Eng Int Syst (2020) 1: 15–22

Engineering
Intelligent Systems

Static Detection and Simulation of Malicious Code in a Metallurgical Master Control Station Based on Behavior Information Gain

Tingfeng Hu*

Wuxi City College of Vocational Technology, Wuxi 214153, China

This paper uses data mining technology to detect malicious code, and proposes a feature selection method based on behavior information gain. Considering the function of feature frequency and information gain, the proposed method can select the most effective features more accurately, and improve the detection performance, thus realizing a malicious code detection system, N-gram and variable length N-gram binary codes are used as the feature extraction method. As the feature selection method, the method of information gain uses several classifiers to detect malicious codes. Experimental results show that this method can effectively improve the accuracy and detection rate of detecting malicious code.

Keywords: Behavior information gain; Malicious code; Static detection; Simulation

1. INTRODUCTION

Traditional malicious code detection technology is mainly based on signature and heuristic methods. The signature detection method attempts to match the features of the detected virus to the features of known virus samples, and match malicious code characteristics by searching through a library of known viruses. This method has a high detection rate, but it cannot detect emerging viruses [1–3]. The heuristic algorithm uses a software expert to define a set of behaviors for detecting unknown malicious code, this method has a high accuracy but low efficiency [4–7]. Data mining methods can effectively detect unknown malicious code by studying the difference between malicious code and normal code. Kephart has proposed a feature extraction and selection method, using artificial neural networks to detect boot viruses [8-10]. Arnold uses the same method to detect Win32 viruses [10–13].

Yousuf M. proposed the data mining technology detection of unknown malicious code, and respectively extracts code features such as Win32 dll file calls, ASCII strings, byte sequences, and uses a variety of classification algorithms, including RIPPER, Naive Bayesian and multiple Naive Bayesian algorithms. For the maximum accuracy of detecting unknown malicious code the multiple Naive Bayesian algorithm using the byte sequence as the detection feature is used [14]. Wang S., Lu S., Zhou N., et al.proposed an N-gram feature extraction method based on binary code, and uses the K-NN algorithm to detect malicious code [15]. Guerbai Y., Chibani Y., Hadjadji B. uses N-gram to extract the features of the binary code, and uses information gain for feature selection, then uses Naive Bayesian, SVM, and boosted J48 algorithms to realize the classification. Among these algorithms, boosted J48 has the highest detection rate [16]. As variable length N-gram is applied to intrusion detection and text classification, Braccesi C., Cianetti F., Lori G., et al. used N-gram to extract the binary code features, and uses class

*Email: sunn2019@163.com

domain frequency for feature selection, and then uses the J48 classification algorithm for the classification [17].

In order to verify the effectiveness of malicious code detection, a static detection simulation method of malicious code based on behavior information gain is proposed. Firstly, this method extracted N-gram of binary code and variable length N-gram as the feature, and proposed a feature selection method based on behavior information gain. The effective feature is selected for the classifier to learn [18–20]. The occurrence and occurrence frequency of feature detection and the feature selection method of behavior information gain in comparison to the frequency of the occurrence was used to comprehensively evaluate the amount of information contained in a feature in order to make up for the information gains. The information gains only consider whether characteristics appear. Finally, the results show that this method can select the effective features more accurately, and improves the detection rate and accuracy of malicious code.

2. BEHAVIOR INFORMATION GAIN

The feature selection method of behavior information gain is to select a set of the most effective features from many features, and the information gain is also called average mutual information. The definition is as follows:

$$I = (X, Y) = H(X) - H\left(X/Y\right) \tag{1}$$

Among them, H(X) is the information entropy of X; H (X/Y) is the conditional entropy of X in the case of known Y. The above formula shows that the average mutual information I (X; Y) obtained from Y is equal to the elimination of the average uncertainty of X before and after learning Y.

