Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

10th International Conference
on Electrical Engineering
ICEENG 2016

Spyware Detection by Extracting and Selecting Features in Executable Files

By

Mohamed Adel Sheta [*], Mohamed Zaki [§], Kamel Abd El Salam El Hadad [‡], and H. Aboelseoud M [†]

Abstract:

Spyware detection techniques have been presented using three approaches; signature-based, behavior-based, and specification-based. These approaches failed in detecting new spyware. Data mining is a new approach in detecting spyware that has the ability to detect new spyware or mutated effects of existing spyware. The main challenges in designing anti-spyware systems using data mining techniques are in extracting and selecting the most strong and significant features from spyware data set. In this paper a new approach of extracting and selecting features is proposed. In this approach, the unique features are extracted from all executable files in each class type. Then the selection of the strongest features is done based on the occurrence or the frequency of the features in the data set. The experimental results of the proposed approach outperform all the previous competing approaches.

Keywords:

Spyware, data mining, feature extraction, and feature selection.

---

* Ph.D. Student, Department of Computer Engineering, Military Technical College, Egypt. mohshe2007@hotmail.com.
§ Prof. of Computer and System Engineering, Al-Azhar University, Egypt. azhar@eun.eg.
‡ Dr., Department of Computer Engineering, Military Technical College, Egypt. kamel_elhadad@hotmail.com.
† Dr., Department of Computer Engineering, Military Technical College, Egypt. h_aboelsoud@yahoo.com.

1. Introduction:

Spyware is deemed a big danger to the privacy and secrecy of both individuals and firms; it frequently results in the damage and loss of electronic data stored on all types of computers. The way spyware operates is as follows; data is stolen from a computer without the authorization of the original owner, and then it is sent to a third party for illegal use. To create an anti-spyware system, firstly, we should have sufficient background of the feature types, analysis and detection techniques; this will be discussed in the following sections.

1.1 Feature Types:

The input data given to any spyware-detection system is called a feature type. The provided data acts as the main criterion for classification for the majority of spyware detection systems. If the data is gathered from a workstation, the detection type is labeled a "host-based detection;" however, if the source of data is a network, then the detection in this case is called a "network-based detection." The different feature types existent are byte sequences, application programming interfaces (API), call sequences, instruction sequences, and network connection records. All these types will be discussed below.

1.1.1 N-grams:

N-grams are defined as sequences of bytes of either fixed or variable lengths, which are extracted from the hexadecimal dump, which is the raw computer data as seen in the memory of an executable program.

1.1.2 API/System Calls:

Application programming interfaces (API) are source code interfaces that both libraries and operating systems provide to carry out orders given to it by computer programs. For these operating system orders or requests, API is alternatively named "system calls." These call sequences monitor the activity of all programs, which makes them the prime method for detecting any dangerous or malicious behavior.

1.1.3 Assembly Instructions:

In later and newer developments, and because n-grams fail to capture program semantics, instruction sequences should be utilized instead. Unlike extracting n-grams from hexadecimal dumps, instructions need to be disassembling from their original binary forms.

1.1.4 Network Connection Records:

A network connection record is defined as a set of information, like the number of transmitted bytes, duration, protocol type, amongst others, which themselves represent a sequence of data flow to and from a well defined source and target.

1.2 Analysis Techniques:

The analysis can be executed either at the source code or at the binary levels, where only the executable program is available. It is incorrect to say that source code is available for every program. The only option remaining is to use executables [1]. The information collected from these executables can be gathered either by running them or performing static reverse engineering or by using both techniques. On the other hand, the counter engineering method is called static analysis, in which the process where the program executed to gather the information is called dynamic analysis. An approach which utilizes a mixture of both methods is also sometimes used.

1.2.1 Static Analysis:

The static analysis approach provides information about programs control, data flow and other statistical features without actually requiring running the program. Many alternate reverse engineering methods are occasionally used to formulate an intermediate binary-code representation. As soon as the data changes into human readable format, other techniques can be applied for further analysis. The most notable advantage that static analysis provides ahead of its dynamic counter-method is that it is time-saving because it does not require any execution time. However, the major disadvantage lies ultimately in the fact that any of the static reverse engineering methods are merely approximations of the actual execution.

1.2.2 Dynamic Analysis:

Dynamic analysis on the other hand, actually executes the program and then keeps track of the execution in either a real or a virtual machine environment. Even though this provides the actual information about both the control and data flow, this method also requires a substantial amount of execution time. The hybrid approach is usually used; it runs selected parts of the code when and if the static analysis approach does not succeed at making the correct decision.

1.3 Detection Techniques:

Spyware detection techniques are used to monitor and detect different types of spyware and hence preventing the infection of computer systems. These techniques can be categorized into either one of four types of detection; signature based detection, behavior based detection, specification based detection, and data mining based detection; [2] these types will be discussed in depth in the coming sections.

1.3.1 Signature Based Detection:

The first method, signature based detection detects spyware by comparing the spyware signature to the database. These signatures are formed by examining the disassembled binary code of the spyware. This disassembled form of code is analyzed, and from then its features are extracted. These same features are used to then construct the signature of a certain spyware family [3]. The major advantages of this technique of detection are its high degree of accuracy of detection of the known spyware and also its lower usage of resources that are needed to detect the spyware. This is mainly because it focuses on the signatures of attack. The main disadvantage however, is its inability to detect the new spyware due to the unavailability for signatures for this new type of spyware [4].

