A Multi-threading Solution to Multimedia Traffic in NIDS Based on Hybrid Genetic Algorithm

Xu Zhao[1], Guangqiu Huang[2], and Reza Mousoli[3]
(Corresponding author: Xu Zhao)
School of Computer Science, Xi'an Polytechnic University[1]
No. 19, Jinhua South Road, 710048, Xi'an, China
(Email: 37274679@qq.com)
School of Management, Xi'an University of Architecture and Technology[2]
No. 13, Yanta Road, Beilin District, 710055, Xi'an, China
School of Law, Criminal Justice and Computing, Canterbury Christ Church University[3]
North Holmes Road, CT1 1QU, Canterbury, UK
(Received July 27, 2018; Revised and Accepted Jan. 24, 2019; First Online July 30, 2019)

Abstract

Packet omission and subsequently data loss is inevitable when the network traffic exceeds the load capacity threshold of Network Intrusion Detection System (NIDS). In these circumstances, relatively dangerous packets should be given priority for processing by NIDS. To address this problem and offer a possible remedy, this paper proposes a multi-threading solution specifically for multimedia packets in NIDS by using two different genetic algorithm. In this solution, two optimization objectives are achieved simultaneously: One is to maximize the sum of danger coefficient of multimedia packets in every thread and the other is to process the workload of each thread at its maximum workload. This paper also compares the advantages of hybrid genetic algorithm to simple genetic algorithm in the implementation process of the proposed solutions. By using this proposed solution, NIDS can identify multimedia packets and then select the more dangerous multimedia packets for processing within the maximum processing capacity of different threads when packet omission occurs. Experimental results indicate that this solution can help NIDS to improve its differentiation and selection ability for dangerous multimedia packets effectively.

Keywords: Danger Coefficient; Genetic Algorithm; Multimedia Packets; Network Intrusion Detection System (NIDS); The Solution of Choosing Danger

1 Introduction

1.1 Background

NIDS is a system that detects malicious activities by monitoring network traffic. With the rapid increase of network speed, the requirements of the NIDS's processing efficiency has also increased. Nevertheless, packet omission is inevitable when the network traffic exceeds the load capacity of NIDS. This problem has stimulated extensive research studies on how to reduce the omission ratio of NIDS and how to minimize the security risks when the omission becomes inevitable. Among these reseach activities for finding a optimal solution, artificial intelligence has attracted considerable interest from the research community and various artificial intelligence algorithms in improving NIDS has been investigated, such as Artificial Neural Networks [9, 12], Clustering algorithm [3, 7, 11], Particle Swarm Optimization [1, 4, 10] and genetic algorithm [2, 5, 13, 14]. However, artificial intelligence algorithms have not been found for multimedia traffic analysis for NIDS, and the study in this paper addresses this gap in order to solve this shortcoming.

With the rapid development of high-speed networks, the proportion of multimedia packets in the network traffic is increasing. The targeted method of processing multimedia packets can greatly improve the efficiency of NIDS. Previously we have proposed an identifying method and two separate processing methods for multimedia packets to raise the efficiency of the NIDS and have gained satisfactory results [6].

Because of the various types of multimedia, the security of multimedia packets can vary according to different types of packets. Under the premise of limited system processing capacity, the more dangerous multimedia packets should be given priority for processing when the network traffic is too great and packet omission becomes inevitable. On this basis, we propose a multi-threading solution to multimedia packets in NIDS systems based on genetic algorithm. When packet omission occurs, this solution can select more dangerous multimedia packets for processing within the maximum processing capacity of different threads.

1.2 Contributions Summary

The main contributions of this paper are:

1) The study of multimedia packets relating to NIDS in the networks. On the basis of previous studies [6, 8, 15, 16] for multimedia packets, we propose a multi-threading solution for multimedia packets based on hybrid genetic algorithm. By using this solution, NIDS can focus its limited processing power on more dangerous multimedia packets when omission becomes inevitable.

2) We propose two optimization objectives to optimize NIDS:

   - The sum of danger coefficients of multimedia packets in every threads is maximal.
   - When the above objective is achieved, the load of each thread exacctly reaches the highest.