In malicious code detection, the information gain IG (J) represents the average information content transmitted by the j-th feature, which is obtained by Formula (1):

$$IG(j) = \sum_{v_j \in (0,1)} \sum_{C_i} P\left(v_j, C_i\right) b \frac{P\left(v_j, C_i\right)}{p\left(v_j\right) p\left(C_i\right)} \tag{2}$$

Among them, v_j is the j-th feature attribute value. $v_j = 1$ shows that this feature has occurred. $v_j = 0$ shows that this feature has not appeared; C_i represents the i-th category. There are two categories: malicious code and normal code; $P\left(v_j, C_i\right)$ represents the proportion that j-th eigenvalue is v_j in class C_i; $P\left(v_j\right)$ represents the proportion that j-th eigenvalue is v_j in the training set; $P\left(C_i\right)$ represents the proportion of class C_i in the training set. The greater the information gain, the more useful this feature is for classification. When calculating the information gain, the information gain feature extraction method sets the existence of a feature as a Boolean value, which only considers whether it exists in the code, but ignores the function of the frequency of the occurrence of each feature. In a real-world situation, the frequency of occurrence of some features in two classes of code can be different, which leads to a very effective way to detect malicious code correctly. This paper considers the effect of feature frequency and information gain, and proposes a feature selection method based on weighted information gain, which can more accurately select the effective features, and improve the accuracy and detection rate of malicious code.

The behavior information gain of the j-th feature is defined as follows:

$$IW(j) = \lambda_1 \sum_{v_j \in (0,1)} \sum_{C_i} P\left(v_j, C_i\right) b \frac{P\left(v_j, C_i\right)}{p\left(v_j\right) p\left(C_i\right)} \tag{3}$$

Among them, λ_j represents the weight corresponding to the j-th feature.

A larger behavior information gain IW(j) value indicates that this feature is more effective for correctly classifying malicious code.

3. STRUCTURE OF STATIC DETECTION MODEL OF MALICIOUS CODE BASED ON BEHAVIOR INFORMATION GAIN

The malicious code static detection system based on data mining method is used, the N-gram and variable length N-gram of binary code sequence are used as the feature, the weighted information gain is used as the feature selection method and the multiple classification algorithm is used to achieve the system of malicious code detection. The model is shown in Figure 1.

The training part firstly selects a certain number of malicious and normal code as the training set, extracts N-gram and variable length N-gram of code binary sequence as the feature; performs the feature selection and calculates the weighted information gain IW(j) corresponding to each feature, and then makes a descending sort in accordance with IW(J), and selects several of the features as effective features. According to whether each training sample includes these features or not, a Boolean vector space is formed which allows the classifier to learn. The detection part extracts N-gram and variable length N-gram of code binary sequence to be detected as the feature, and forms a Boolean vector space according to whether each sample code contains effective features selected by the training part. The classifier uses several classification algorithms to analyze the vector space, and judges if malicious code is detected.

Supposing that the number of samples of codes to be detected is N, thus N = TP + TN + FP + FN. Among them, TP is the number of malicious codes that are correctly classified; FP is the number of normal codes that are falsely marked as malicious codes; TN is the number of normal codes that are correctly classified; FN is the number of malicious codes that are falsely marked as normal code. A malicious code detection tool has the following two evaluating indicators:

(1) accuracy rate $\frac{TP+TN}{N}$, that is, the proportion of the total code that is correctly classified in the set to be detected.

(2) detection rate $\frac{TP}{TP+FN}$, that is, the proportion of malicious code that is correctly classified within all the malicious code of the set to be detected.