1.3.2 Behavior Based Detection:

The main function of this type of detection, behavior based detection, is to analyze the behavior of either known or unknown spyware. It often happens in one of two phases: the first being the training phase and the latter, the detection phase. During the training phase, the behavior of the system in the ideal state is observed and then the machine-learning technique is used to create a profile of such normal behavior. The detection phase is the comparison of the current system behavior after attack to the normal behavior. The differences between both behaviors are marked as potential threats [5]. The most important advantage of this detection technique is that it is able to find and locate both known and unknown or new instances of spyware. However, the main disadvantage of this technique of detection is that it needs to update the data that describes the system behavior on a consistent basis. It needs many resources such as CPU time, memory and disk space, as well as its suffering from a high level of false positive rate [6].

1.3.3 Specification Based Detection:

The third type, specification based detection, comes from behavior based detection as it attempts to overcome the high false positive rate that comes with it. The process of

monitoring programs is involved in executions and detecting the degree of difference from the behavior from the specification, rather than detecting the occurrence of specific threat patterns [7]. The main advantage of this technique is similar to the previous behavior based detection; it can detect known as well as new or unknown instances of spyware and the level of false positive rate is low. The main disadvantages of this technique are its high level of false negative rate, and this makes it less effective as the behavior based method of detection in detecting new attacks. In addition to this, the development of detailed specification is highly time consumption.

1.3.4 Data Mining Based Detection:

Data mining is the main focus of many researchers in the field of new and unknown spyware detection. They have proposed a fourth method of detection known as data mining based detection [2]. Data mining mainly helps in analyzing data using automated statistical analysis techniques, by identifying meaningful patterns or correlations. The results from this method of analysis can then be summarized into information that is useful and can be used for prediction. Machine learning algorithms are used for pattern detection or relations in data which are then used to build a classifier. Data mining is capable of detecting new or unknown spyware with high detection rate compared to signature, behavior, and specification based detection methods.

This paper is organized as follows; related work is discussed in section 2. The proposed anti-spyware system design is explained in section 3. The experimental results and conclusions are discussed in section 4 and section 5 respectively.

2. Related Work:

Matthew G. S. et al. [8] presented a method for detecting previously undetectable malicious executables. They showed this by comparing their results with traditional signature based methods and with other learning algorithms. They used a data set consisted of a total of 4266 programs split into 3265 malicious binaries and 1001 clean programs. The Multi-Naive Bayes with byte sequence method had the highest accuracy and detection rate of any algorithm over unknown programs, 97.76%, over double the detection rates of signature based methods. Naive Bayes algorithm with strings had the highest overall accuracy using the, 97.11%.

Raja K. et al. [9] detected spyware by using data mining method with a byte sequence mining approach. The experiment applied on low amount of data set contained 137 files. Out of these, 18 files were spyware and 119 files were benign. Feature sets generated by Common Feature Based Extraction (CFBE) selection method produced better results than feature sets generated by Frequency Based Feature Extraction (FBFE).

The overall Accuracy (ACC) was 90.5% and the Area under Receiver Operating Characteristic Curve (AUC) score of 0.83.

Ivan Firdausi et al. [10] used behavior based detection method combined with machine learning algorithm in detecting malware. They applied their algorithms on a data set of 220 malware and 250 benign software samples. They used four algorithms depend on binary with no feature selection, frequency with no feature selection, binary with feature selection, and frequency with feature selection. They used in their experiments five types of machine learning algorithms. The overall best performance was achieved by decision tree based learning algorithm J48 using the term frequency weight without feature selection data set, with a true positive rate (TPR) of 95.9%, a false positive rate (FPR) of 2.4%, and an accuracy of 96.8%.

Raja K. et al. [11] detected adware by using data mining method with an opcode sequence extraction mining approach. The experiment applied on a data set contained 600 files. Out of these, 300 files were spyware and 300 files were benign and AUC was equal to 0.949.

Donghwi Lee et al. [12] classified malicious codes by extracting hexadecimal values and calculating the frequency for a specific n-gram. They calculated the similarity for each malware family using vector space model.

Raja K. and Niklas L. [13] detected scareware by using data mining method with a variable length instruction sequence mining approach. The experiment applied on a data set contained 800 files. Out of these, 550 files were scareware and 250 files were benign .The results were AUC equal to 0.972 and low false negative rate of 2.3%.

Asaf Shabtai et al. [14] detected unknown malware by applying classification techniques on opcode n-grams patterns as features. Opcode n-grams patterns of various size representations and eight types of classifiers were evaluated. The experiment applied on a data set contained 26093 files. Out of these, 5677 files were malware and 20416 files were benign. Evaluation results indicate that the evaluated methodology achieves a level of accuracy higher than 96% with TPR above 0.95 and FPR approximately 0.1.

Zongqu Z. et al. [15] detected malware by using data mining method based on the control flow of software mining approach. The experiment applied on a data set contained 9398 files. Out of these, 4828 files were malware and 4570 files were benign. The results were ACC equal to 97%, AUC score of 0.993 and low false positive rate equal to 3.2%.

Leena T. P. et al. [16] extracted features by byte sequence n-grams method from executable spyware and benign files where feature selection by CFBE and FBFE. They used decision tree classifier to classify the input files to their families. Decision tree produces some roles that will be used in building the anti-spyware system.

3. The Proposed Anti-spyware System Design:

In this section, the proposed anti-spyware system design is introduced. The proposed design depends on the feature extraction and feature selection methods. The proposed design gets most strong and most significant features from the used spyware data set. Section 3.1 introduced the block diagram of the proposed approach. Data set generation will be discussed in section 3.2. Feature extraction and feature selection are discussed in section 3.3 and 3.4 respectively.

3.1 The Proposed Block Diagram:

Figure (1) shows the proposed anti-spyware system block diagram that uses the data mining approach. It consists of two main parts; classifier building (offline) and classifier execution (online). The left part (classifier building) extracts and selects the most significant features that generate the training data set used in building classifier. The right part (classifier execution) uses the same extraction and selection methods on the scanned file. The proposed approach takes a decision using classifier on the scanned file to decide that it is a spyware or benign and also its spyware type.