3) We introduce two concrete implementations with simply genetic algorithm and hybrid genetic algorithm of the solution. The advantages and disadvantages of the two solutions are also analyzed. In addition, we also design several experiments which compare the packet loss rate, the sum of danger coefficients, the detection number of multimedia packets and the detection rate of dangerous incident when using different solutions. We also prove its effectiveness based on above-mentioned experimental results.

1.3 Paper Organization

The rest of the paper is organized as follows: First, related studies are discussed in Section 2; Section 3 presents the description of the solution; Section 4 describes the determination of the value of parameters. Section 5 presents the implementation of the solution with two different genetic algorithms; The experiment and an analysis of the result for contrasting the differences before and after using the solution is in Section 6; Section 7 summarizes the whole paper and presents some directions for the future work.

2 Related Work

2.1 Artificial Intelligent Algorithms on NIDS

Artificial intelligent algorithms usually offer an automatic mechanism to enhance the performance of NIDS. Here are several common algorithms.

1) Artificial Neural Networks [3,9,12]: DeLima [12] presented a method for building a prototype for NIDS, which uses an artificial neural network as a detection mechanism. Nevertheless, the adjustment of weights is somewhat complex.

2) Clustering algorithm [1,4,7,11]: Chandrashekhar [11] proposed an efficient intrusion detection model by amalgamating competent data mining techniques such as K-means clustering, Multilayer layer perception (MLP) neural network and support vector machine (SVM).This model can improve the prediction of network intrusions, but it also has problems. For example, it might mistake dubious data for normal data.

3) Particle Swarm Optimization [2,5,10,13]: Cleetus [5] proposed an intrusion detection solution based on particle swarm optimization by using multiobjective functions. This solution has a strong global search capability which is used for dimensionality optimization. However, it is easy to fall into local optimum.

4) Genetic algorithm [6, 8, 14, 15]: In the field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions.

Samaneh Rastegari [14] proposed a solution that uses genetic algorithm to evolve a set of simple, interval-based rules based on statistical, continuous-valued input data. This new approach provides a very compact set of simple, human-readable rules with strongly competitive detection performance in comparison to other machine learning techniques. But this approach should be modified for multi-class classification, to discover rulesets that can identify which kind of attack is taking place.

Additionally, although [6] using GA to improve the network attack detection accuracy, but without considering the GA individual selection adjustment, and it only uses KDD dataset to validate their methods, and not for practical applications.

Furthermore, Dustin Y. Harvey [8] described a method using GA to identify irregular network intrusion. The process includes both quantitative and definite features of network data for deriving classification rules. Though, the addition of quantitative feature can amplify the detection rate no tentative results are present.

2.2 The Identifying and Processing Methods for Multimedia Packets

With the increasing speed of network, the proportion of multimedia packets in network traffic is increasing. Compared to other packets, because the multimedia packets are relatively safe, the NIDS has less detection rules for specific multimedia types [16]. Therefore, the recognition of multimedia packets and the separate processing according to different multimedia types will greatly improve the performance of NIDS. O.Marques of Florida Atlantic University originally proposed this idea, but he did not give specific solutions [15]. We have put forward a series of