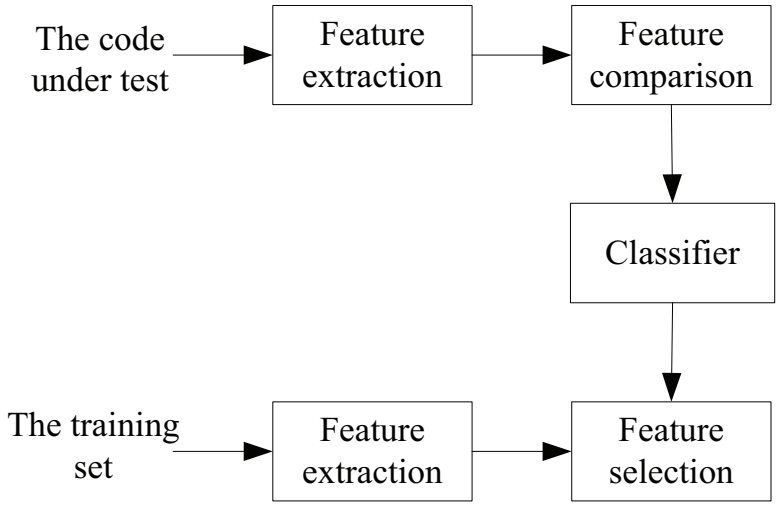

Figure 1 Malicious Code Static Detection Model

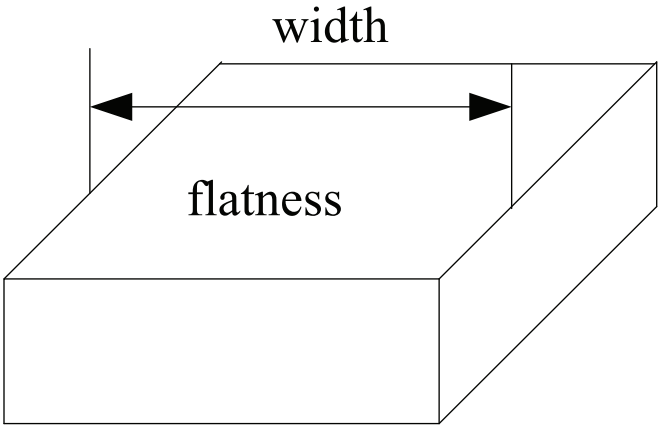

Figure 2 Geometric Quantity Detection

4. DESIGN OF THE SIMULATION SYSTEM OF THE STRUCTURE OF MALICIOUS CODE STATIC DETECTION MODEL BASED ON BEHAVIOR INFORMATION GAIN

The object-oriented programming language VC + + 6 is used as the system development tool, and OpenGL is used as the 3D scene development tool. The static detection simulation system of malicious code is developed according to the concepts of object-oriented programming. Generally, this simulation system should focus on the following problems: (1) establishment of a static detection environment (2) extraction of detection information (3) drive of virtual probe.

4.1 Establishment of a Static Detection Environment

In a CAD/CAM system, graphics modeling and numerical control programming are essentially the same. Established graphical models provide direct simulation scenes for simulation detection. Geometrical detection is mainly aimed at geometric features. The detection of height is aimed at the geometric features of a plane, and the geometric quantity detection isn't for the workpiece, which makes it difficult to establish a simulated detecting scene. It is therefore difficult to establish a realistic three-dimensional model of the work piece. Taking into account the periodicity of development of engineering technology, the eclectic establishment method of detection environment is used in this paper, so as to realize the prototype design of the simulation system: the geometrical characteristics of geometric sense measurement corresponds to basic geometric shapes. The simulation scene is built according to basic geometric shapes; at the same time, the basic geometric shape is designed as the parameter type, so that it can be matched with each detection part of the work piece, in order to realize the process of simulating geometrical sense detection. In the simulation system, the available basic geometric shapes include cuboid, groove, cylinder, hole, cone, sphere and so on. Using these basic geometric shapes can realize the process of simulating geometric sense measurement.

For the detection of a cuboid, the geometric characteristics of a cuboid can be used to detect the geometric dimensions such as height and width. As shown in Figure 2, we can use two parallel planes of cuboid to realize the detection of length, and use a plane to realize the detection of height (through the acquisition of three-dimensional coordinates

of some points), thus solving the problem of geometrical visualization of geometric dimension measurement. In order to build a static detection environment, the OpenGL standard for graphics processing is used. OpenGL is a software interface to graphics hardware, it includes a three-dimensional graphics and models library. It has excellent performance in three-dimensional realistic graphics production, which has become the new standard in the three-dimensional graphics industry. It can be used for geometric modeling, graphics transformation, rendering, light, material quality and other operations. Most of the underlying processing of graphics is processed by specialized functions. Program developers are free to focus on their work rather than the graphics modeling aspect.