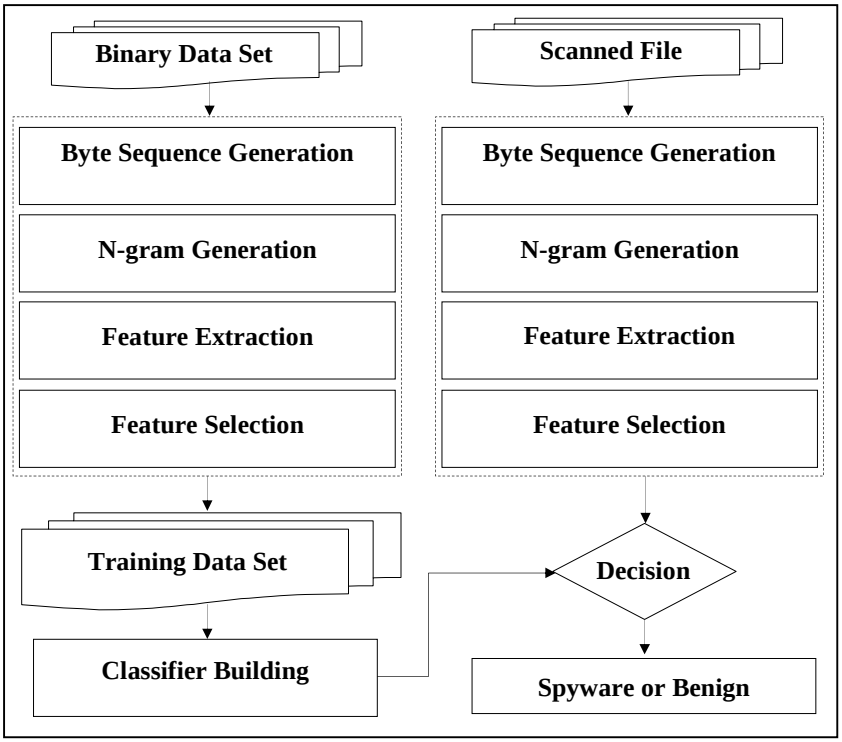


Figure (1): *The proposed anti-spyware system block diagram*

3.2 Data Set Generation:

This section introduces the steps of generating the data set used in our new approach.

3.2.1 Data Collection:

Binary spyware data set is collected from VX Heavens website [17]. Binary benign data set is collected from system directory of a fresh copy and spyware free guaranteed windows7, 32-bit operating system.

3.2.2 Balanced Data Set:

Our data set consists of 310 binary files out of which 270 benign and 40 spyware. As shown in Table (1) we used a balanced data set in number and size. The Malware File Percentage (MFP) is needed to be less or equal 15% of the total population in order to yield a high prediction performance [14] as shown in Figure (2) (balanced in number) and Figure (3) (balanced in size).

***Table (1):** Distribution of files in the data set*

Type	No of files	Size of files (MB)
SPY_AGENT	8	2.08
SPY _BHO	7	2
SPY_BANKER	6	2.54
SPY_KEYLOGGER	8	2.02
SPY_WEBMONER	5	2.20
SPY_WINSPY	6	1.98
Total (Spyware)	40	12.82
BENIGN	270	86.40
Total Percentage	**14.8%**	**14.8%**

