time, the accuracy improved by just over 5% in the ECML/PKDD 2007 dataset. In addition, 90% of the features were reduced in the CSIC 2010 dataset with a slight increase in accuracy of 0.15% with the HFSN, differing from the HFST, where accuracy decreased by 0.6%. Similarly, this decrease was in agreement with those obtained by Nguyen et al. [28], whereas the accuracy of C4.5 decreased by 0.4% with CFS feature selection. In contrast to [43–45], the overall accuracy of our preprocessed CSIC 2010 dataset was found to be 30%, lower than that previously reported. There are 2 likely causes for the low accuracy. First, the preprocessing phase removed the original features of the dataset that support differentiation of the normal and attack traffic. The original features could lead to higher accuracy with overfitting as confirmed by Atienza et al. [29]. Second, the newly constructed features increased the number of features by 79%, which resulted in an accuracy drop.

Remarkably, the hybrid approach was successful in reducing 97% of the irrelevant features from the CICID2017 dataset, with a 1% increase in accuracy. The hybrid approach returned only 2 features, namely FWD IAT MIN and Init_Win_bytes_backward. It seems possible that these results are due to the differences between malicious traffic and normal traffic in forward packet timing and forward packet size. Another reason is

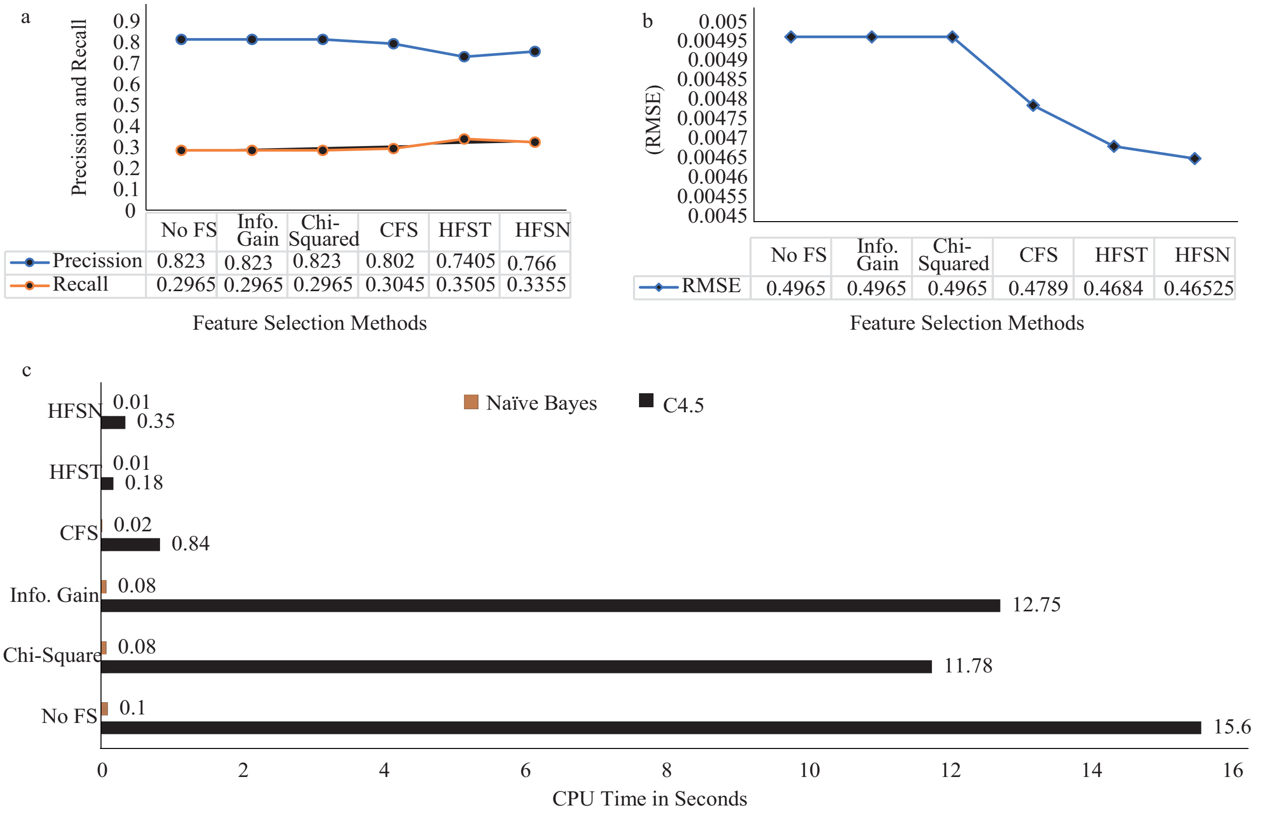

Figure 6. (a) Precision and recall; (b) RMSE; (c) CPU time of the feature selection methods in the CSIC 2010 dataset.