4.2 Extraction of Detection Information

As a malicious code static detection simulation system, it is necessary to accurately reflect each statement of the measurement macro program during the simulation process. The simulation system should have complete detection information extraction ability, which can realize the measurement of grammar within the measurement procedure and can realize relevant calculations and judgments. The most important is to extract the motion trajectory of the measuring head, driving the detection simulation of the measuring head. The measurement macro program stores its results in a text file. When solving the problem of detection information extraction, the corresponding operation instruction of the numerical control system should be read in detail. In this aspect, information extraction of static detection is more difficult than in a CAD/CAM system. A great deal of logical and mathematical problems are involved in a measuring macro program, whereas a CAD/CAM system is relatively straightforward. It is mainly related to G01 and G00 instructions.

Referring to the FANUC86- A20 numerical control system, the simulation system can process and extract the following information:

1) processing of a G01 instruction;

2) processing of a G31 instruction (leapfrogging instruction);

3) assignment processing of system variables (such as #1, #2);

4) processing mathematical expressions;

5) processing IF judgment statements;

6) processing GOTO jump statements;

7) processing logical arithmetic symbols (such as, and, or, non)

Malicious code static detection simulation can use the method of N-gram and variable length N-gram sliding window for the extraction.

4.2.1 N-Gram Feature Extraction Method

N-gram is a series of overlapping substrings collected by a sliding window with the length of N, This window slides one unit length at a time. For example, for 12 11 74 ff 03 b2, the corresponding 3-gram is (12, 11, 74), (11, 74, FF), (74, FF, 03), (FF, 03, B2). N-gram can capture some potential features which are difficult to accurately extract by other means. In the field of malicious code detection, N-gram is a widely used feature extraction method. There are two disadvantages in N-gram feature extraction:

(1) it is very difficult for N-gram to capture byte sequences with different lengths at the same time. When a meaningful byte sequence is not multiples of N, it will produce an edge mismatching, which cannot extract this feature.

(2) the feature sets generated by N-gram are very large, which requires a considerable storage capacity. In the implementation of N-gram algorithm, the Trie data structure is adopted to save storage space and ensure the fast and accurate feature generation and search.

4.2.2 Variable Length N-Gram Feature Extraction Method

Variable length N-gram is also called paragraph, which is a series of meaningful successive byte sequences. It is different from a fixed length N-gram, avoiding the possibility that a meaningful sequence is taken apart. To extract meaningful paragraphs, breakpoints in a series of byte sequences need to be found. The successive sequence between adjacent breakpoints is a paragraph. The paragraph segmentation algorithm in this paper adopts the expert voting algorithm. The expert voting algorithm assumes that there are two experts. One is the frequency expert, who is responsible for measuring the frequency of each sub sequence. The higher the frequency, the greater the possibility that this sequence is a paragraph, the possibility that contains the breakpoint is low. The other is the entropy expert, who is responsible for measuring the entropy of each point. If each time the element that is connected after an element is different, the entropy will be larger, it is more likely to be the end of a paragraph if the element that is connected after an element remains the same. Each position has a score. In a sliding window of fixed length, the scores in the two positions with the maximum frequency and the maximum entropy will be added. Finally, combining the results of the two experts, the possible breakpoints are determined according to the cumulative scores obtained at each position. The continuous sequence between two breakpoints is then extracted as a feature. This paper chooses the Trie data structure with $d = 4$ to realize the expert voting algorithm.

Suppose the string = (01, E8, B8, 01, E8, B8, B8, 01) and its Trie structure is shown in Figure 4. In which the left leaf node E8 represents the byte sequence (01, E8, B8) in the Trie, and the number 2 of the node indicates the number of times the sequence appears. When a window of 3 is left across the string from left to right, first the string (01|E8|B8|)