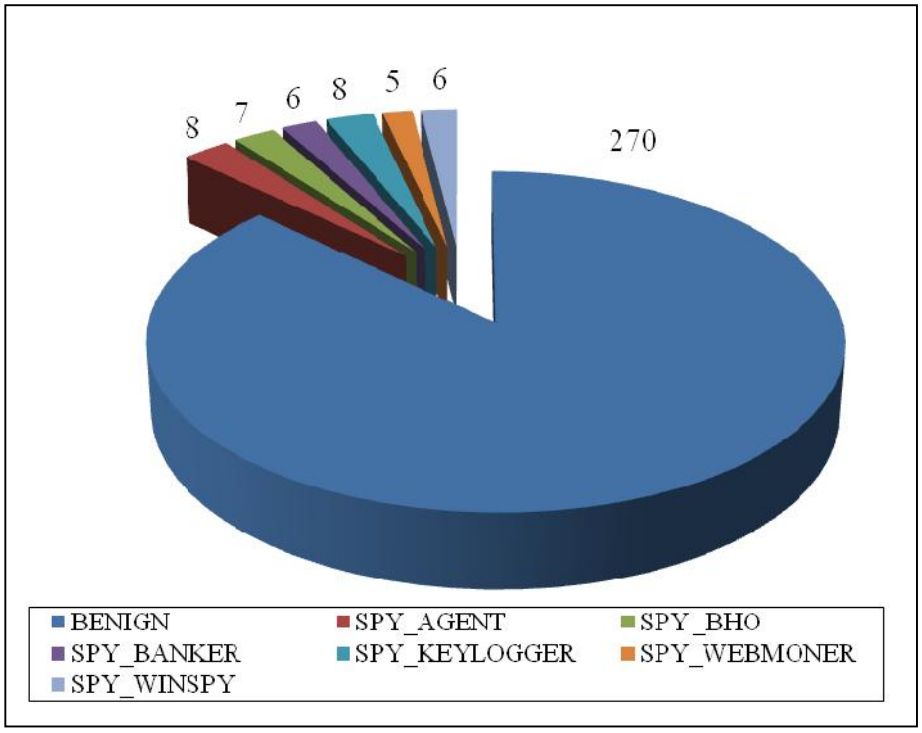


Figure (2): *Number of files distribution in the data set*

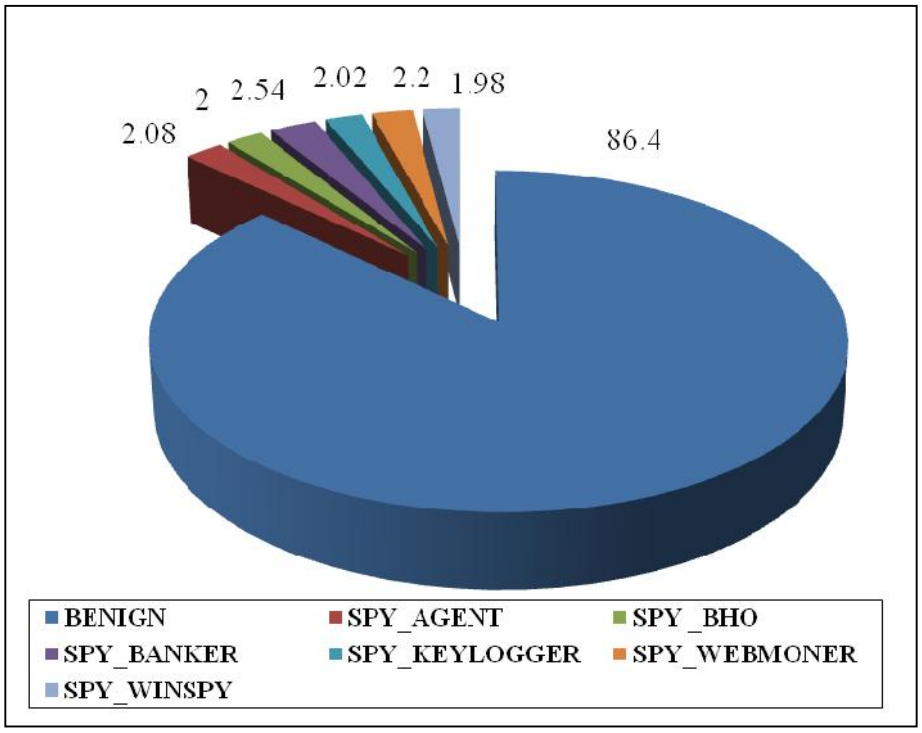


Figure (3): *Size of files (MB) distribution in the data set*

3.2.3 N-grams Generation:

In the n-grams generation method, the data set is converted into hexadecimal format (byte sequence). The generation method extracts byte sequences of the desired n-size. Each row contains one n-gram and the length of a single row is thus equal to the size of n. Previous researches have shown that n-grams of size 5 produced the most overall accurate results [13] [14]. In our study we suggested to use the range of n-grams 4, 5, and 6 to compare the performance of our algorithms in each of them.

3.3 Feature Extraction:

In this section, the feature extraction approach in the proposed system will be discussed in details with its steps as shown below. The first column in next tables is the type of spyware types or benign. The other three columns are the number of features for each type with n-gram equal to 4, 5, and 6.

3.3.1 Generating Features of Each Type:

Table (2) shows the number of features generated in each type of the used data set using n-gram equal to 4, 5, and 6. The numbers of occurrence of each feature that exist in all files of each type are presented in Table (2). The total number of features in this step is nearly between 9 and 13 million which is a huge number of features. So, in the next step the total number of features will be reduced.

***Table (2):** Number of features in each type*

Type	N-gram = 4	N-gram = 5	N-gram = 6
BENIGN	11,274,622	9,019,694	7,516,414
SPY_AGENT	271,862	217,490	181,241
SPY _BHO	262,400	209,920	174,933
SPY_BANKER	333,184	266,547	222,123
SPY_KEYLOGGER	263,936	211,148	175,957
SPY_WEBMONER	288,054	230,444	192,036
SPY_WINSPY	259,968	207,974	173,312
Total	**12,954,026**	**10,363,217**	**8,636,016**

3.3.2 Generating Distinct Features:

Table (3) introduced the distinct features regardless the number of occurrence of each feature in all files in each type. In this step the total number of features decreased to be nearly between 4 and 5 million.

***Table (3):** Number of distinct features in each type*

Type	N-gram = 4	N-gram = 5	N-gram = 6
BENIGN	3,043,166	4,112,403	4,360,495
SPY_AGENT	217,640	191,288	155,561
SPY _BHO	82,471	92,722	93,376
SPY_BANKER	299,153	245,885	207,627
SPY_KEYLOGGER	81,340	117,770	83,429
SPY_WEBMONER	145,096	157,943	141,521
SPY_WINSPY	66,193	82,992	86,512
Total	**3,935,059**	**5,001,003**	**5,128,521**

3.3.3 Generating Unique Features:

Table (4) shows the number of the unique features in each type that not exist in any other types. This step introduces the strongest features which can be considered as a unique gene for each type. The distribution of features in spyware families is nearly between 15% and 20% of the total number of the all features which is still a balanced dataset. Figure (4) shows the distribution of features in each type for n-gram = 5.

***Table (4):** Number of unique features in each type*

Type	N-gram = 4	N-gram = 5	N-gram = 6
BENIGN	2,959,998	4,046,441	4,330,061
SPY_AGENT	197,345	176,484	147,330
SPY _BHO	59,735	71,115	83,461
SPY_BANKER	288,577	238,972	202,972
SPY_KEYLOGGER	52,584	85,981	69,783
SPY_WEBMONER	108,182	126,482	124,972
SPY_WINSPY	43,571	63,893	74,635
Total	**3,709,992**	**4,809,368**	**5,033,214**