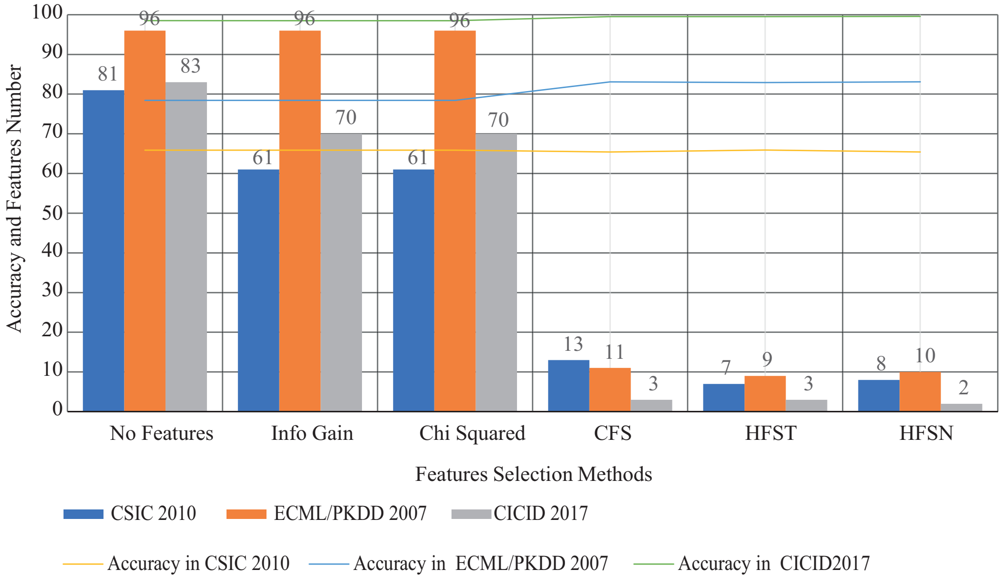

Figure 7. The average classification accuracy against the number of features.

that the first feature, which is related to the time between 2 packets, is sent in forward, while the second feature, which is related to the total number of bytes, is sent in the initial window. These findings suggest that timing could be an important issue for future research related to detection of encrypted web attacks or detection of web attacks. In accordance with the present results, FWD IAT MIN and Init_Win_bytes_backward were selected

Table 3. Comparison of increasing accuracy and reduction of features between the proposed study and related studies.

Dataset	Our Approach HFSN		Nguyen et al. [28] mRMR		Nguyen et al. [28] CFS		Torrano et al [30] n-gram+CFS	
	Accuracy +/-	FR	Accuracy +/-	FR	Accuracy +/-	FR	Accuracy +/-	FR
CICID2017	+1%	97%	-	-	-	-	-	-
ECML/PKDD 2007	+5%	90%	–4%	80%	–10%	93%	-	-
CSIC 2010	+0.15%	90%	–19%	53%	–0.12%	63%	–6%	90%

by Iman et al. [40] among 4 features considered as related to web attacks in the CICID2017 dataset. When it comes to comparing the hybrid approach and filter methods, especially CFS, they performed very similar in the ECML/PKDD 2007 and CICID2017 datasets; however, in CSIC 2010 the hybrid approach had superiority over the CFS method. Overall, the experimental results showed that the hybrid approach can achieve a better accuracy rate and satisfaction performance. The hybrid approach improved the recall with a slight decrease in error compared to the filter methods. We only used a limited number of classification algorithms as a basis, whereas a greater number of algorithms could lead to a higher generalization of our results. Moreover, a higher model of n-gram could generate more features that are more relevant and could enhance accuracy. A greater understanding of our findings could lead to a theoretical improvement in an automated investigation for web application attacks and could speed up the process of the attack investigation.

7. Conclusion

The benefit of using a hybrid approach is the superiority of filter methods over wrapper methods in terms of accuracy and performance. We have obtained accurate and comprehensive results proving that combining filter methods with wrapper methods could enhance the recall, performance, and accuracy of the digital evidence of web application attacks. Considerable insight has been gained with regards to relevant features for web attacks in both upper- and lower-level network traffic. However, more definite conclusions will be possible when a higher model of n-gram is applied, or more effective expert knowledge rules are developed in the feature construction process. Indeed, that will increase the number of features, but it could let the feature selection methods provide much more interesting results in finding digital evidence.

The present study is the first to investigate the in web application forensics. Indeed, feature selection methods are widely considered to be the most important step in machine learning. Our work has led us to conclude that the hybrid approach is the most effective feature selection method in terms of accuracy and stability. This approach has the potential requirements to be considered as a forensic technique in finding digital evidence of web attacks. Therefore, using the hybrid approach for building machine learning and data mining models could be a useful aid for intelligent automation system for web attack forensics.