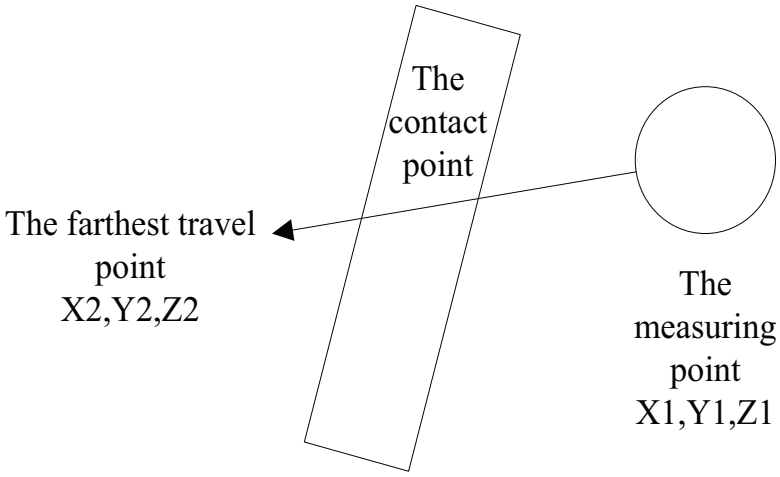

Figure 3 The G31 Command Movement Sketch

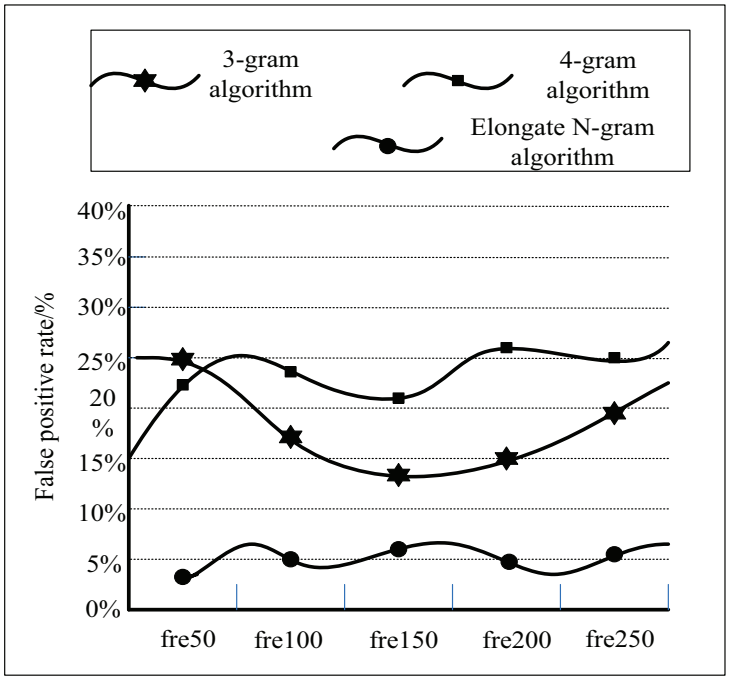

Figure 4 Malicious Code with Characteristic Length N = 3 Detects False Positives

is encountered. The frequency of the first position is the sum of the frequencies of the two sequences separated from each other, i.e. f=f (01) + f (E8, B8). F (01) = f (01) + f (E8, B8) represents the frequency (01) at the Trie first layer, and f (E8, B8) represents the frequency at the Trie second layer and the parent node is (E8) (B8). The entropy of the first position is expressed in the entropy of the first layer (01) of the Trie, and the entropy of the second position is expressed in the Trie second layer, and the parent node is the entropy of (01), and so on. (E8). In this window, the maximum frequency and entropy positions are marked out, and the corresponding score is added. The window is then moved to the right until the whole string is traversed.

The frequency of nodes in Trie is

$$p\left(x_0\right)=\frac{f\left(x_0\right)}{f\left(parent\left(x_0\right)\right)} \tag{4}$$

The entropy of the node is

$$E\left(parent\left(x_0\right)\right)=-\sum_{i=0}^{m} P\left(x_i\right)b \tag{5}$$

Among them, m is the number of subtree of father nodes of 0x.

When the sliding window traverses the whole string, local maximums can be found according to the expert scores, and the corresponding position is marked as the breakpoint. The paragraph segmentation result of string is ((01, E8, B8) (01, E8, B8) (B8, 01)).