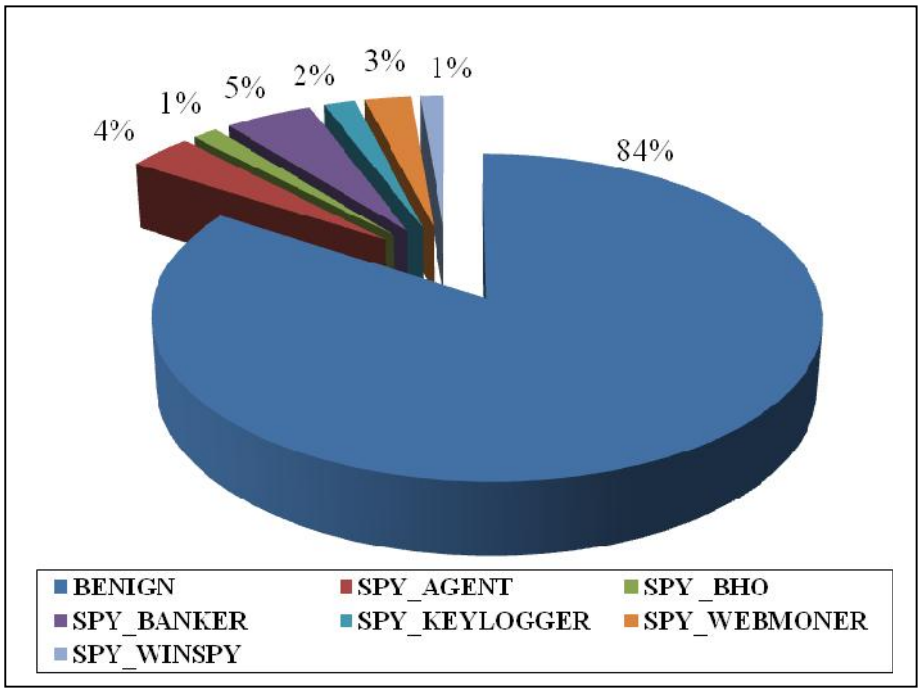


***Figure (4):** Features distribution in each type for n-gram = 5*

3.4 Feature Selection:

In this section, the feature selection approach in the proposed system will be discussed in details with its steps as shown below. Two different methods will be used in selecting features that will be discussed in the next sub section.

3.4.1 Methodology:

The next two methods are proposed to select the strongest features in each class type to introduce the training data set that will be used in the classifier building. The first method is Common Feature Based Extraction (CFBE) which selects the features of the highest occurrence in the files of each class type in the dataset. The second method is Frequency Based Feature Extraction (FBFE) which selects the features of the highest frequency in the files of each class type in the dataset.

3.4.2 Generating Attribute Relation File Format:

For each n-gram, two files in Attribute Relation File Format (ARFF) based on CFBE or FBFE were generated. Each of them has two columns; the feature column and the type column which will be considered as the class type during the operation of building the classifiers as shown in Table (5).

***Table (5):** Sample of ARFF files*

Feature	Type
9194C8C3	SPY_AGENT
E5FC6856	SPY_AGENT
00FFFC00	SPY_BANKER

4. Experimental Results:

The implementation setup and evaluation metrics are discussed in section 4.1 and 4.2 respectively. The analysis and discussion are introduced in section 4.3.

4.1 Implementation Setup:

Our implementation runs on Intel Core i7 with 6 GB of RAM. We use the Waikato Environment for Knowledge Analysis (WEKA) version 3.6.12 [18] to perform the experiments.

4.2 Evaluation Metrics:

From the response of classifiers relevant confusion matrices were created. The following four metrics define the members of the matrix: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).
TP: represents the correctly classified benign programs.
FP: represents the incorrectly classified spyware programs.
TN: represents the correctly classified spyware programs.
FN: represents the wrongly classified benign programs.

In this experiment, the performance of each classifier of competing approaches is evaluated by the true positive rate, false positive rate, overall accuracy, and area under receiver operating characteristic curve that are defined as follows:

True Positive Rate (TPR), the higher the better:

$$TPR=TP/(TP+FN) \quad (1)$$

False Positive Rate (FPR), the lower the better:

$$FPR=FP/(FP+TN) \quad (2)$$

The overall Accuracy (ACC), the higher the better:

$$ACC=(TP+TN)/(TP+TN+FP+FN) \quad (3)$$

Area Under Receiver Operating Characteristic Curve (AUC):

It is the tradeoff between TPR and FPR, the higher the better.

4.3 Analysis and Discussion:

Table (6), (7), and (8) show the ACC results for n-gram equal to 4, 5, and 6. In our experiments we used six different learning algorithms ZeroR, Naïve Bayes, C4.5 Decision Tree known as (J48), Support Vector Machines with the training algorithm known as (SMO), JRip, and Random Forest as candidates for our study.
In our experiments the two proposed methods are applied using three ranges 100, 200, and 300 of selected strongest features. These features are the strongest since they are the most common in CFBE, and the most frequent in FBFE. The range of the selected features depending on the features with the highest weight that calculated as,

Feature Weight (FW)=(Feature Size (FS))/(Total Files Size(TS)) (4)

Where the feature size (FS) is the size of the features in the file distribution of the class type and can be calculated as,

FS=Feature frequency×(n-gram) (5)