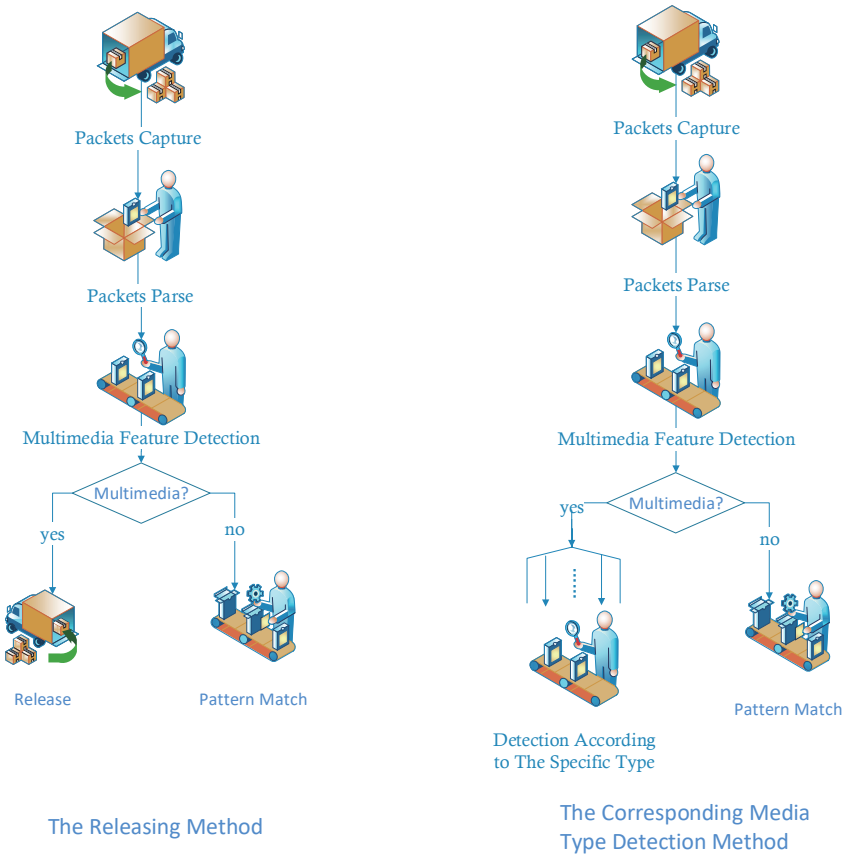

Figure 1: The work flow of two methods *** Please redraw this figure. Part of the icons (pictures) maybe have copyright problem. (Please don't use the commercial picture or icons). Please use a "rectangle with text" to replace the icons. ***

processing methods [17–20] of NIDS for multimedia packets with good results. In the meantime, O. Marques and Pierre have also carried out follow-up studies [21], but their focus is mainly on the loopholes of streaming and non streaming specific multimedia files. Similarly, Zander of the Murdoch University in Australia proposed the classification [22] of multimedia traffic in the firewall by machine learning technology, which provides a reference for the depth detection method of multimedia packets.

Specifically, we have proposed an identifying method and two particular processing methods for multimedia packets in [16]. The principle of the identifying method for multimedia packets is to detect the multimedia features information at the front of multimedia packets.

The two processing methods are the Releasing method and the CMTD (corresponding media type detection) method. Figure 1 illustrates the work flow of two methods. The Releasing method lets identified the multimedia packet skip over the conventional detection process. Though this method is simple and efficient, its security is lower.

The CMTD method [16] is a safe and efficient method. In order to achieve it, a multimedia rule base is created. The multimedia rule base stores the rules which are collected specifically for multimedia packets. The CMTD method can be used to choose the corresponding multimedia rules according to the specific multimedia type that the packet carries in order to pre-detect intrusive characteristics. If there is no problem, it is released immediately; if there exists a problem, put it into the conventional detection process. Because there are far fewer multimedia rules than rules for conventional detection process in NIDS, this method can significantly improve the detection efficiency for most of the safe multimedia packets. The safety of this method is also greater than that of the releasing method.

Although these methods can raise the efficiency of NIDS, they are mainly suitable for no-omission. On this basis, we propose a multithreading solution to multimedia packets in NIDS based on genetic algorithm. When omission occurs this solution can choose more dangerous multimedia packets for processing within the maximum processing capacity of different threads.

3 Description of Solution

According to the MIME protocol, the multimedia packet types in the network are as many as 133. Nevertheless, the risk of every type of multimedia packets is different. For example, octet-stream*.exe is more dangerous than others. According to the set-method of the danger co-

efficient of multimedia types in [20], the multithreading solution to multimedia packets in NIDS can be described as following:

N multimedia packets $P_1, P_2, \cdots, P_n$ have been captured, let the load of each thread in NIDS be set as LT, then the load of these multimedia packets to NIDS is $L(P_i) \in (0, LT](i = 1, 2, \ldots, n)$ and the danger coefficient of these multimedia packets is $D_k(P_i)(k = 1, 2, \ldots, 133, 1 \leq i \leq n)$.The key point is how to determine the distribution solution which can make the highest sum of danger coefficient of multimedia packets in each thread and guarantee the load of these packets is less than the load of each thread.