The proposed approach worked on log data, so it could easily be integrated into a live system. The log file could be manually transferred after an attack, but transferring log data automatically by syslog service should be preferred. The devices and web servers would be configured to send logs to the predefined syslog server IP address. A Linux server, which runs a syslog deamon, would be enough. The syslog daemon would be configured to listen for messages from remote devices and do possible filtering if needed. Our implementation could run after an event or could be configured to do some regular checks on the collected log data. Future work will concentrate on building machine learning models for web attack forensics. Moreover, we are currently in the process of investigating an automated method to transfer and protect digital evidence after extraction. As data increases in size, it requires a higher level of investigation. The next decade is likely to witness a considerable rise in automated digital forensic techniques, which are based on machine learning and data mining. At that time, finding digital evidence will become easier and faster.

References

[1] Tully G. Codes of Practice and Conduct for Forensic Science Providers and Practitioner in the Criminal Justice System. UK: The Forensic Science Regulator, 2017.

[2] Palmer G. A road map for digital forensic research. In: Proceedings of the 1st Digital Forensics Research Workshop Conference; Utica, NY, USA; 2001. pp. 1-42.

[3] James JI, Gladyshev P. Challenges with automation in digital forensic investigations. arXiv: Computers and Society, 2013.

[4] Katzir Z, Elovici Y. Quantifying the resilience of machine learning classifiers used for cyber security. Expert Systems with Applications 2018; 92: 419-429.

[5] Dong Y, Zhang Y, Ma H, Wu Q, Liu Q et al. An adaptive system for detecting malicious queries in web attacks. Science China Information Sciences 2018; 61 (3): 32-114.

[6] Santhosh RP, Silambarasan G, Scholar MP. Role of data mining in cyber security. International Journal of Engineering Science and Computing 2017; 7 (7): 13932.

[7] Tianfield H. Data mining based cyber-attack detection. System Simulation Technology 2017; 13 (2). 90-104.

[8] McWhirter PR, Kifayat K, Shi Q, Askwith B. SQL injection attack classification through the feature extraction of SQL query strings using a gap-weighted string subsequence kernel. Journal of Information Security and Applications 2018; 40: 199-216.

[9] Zamani M, Movahedi M. Machine learning techniques for intrusion detection. arXiv:Cryptography and Security, 2013.

[10] Pu W, Jun-Qing W. Intrusion detection system with the data mining technologies. In: IEEE 2001 3rd International Conference on Communication Software and Networks; Xi'an, China; 2011. pp. 490-492.

[11] Choras M, Kozik R. Machine learning techniques applied to detect cyber attacks on web applications. Logic Journal of the IGPL 2014; 23 (1): 45-56.

[12] Goodison SE, Davis RC, Jackson BA. Digital evidence and the U.S. criminal justice system: identifying technology and other needs to more effectively acquire and utilize digital evidence. USA: RAND Corporation, 2015.

[13] Ashcroft J. A guide for first responders. USA: United States Department of Justice Off. Justice, 2001.

[14] Pichan A, Lazarescu M, Soh ST. Towards a practical cloud forensics logging framework. Journal of Information Security and Applications 2018; 42: 18-28.

[15] Šuteva N, Mileva A, Loleski M. Finding forensic evidence for several web attacks. International Journal of Internet Technology and Secured Transactions 2015; 6 (1): 64.

[16] Hraiz S. Challenges of digital forensic investigation in cloud computing. In: ICIT 2017 8th International Conference on Information Technology; Amman, Jordan; 2017. pp. 568-571.

[17] Kyaw AK, Sioquim F, Joseph J. Dictionary attack on wordpress: security and forensic analysis. In: 2015 2nd International Conference on Information Security and Cyber Forensics, InfoSec; Cape Town, South Africa; 2015. pp. 158-164.

[18] Khobragade PK, Malik LG. Data generation and analysis for digital forensic application using data mining. In: 2014 4th International Conference on Communication Systems and Network Technologies; Bhopal, India; 2014. pp. 458-462.

[19] Meyer R. Detecting Attacks on Web Applications from Log-files. USA: SANS Institute, 2008.

[20] Seyvar BM, Catak FO, Gul E. Detection of attack-targeted scans from the apache HTTP Server access logs. Applied Computing and Informatics 2018; 14: 28-36.

[21] Lu Q, Li X, Dong Y. Structure preserving unsupervised feature selection. Neurocomputing 2018; 301: 36-45.

[22] Guyon I. An introduction to variable and feature selection. Journal of Machine Learning Research 2003; 3: 1157-1182.