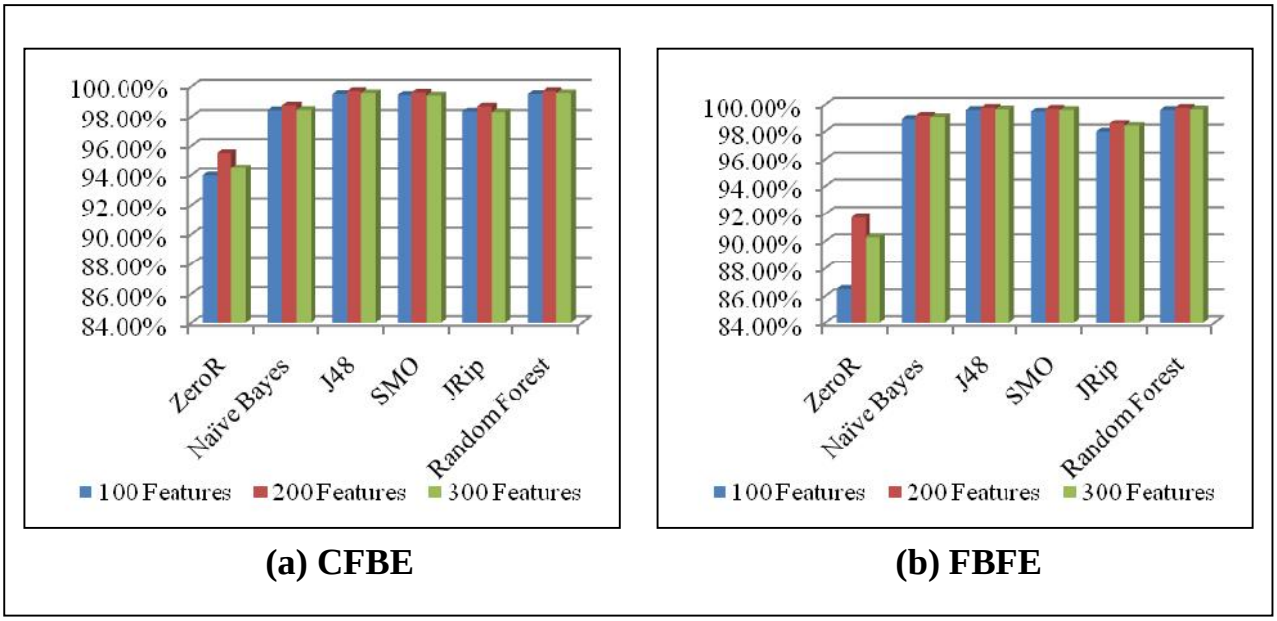
And the total files size (TS) is the size of all files in each class type and can be estimated as,

TS=∑ Files size (6)

Table (6), (7), and (8) show that there are peak values that specify the maximum number of chosen features depending on features weight in the data set. The peak values were at the strongest 200 features in n-gram equal to 4 and 6, while in n-gram = 5 the peak values exist at the strongest 100 features.
Table (7) shows that the results of FBFE applied to six different classifiers are the best results of these experiments. N-gram equal to 5, selecting the highest 100 features, and using J48 classifier resulting ACC equal to 99.86 % with reliable TPR equal to 99.8 %, FPR equal to 1.8 %, and AUC equal to 0.99. The charts in Figure (5), (6), and (7) show comparisons between the results using the different two methods explained in section 3.4 for n-gram equal 4, 5, and 6.
Table (9) shows a complete comparison in ACC between one of the previous studies [9] and our proposed approaches. We applied the previous methods on our data set due to the fact that most of the links used in this study provided by SpywareGuide.com were broken, i.e., these links did not lead to pages where the spyware executables could be downloaded.
The experimental results of the proposed approach outperform this previous competing approach in n-gram equal to 4, 5, and 6.

***Table (6):** ACC results of experiments for n-grams = 4*

The Method	Selected Features	ZeroR	Naïve Bayes	J48	SMO	JRip	Random Forest
CFBE	**100**	93.98 %	98.42 %	99.50 %	99.45 %	98.35 %	99.50 %
	200	95.53 %	98.74 %	**99.70 %**	99.60 %	98.68 %	99.70 %
	300	94.49 %	98.45 %	99.56 %	99.40 %	98.30 %	99.56 %
FBFE	**100**	86.51 %	98.96 %	99.61 %	99.50 %	97.98 %	99.61 %
	200	91.70 %	99.19 %	**99.76 %**	99.70 %	98.60 %	99.76 %
	300	90.27 %	99.10 %	99.65 %	99.60 %	98.40 %	99.65 %