4.3 Driver for Virtual Probes

The core problem of the whole simulation detection system is that the simulation must realistically reproduce the process of a collision between the measuring head and the object to be measured to determine the location information of

Table 1 Several Feature Extraction and Selection Method Combination Test Results (%)

Algorithm	Detection Rate	Accuracy
3 - gram(IG)	95.2	95.4
3 - gram (action IG)	96.7	97.2
4-gram(IG)	96.3	97.2
4 - gram (action IG)	98.3	98.8
longer N-gram (IG)	97.9	98.6
longer N-gram (action IG)	98.3	99.2

Table 2 The Combination Of The N-Gram and The Action IG Uses Various Classifiers For Test Results (%)

Algorithm	Detection rate	Accuracy
SVM	97.2	94.3
J48	98.5	99.2
Naive Bayes	95.6	96.4

contact points. Because the virtual object to be measured has accurate geometric data in the computer corresponding to it, the collision detection process of the measuring head is transformed into the intersecting problem between the detection path and the virtual object. If there is an intersection point, the measuring head should move to the intersection point then stop, this shows that the measuring head touches the object and the detection and simulation function is realized. For the on-line inspection system of the machining center, the key actions for realizing the detection is that the measuring head will hit the object to be measured in a straight line. In order to ensure the measuring head can reliably impact the object to be measured, the maximum stroke of the measuring head detection movement should be made to be greater than the distance from the measuring head to the actual contact point, namely, the actual contact point is located at the straight line between the starting point of measurement and the maximum stroke point of the measuring head. For the on-line detection system of the machining center, the key instruction to achieve the detection function is the leapfrogging instruction G31. Two key data associated with this instruction is: position of the starting point of the measuring head movement (X1, Y1, Z1) and the position of the maximum stroke point of the measuring head (X2, Y2, Z2), therefore the measuring head moving trajectory is reduced to a straight line. The work piece to be detected is composed of several planes. The "intersection" problem of the detection path and the virtual object is transformed into the intersection problem of the tangential path and these planes. By calculating the specific intersection point, the virtual probe can be driven to move according to the programmed instructions. The detection principle is shown in Figure 3.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Experimental Steps

This paper uses 429 normal codes and 408 malicious codes as the training sample set. All malicious codes come from the website of http://vx.netlux.org. The normal codes are obtained from a computer with a freshly installed Windows XP operating system.

The number of effective features is 500. The classification algorithm is realized by WEKA, and the experiment uses the tenfold cross validation, and uses Matlab 2012b and Wireshark as simulation tools, and uses Spread (M, r, s, t) information propagation model in Matlab as the model of a wireless sensor network. The number M of initialized network nodes = 20, and the information to be transmitted and received by nodes uses r = random mode. The information consumption of network node side is set to s = 0. The intrusion feature delay is T = 1 second.

5.2 Experimental Results

When using 3-gram, 4-gram and variable length N-gram algorithm, this paper uses information gain and behavior information gain as the feature selection method, and selects Naive Bayes, SVM, and J48 as the detection system of malicious code of classifiers, and selects detection results with the optimal classification effect for each combination to fill in Table 1. Among them, the combination of variable length N-gram and behavioral information gain has the highest detection rate. From Table 1, the detection results using behavioral information gain are better than the detection results of information gain. The performance of 4-gram is better than that of 3-gram, and the performance of variable length N-gram is better than that of N-gram.

When using variable length N-gram as feature extraction method and using behavior information gain as a feature selection method, the detection results of various classification algorithms are shown in Table 2. Among them, the detection rate of the J48 algorithm is the highest.

The above table clearly shows that the static detection simulation of malicious code based on behavior information gain uses N-gram as the feature extraction method, which effectively improved the efficiency of detection simulation.