4 How to Determine the Value of Parameters $L(P_i)$ and $D(P_i)$

$L(P_i)$: The load to the system which is caused by multimedia packet Pi. Because there are significant positive correlations [17] between the time complexity of the pattern matching algorithm and length of the string to be matched, $L(P_i)$ is determined by the ratio of the actual length of the packet load and the total length.

$D(P_i)$: The value of $D(P_i)$ should be set for different media types according to the degree of risky information [17] carried by packets (as shown in Table 1). For instance, executable files can appear in multimedia files of octet-stream type, the value of D(Pi) of octet-stream type can be set higher. The table below shows the value of $D(P_i)$ for several common multimedia types.

Table 1: The value of P for several common multimedia types

multimedia types	file type	$D(P_i)$
octet-stream	exe rar	3.0
x-JavaScript	js	2.1
x-tar	tar	2.6
jpeg	Jpzjpg jpeg	1.5
gif	gif	1.5
html	htm html hts	1.3
x-shockwave-flash	swf swfi	1.8
.....	.....	.....

5 Implementation of the Solution

To solve the above-mentioned problem, we will illustrate the solution by two genetic algorithms in the following sections and compare the differences between them. One is simple genetic algorithm, the other is Hybrid genetic algorithm which consists of the simple genetic algorithm and the FFD approximation algorithm.

5.1 The Solution by Using Simple Genetic Algorithm

Step 1: Individuals Code. Because the operation object of genetic algorithm is the symbol string which indicates individual, the operation object must be encoded as the symbol string in this solution. The chromosome coding method is defined as follows:

Chromosome coding method: Let K thread number be $T_1, T_2, T_3, \ldots\ldots T_k, (k \leq n)$, n multimedia packets will be loaded into these K threads. The number sequence of each multimedia packet $P_i(i = 1, 2, ..., n)$ which is loaded into these threads constitutes the chromosome coding of this problem. For example, $T_1T_3T_1T_2...T_2T_1$ means that multimedia packets P_1, P_2, P_3 are loaded into the thread T_1, multimedia package P_2 is loaded into thread T_3, *etc.* The initial population can be generated by random permutation of $T_1, T_2, T_3, \ldots\ldots T_k$.

Step 2: Initialization. Let the evolution generations counter be t and give t an initial value 0. Let the maximum value be T; the initial population P(0) can be generated by random permutation as in Step 1.

Step 3: Evaluation of the fitness value: P(t) indicates the population that evolves to the t generation. Genetic Algorithm determines the probability of an individual in the current population P(t) to the next generation population by adapting the proportional probability of individual fitness. In order to estimate this probability correctly, every individual fitness in population P(t)must be calculated.

The objective function and the fitness function: Let m be the number of threads used in a distribution scheme, and let $T(P_i)$ be the number of the thread in which multimedia packet P_i is loaded. Besides, let S_j be the sum of load of the multimedia packets in thread T_j. In order to make the best use of all threads, the optimization objective function can be written as Equation (1).

$$\begin{aligned} f(x) &= m \cdot \{m - \sum_{j=1}^{m} s_j\} \\ &= m \cdot \{m - \sum_{j=1}^{m} [\sum_{T(P_i)=T_j} L(P_i) \\ &\quad - a \cdot \max(0, \sum_{T(P_i)=T_j} L(P_i) - 1)]\}. \end{aligned} \tag{1}$$

The danger coefficient objective function of multimedia packets in each thread is:

$$max \sum P_i D_i, 1 \leq i \leq n \tag{2}$$

In Equation (1), a indicates the penalty factor when the sum of load of the multimedia packets in thread T_j exceeds the load of thread T_j. These two objective

functions not only maximize the sum of danger coefficients of multimedia packets in every thread, but also make each thread reach the highest load. The fitness function is:

$$F(X) = \begin{cases} C_{\max} - f(X), f(X) < C_{\max} \\ 0, f(X) \geq C_{\max} \end{cases} \quad (3)$$

In Equation (3), $C_{\max}$ indicates an appropriate positive which adjusts the fitness function to take a non-negative value.