[23] Cai J, Luo J, Wang S, Yang S. Feature selection in machine learning: a new perspective. Neurocomputing 2018; 300: 70-79.

[24] Hu L, Gao W, Zhao K, Zhang P, Wang F. Feature selection considering two types of feature relevancy and feature interdependency. Expert Systems with Applications 2018; 93: 423-434.

[25] Hidayah MSN, Faizal MA, Selamat SR, Fadhlee MDR, Ramzi WYWA. Revealing the feature influence in HTTP botnet detection. International Journal of Communication Networks and Information Security 2017; 9 (2): 274-281.

[26] Kruegel C, Vigna G, Robertson W. A multi-model approach to the detection of web-based attacks. Computer Networks 2005; 48 (5): 717-738.

[27] Robertson W, Vigna G, Kruegel C, Kemmerer R. Using generalization and characterization techniques in the anomaly-based detection of web attacks. In: Proceedings of the 13th Symposium on Network and Distributed System Security; San Diego, CA, USA; 2006. pp. 15.

[28] Nguyen HT, Torrano-Gimenez C, Alvarez G, Petrović S, Franke K. Application of the generic feature selection measure in detection of web attacks. Lecture Notes in Computer Science. Berlin, Heidelberg, Germany: Springer, 2011.

[29] Atienza D, Herrero Á, Corchado E. Neural analysis of HTTP traffic for web attack detection. Advances in Intelligent Systems and Computing 2015; 369: 201-212.

[30] Torrano-Gimenez C, Nguyen HT, Alvarez G, Petrovic S, Franke K. Applying feature selection to payload-based web application firewalls. In: Proceedings of the 3rd International Workshop on Security and Communication Networks; Gjovik, Norway; 2011. pp. 75-81.

[31] Torrano-Gimenez C, Nguyen HT, Alvarez G, Franke K. Combining expert knowledge with automatic feature extraction for reliable web attack detection. Security and Communication Networks 2015; 8 (16): 2750-2767.

[32] Zhang Z, George R, Shujaee K. Efficient detection of anomolous HTTP payloads in networks. In: Conference Proceedings of IEEE SOUTHEASTCON; Norfolk, VA, USA; 2016. pp. 1-3.

[33] Choi JH, Choi C, Ko BK, Kim PK. Detection of cross site scripting attack in wireless networks using n-Gram and SVM. Mobile Information Systems 2012; 8 (3): 275-286.

[34] Wressnegger C, Schwenk G, Arp D, Rieck K. A close look on n-grams in intrusion detection. In: Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security AISec13; Berlin, Germany; 2013. pp. 67-76.

[35] Nascimento G, Correia M. Anomaly-based intrusion detection in software as a service. In: Proceedings of the International Conference on Dependable Systems and Networks; Hong Kong; 2011. pp. 19-24.

[36] Torrano Giménez C, Pérez Villegas A GÁM. Http dataset CSIC 2010. Spain: Information Security Institute of CSIC Spanish Research National Council, 2010.

[37] Gallagher B, Eliassi-Rad T. Classification of HTTP Attacks: A Study on the ECML/PKDD 2007 Discovery Challenge. USA: U.S. Department of Energy Lawrence National Laboratory, 2008.

[38] Habibi Lashkari A, Draper Gil G, Mamun MSI, Ghorbani AA. Characterization of tor traffic using time based features. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy; Porto, Portugal; 2017. pp. 253-262.

[39] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence; Chambery, France; 1993. pp. 1022-1027.

[40] Sharafaldin I, Habibi Lashkari A, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy; Portugal; 2018. pp. 108-116.

[41] Hall M. Correlation-based feature selection for machine learning. PhD, University of Waikato, Hamilton, New Zealand, 1999.

[42] Nguyen HT, Torrano-Gimenez C, Aalvarez G, Franke K, Petrović S. Enhancing the effectiveness of web application firewalls by generic feature selection. Logic Journal of the IGPL 2013; 21 (4): 560-570.

[43] Althubiti S, Yuan X, Esterline A. Analyzing HTTP requests for web intrusion detection. In: 2017 KSU Conference on Cybersecurity Education Research and Practice; Kennesaw State University, GA, USA; 2017.

[44] Pham TS, Hoang TH, Vu VC. Machine learning techniques for web intrusion detection-a comparison. In: 2016 8th International Conference on Knowledge and Systems Engineering, KSE 2016; Hanoi, Vietnam; 2016. pp. 291-297.

[45] Rafaland K, Choras M. A Proposal of algorithm for web applications cyber attack detection. In: IFIP International Conference on Computer Information Systems and Industrial Management; Ho Chi Minh City, Vietnam; 2014. pp. 680-687.