The above experiments show that the behavior information gain detection method is superior to the information gain detection method. Then, the 3-gram, 4-gram and variable length N-gram algorithm methods are used to compare the

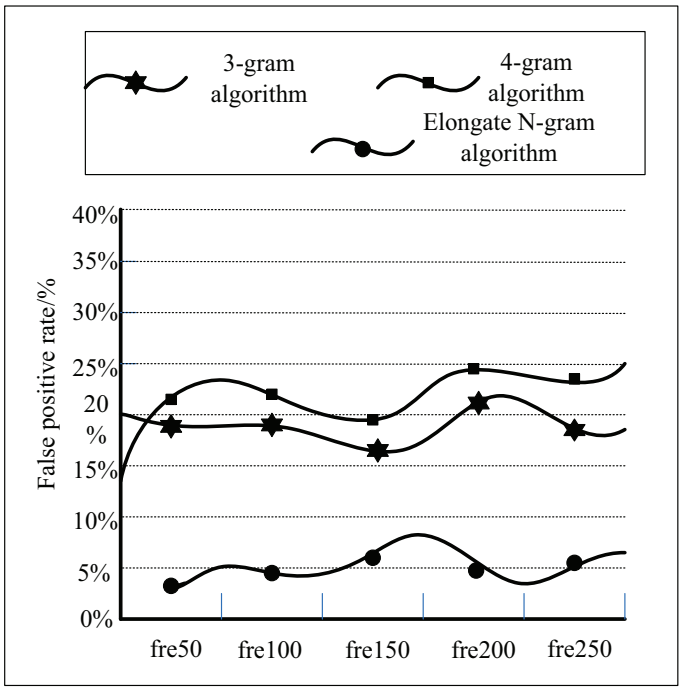

Figure 5 Malicious Code with Characteristic Length N = 4 Detects False Positives

static detection false positive rate of malicious codes in different feature length of contrast, and 3-gram, 4-gram and variable length N-gram algorithm are used for experiments. With the increase of the frequency characteristic of document, the false positive rate under different feature length is recorded.

From Figure 4, when the characteristic length is 3, the false positive rate of the three algorithms is obviously different with the increase of the frequency feature of documents. When the frequency feature of documents is within the 50 interval, the false alarm rate of the 3-gram algorithm reaches 25%. The 4-gram algorithm is slightly lower than this yet its false alarm rate is still above 20%. In the same interval of document frequency, the false alarm rate of variable length N-gram algorithm is about 2.5%. In the process of increasing the frequency feature of the documents to 250, the false positive rate of the 3-gram algorithm and the 4-gram algorithm is still above 10%, the maximum is 26%, and the false alarm rate is very unstable. The curve fluctuates greatly. In the process of the whole experiment, the false positive rate of variable length N-gram algorithm is kept below 6%, and the fluctuation of the curve is very small.

On the basis of above experiments, in order to more effectively illustrate the effectiveness of the N-gram algorithm, under the premise of the characteristic length N=4, the experiment is carried out again. Experimental comparison results are as follows.

According to Figure 5, it can be concluded that when the characteristic length N = 4, the false positive rate of detection of the 3-gram and 4-gram algorithms is still between 15% and 25%, compared with the characteristic length N = 3, but the false positive rate is not obviously reduced. In this experiment, the false positive rate of N-gram algorithm is between 2.5% and 7%, and the false positive rate is low. After two experiments, this strongly suggests that using N-gram as the feature detection method can effectively improve the detection accuracy.

6. CONCLUSION

Malicious code static detection simulation technology is an effective means to further improve the work efficiency of the detection center, which makes the whole detection process flow more smoothly. CAD/CAM technology solves the problem of automating numerical control programming, and static detection simulation technology will solve the problem of malicious code. How to improve the operability of simulation detection technology is a key problem. Based on the CAD/CAM technology development idea, this paper shows that the simulation technology will further enhance the feasibility of static detection technology of malicious code. In this paper, the object-oriented programming design concept is used to make a comprehensive plan of the simulation system, and the corresponding simulation software was developed. The actual application shows that the simulation system detection based on behavior information gain enhances the operability of static detection of malicious code and improves the activity of the simulation detection technology, which lays a solid foundation for the future of simulation technology.