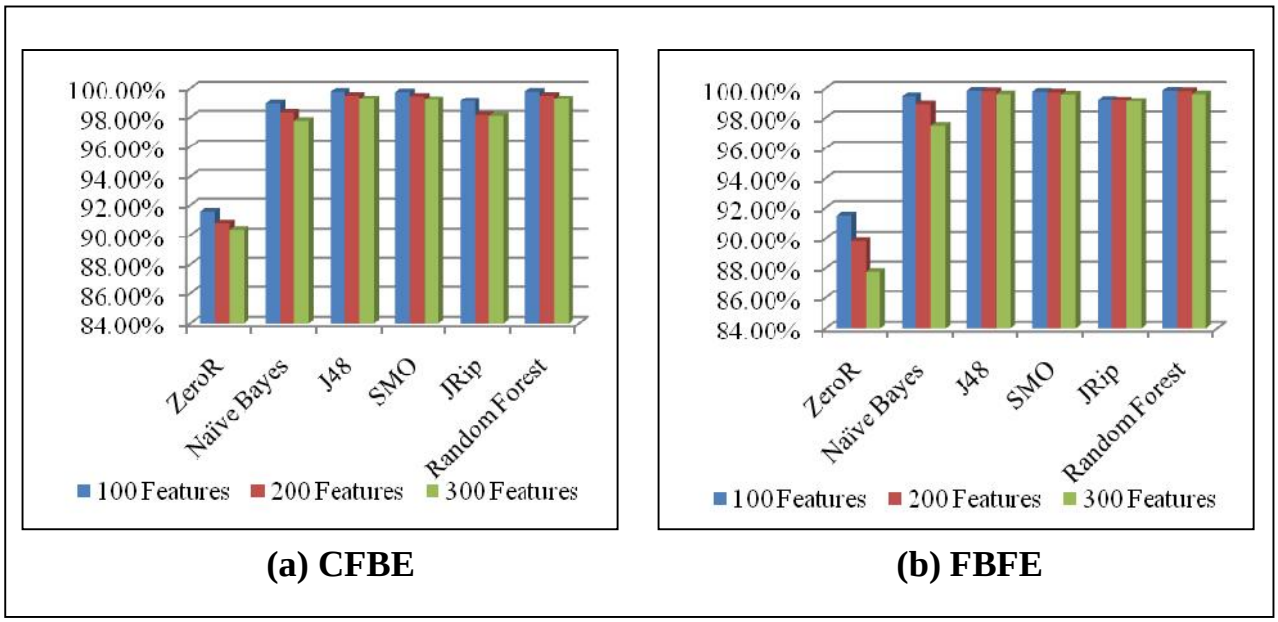


(a) CFBE **(b) FBFE**

***Figure (5):** ACC results of experiments analysis for n-grams = 4*

***Table (7):** ACC results of experiments for n-grams = 5*

The Method	Selected Features	ZeroR	Naïve Bayes	J48	SMO	JRip	Random Forest
CFBE	**100**	91.58 %	98.93 %	**99.79 %**	99.75 %	99.12 %	99.79 %
	200	90.80 %	98.31 %	99.50 %	99.45 %	98.16 %	99.50 %
	300	90.37 %	97.75 %	99.24 %	99.20 %	98.10 %	99.24 %
FBFE	**100**	91.50 %	99.49 %	**99.86 %**	99.80 %	99.24 %	99.86 %
	200	89.86 %	98.89 %	99.84 %	99.75 %	99.20 %	99.84 %
	300	87.76 %	97.50 %	99.63 %	99.60 %	99.12 %	99.63 %

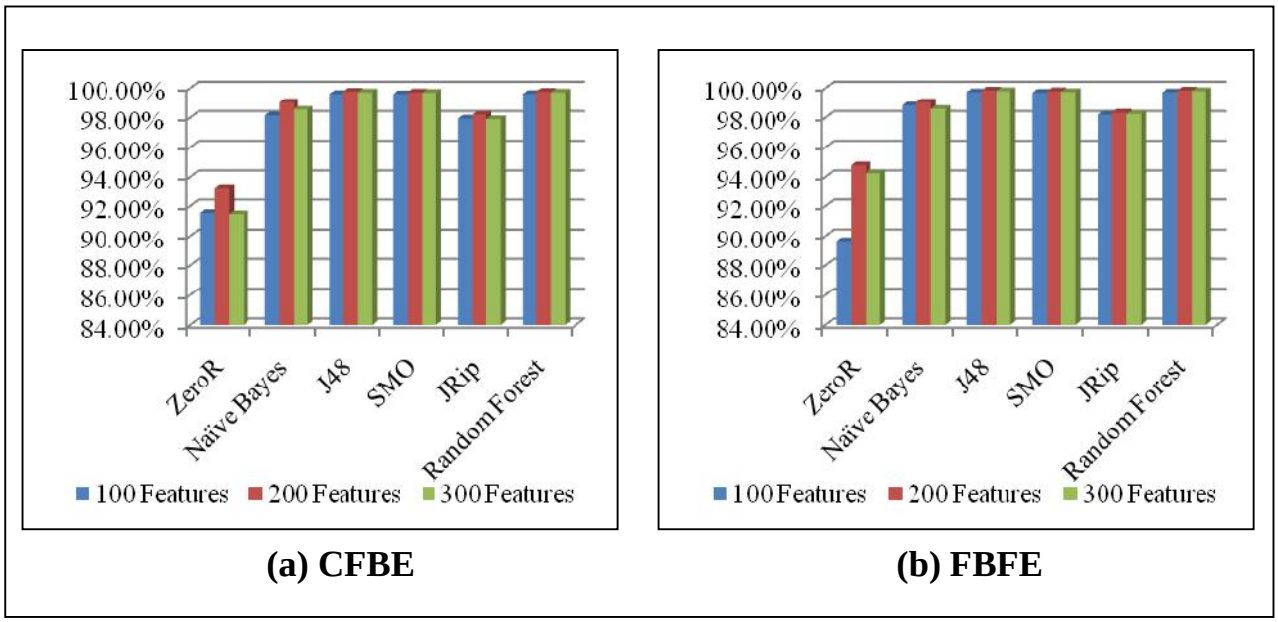


(a) CFBE **(b) FBFE**

***Figure (6):** ACC results of experiments analysis for n-grams = 5*

***Table (8):** ACC results of experiments for n-grams = 6*

The Method	Selected Features	ZeroR	Naïve Bayes	J48	SMO	JRip	Random Forest
CFBE	**100**	91.54 %	98.15 %	99.61 %	99.60 %	97.93 %	99.61 %
	200	93.26 %	99.04 %	**99.77 %**	99.72 %	98.20 %	99.77 %
	300	91.46 %	98.54 %	99.72 %	99.70 %	97.90 %	99.72 %
FBFE	**100**	89.66 %	98.83 %	99.73 %	99.70 %	98.20 %	99.73 %
	200	94.79 %	99.04 %	**99.84 %**	99.80 %	98.35 %	99.84 %
	300	94.26 %	98.59 %	99.81 %	99.75 %	98.25 %	99.81 %