REFERENCES

1. Rogers A.E.E., Pratap P., Carter J.C., et al. Radio Frequency Interference Shielding and Mitigation Techniques for a Sensitive Search for the 327 Mhz Line of Deuterium. Radio Science, 2016, 40(5): 1–10.
2. Stavropoulos T., Andreadis S., Bassiliades N., et al. The Tomaco Hybrid Matching Framework for SAWSDL Semantic Web

Services. IEEE Transactions on Services Computing, 2016, 9(6): 954–967.
3. Zhang B., Li Q., Ma Y. Research on Dynamic Heuristic Scanning Technique and the Application of the Malicious Code Detection Model. Information Processing Letters, 2017, 117: 19–24.
4. Topal A.O., Altun O. A Novel Meta-Heuristic Algorithm: Dynamic Virtual Bats Algorithm. Information Sciences, 2016, 354: 222–235.
5. Ordin B., Bagirov A.M. A Heuristic Algorithm for Solving The Minimum Sum-Of-Squares Clustering Problems. Journal of Global Optimization, 2015, 61(2): 341–361.
6. Allahyari S., Salari M., Vigo D. A Hybrid Metaheuristic Algorithm for the Multi-Depot Covering Tour Vehicle Routing Problem. European Journal of Operational Research, 2015, 242(3): 756–768.
7. Wang L., Song J., Shi L. Dynamic Emergency Logistics Planning: Models and Heuristic Algorithm. Optimization Letters, 2015, 9(8): 1533–1552.
8. Taudte R.V., Roux C., Bishop D., et al. Development of a UHPLC Method for the Detection of Organic Gunshot Residues Using Artificial Neural Networks. Analytical Methods, 2015, 7(18): 7447–7454.
9. Gelisli K., Kaya T., Babacan A.E. Assessing the Factor of Safety Using an Artificial Neural Network: Case Studies on Landslides in Giresun, Turkey. Environmental Earth Sciences, 2015, 73(12): 1–8.
10. Lee D.H., Kang D.S. The Application of the Artificial Neural Network Ensemble Model for Simulating Streamflow. Procedia Engineering, 2016, 154: 1217–1224.
11. Tun K.M., Imwong M., Lwin K.M., et al. Spread of Artemisinin-Resistant Plasmodium Falciparum in Myanmar: A Cross-Sectional Survey Of The K13 Molecular Marker. Lancet Infectious Diseases, 2015, 15(4): 415–21.
12. Kepko L., Mcpherron R.L., Amm O., et al. Substorm Current Wedge Revisited. Space Science Reviews, 2015, 190(1): 1–46.
13. Kateera F., Mens P.F., Hakizimana E., et al. Malaria Parasite Carriage and Risk Determinants in a Rural Population: A Malariometric Survey in Rwanda. Malaria Journal, 2015, 14(1): 1–11.
14. Yousuf M. Preparation and Characterization of Nanoparticles of Active Phytoconstituents Extracted from Natural Herbs Used as Psychotropic and Behavioral Drugs. Febs Letters, 2015, 579(11): 2499–506.
15. Wang S., Lu S., Zhou N., et al. Dynamic-Feature Extraction, Attribution, and Reconstruction (DEAR) Method for Power System Model Reduction. IEEE Transactions on Power Systems, 2014, 29(5): 2049–2059.
16. Guerbai Y., Chibani Y., Hadjadji B. The Effective Use of the One-Class SVM Classifier for Handwritten Signature Verification Based on Writer-Independent Parameters. Pattern Recognition, 2015, 48(1): 103–113.
17. Braccesi C., Cianetti F., Lori G., et al. Random Multiaxial Fatigue: A Comparative Analysis Among Selected Frequency and Time Domain Fatigue Evaluation Methods. International Journal of Fatigue, 2015, 74(3): 107–118.
18. Emma B.F., Cyril G., Carmen C., et al. Selective Catalytic Deuteration Of Phosphorus Ligands Using Ruthenium Nanoparticles: A New Approach to Gain Information on Ligand Coordination. Chemical Communications, 2015, 51(91): 16342–5.
19. Rogge-Solti A., Weske M. Prediction of Business Process Durations Using Non-Markovian Stochastic Petri Nets. Information Systems, 2015, 54(C): 1–14.
20. Ortiz-Servin J.J., Cadenas J.M., Pelta D.A., et al. Nuclear Fuel Lattice Performance Analysis by Data Mining Techniques. Annals of Nuclear Energy, 2015, 80: 236–247.