(a) CFBE **(b) FBFE**

***Figure (7):** ACC results of experiments analysis for n-grams = 6*

***Table (9):** ACC comparison between the old methods and the proposed*

The Classifier	N-grams	CFBE		FBFE	
		Raja et. al., 2010 [9]	Proposed	Raja et. al., 2010 [9]	Proposed
ZeroR	**4**	86.25 %	95.53 %	90.13 %	91.70 %
	5	86.72 %	91.58 %	91.13 %	91.50 %
	6	85.54 %	93.26 %	92.79 %	94.79 %
Naïve Bayes	**4**	69.19 %	98.74 %	87.43 %	99.19 %
	5	67.29 %	98.93 %	95.13 %	99.49 %
	6	77.72 %	99.04 %	96.38 %	99.84 %
J48	**4**	86.25 %	99.70 %	94.74 %	99.76 %
	5	**86.72 %**	**99.79 %**	95.13 %	**99.86 %**
	6	85.54 %	99.61 %	97.84 %	99.84 %
SMO	**4**	83.41 %	99.60 %	96.50 %	98.60 %
	5	83.64 %	99.75 %	96.80 %	99.80 %
	6	82.70 %	99.72 %	97.19 %	99.80 %
JRip	**4**	86.25 %	98.68 %	94.73 %	98.60 %
	5	86.72 %	99.12 %	95.13 %	99.24 %
	6	85.78 %	98.20 %	**97.89 %**	98.35 %
Random Forest	**4**	82.46 %	99.70 %	92.31 %	99.76 %
	5	83.17 %	99.79 %	94.39 %	99.86 %
	6	81.75 %	99.77 %	95.44 %	99.84 %

5. Conclusions:

The main challenges in designing anti-spyware systems using data mining technique are extracting and selecting the most strong and significant features. New approaches of extracting and selecting features are proposed in this paper. The unique features are extracted and then selecting the strongest features based on the occurrence or the frequency of the features in the data set as introduced in the proposed approaches. The experimental results show that the ACC of the proposed approach FBFE outperforms all the competing approaches by 99.86 % with n-gram equal to 5 using the highest 100 selected features and the J48 classifier. This study presents, for the first time, the importance of using balanced data, unique features, and feature weight as new parameters to get a high classification performance.

References:

[1] Jyoti Landage and Wankhade, *Malware and Malware Detection Techniques: A Survey*, International Journal of Engineering Research & Technology (IJERT), Vol. 2, pp. 61-68, India, 2013.

[2] G. Padmavathi and S. Divya, *A Survey on Various Security Threats and Classification of Malware Attacks, Vulnerabilities and Detection Techniques*, The International Journal of Computer Science & Applications (TIJCSA), Vol. 2, pp. 66-72, India, 2013.

[3] Mohamad Fadli Zolkipli and Aman Jantan, *A Framework for Malware Detection Using Combination Technique and Signature Generation*, IEEE International Conference on Computer Research and Development, pp. 61-68, Malaysia, 2010.

[4] Kai Huang, Yanfang Ye and Qinshan Jiang, *ISMCS: An Intelligent Instruction Sequence based Malware Categorization System*, IEEE International Conference of Anti-counterfeiting, Security, and Identification in Communication, pp. 509-501, China, 2010.

[5] Mohammad Wazid, Avita Katal, R.H. Goudar, D.P. Singh and Asit Tyagi, *A Framework for Detection and Prevention of Novel Keylogger Spyware Attacks*, IEEE International Conference on Intelligent Systems and Control (ISCO), pp. 433-438, India, 2012.

[6] Karan Sapra, Benafsh Husain, Richard Brooks and Melissa Smith, *Circumventing Keyloggers and Screendumps*, IEEE International Conference on Malicious and Unwanted Software, pp. 103-105, USA, 2013.

[7] Raihana Md Saidi, Siti Arpah Ahmad, Noorhayati Mohamed Noor and Rozita Yunos, *Windows Registry Analysis for Forensic Investigation*, IEEE International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), pp. 132-136, Malaysia, 2013.

[8] Matthew G. Schultz, Eleazar Eskin, Erez Zadok and Salvatore J. Stolfo, *Data Mining Methods for Detection of New Malicious Executables*, IEEE International Symposium on Security and Privacy, pp. 38-49, USA, 2001.

[9] Raja Khurram Shazhad, Syed Imran Haider and Niklas Lavesson, *Detection of Spyware by Mining Executable Files*, IEEE International Conference on Availability, Reliability and Security (ARES), pp. 295-302, Sweden, 2010.

[10] Ivan Firdausi, Charles Lim, Alva Erwin and Anto Satriyo Nugroho, *Analysis of Machine Learning Techniques Used in Behavior-Based Malware Detection*, IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 201-203, Indonesia, 2010.

[11] Raja Khurram Shahzad, Niklas Lavesson and Henric Johnson, *Accurate Adware Detection using Opcode Sequence Extraction*, IEEE International Conference on Availability, Reliability and Security (ARES),pp. 189-195, Czech Republic, 2011.

[12] Donghwi Lee, Won Hyung Park and Kuinam J Kim, *A Study on Analysis of Malicious Codes Similarity Using N-Gram and Vector Space Model*, IEEE International Conference on information and applications (ICISA), pp. 1-4, Republic of Korea, 2011.

[13] Raja Khurram Shahzad and Niklas Lavesson, *Detecting scareware by mining variable length instruction sequences*, IEEE International Conference on Information Security South Africa (ISSA), pp. 1-8, South Africa, 2011.

[14] Asaf Shabtai, Robert Moskovitch, Clint Feher, Shlomi Dolev and Yuval Elovici, *Detecting unknown malicious code by applying classification techniques on opcode patterns*, Springer on Security Informatics, Germany, 2012.

[15] Zongqu Zhao, Junfeng Wang and Jinrong Bai1, *Malware detection method based on the control-flow construct feature of software*, International Journal of The Institution of Engineering and Technology (IET) on Information Security, Vol. 8, pp. 18-24, England, 2013.

[16] Leena T. Patil, Shamal S. Pawar, Shrutika N. Lad and Nilambari Joshi, *Implementation of Spyware Detection Using Data Mining With Decision Tree Algorithm*, International Journal of Engineering Research and Applications (IJERA),Vol. 4, pp. 01-04, India, 2014.

[17] VX Heavens, http://vx.netlux.org, accessed 01-10-2015.

[18] Ian H. Witten, Eibe Frank and Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd edition, San Francisco, CA, Morgan Kaufmann Publishers, Inc., USA, 2011.