Step 4: Selection operation. The selection operator can adopt the proportional selection operator. $P^{'}(t)$ can be obtained when the selection operator acts on the population $P^{'}(t)$.

Step 5: Crossover operation. The crossover operator can use the single point crossover operator. $P^{''}(t)$ can be obtained when the crossover operator acts on the population $P^{'}(t)$.

Step 6: Mutation operation. The mutation operator can adopt uniform random variation in the coded character set $V = \{T_1, T_2, T_3, \ldots\ldots T_k\}$. When the mutation operator acts on the population $P^{''}(t)$, $P^{'''}(t+1)$ will be achieved after selection, crossover and mutation operation of $P^{''}(t)$.

Step 7: The judgment of termination condition. When $t <= T$, then $t \leftarrow t+1$. As a new population, $P^{'''}(t+1)$ will replace $P(t+1)$.Next go to Step 2 and start the next cycle. If $t > T$, then the population with the greatest fitness in evolutionary process will be output as the optimal solution and computation will terminate.

5.2 The Disadvantage of the Above mentioned Solution

The disadvantage of the abovementioned solution is that some invalid chromosomes would be generated in initialization and evolutionary process. In the distribution scheme represented by these invalid chromosomes, the sum of load of the multimedia packets in a thread will exceed the load of this thread. This will reduce the operating efficiency of NIDS.

In view of the above-mentioned facts, we propose the following solution by using a hybrid genetic algorithm which consists of the simple genetic algorithm and the FFD approximation algorithm.

5.3 The Solution Combined with FFD Approximation Algorithm

According to the principle of the FFD approximation algorithm, all multimedia packets are firstly listed in descending order according to the amount of load for each packet. Then these packets are distributed to the threads.

The above approaches are applied to the decoding process of the chromosome chain (Step 1 above) in genetic algorithm. Their specific steps are described below:

1) All multimedia packets are listed in descending order according to the amount of load for each packet.

2) The above-mentioned packets are distributed to all threads. When the sum of load of the multimedia packets in thread T_j exceeds the load of thread T_j, then the extra multimedia packets will be distributed to thread T_{j+1}.

In the case of the sum of load of the multimedia packets in thread T_m exceeds the load of thread T_m, the extra multimedia packets will be distributed to a new thread, and $m \leftarrow m+1$.

5.4 The Advantage of the Improved Solution

By using the above method, not only does the distribution scheme of the above steps meet the principle of FFD approximation algorithm, but also the sum of load of the multimedia packets in each thread does not exceed the load of thread. So, Penalty function is not required when the objective function is calculated. The objective function is listed in Equation (4).

$$\begin{aligned} f(x) &= m \cdot \left\{ m - \sum_{j=1}^{m} s_j \right\} \\ &= m \cdot \left\{ m - \sum_{j=1}^{m} \sum_{T(P_i)=T_j} L(P_i) \right\} \end{aligned} \quad (4)$$

The Fitness value can also be the value of the above objective function, which is described as following:

$$F(X) = f(X). \quad (5)$$

6 Experiment and Result Analysis

6.1 Experimental Environment

The experiments reported here demonstrate a variety of changes before and after using the multithreading solution to multimedia packets. Experimental environment consists of three computers which are configured as follows:

CPU: Intel Core i7 5960X (16 threads);

Memory: 8 GB DDR4;

OS: ubuntu-18.04.1.

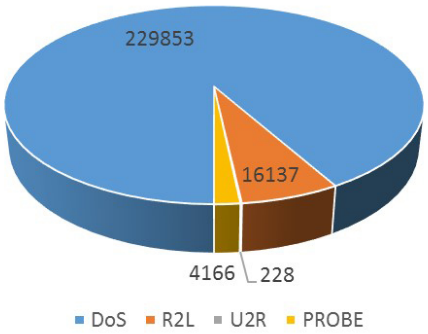

Figure 2: The types and quantities of attack category in attack traffic

Experimental data is a mixture of MIT Lincoln Laboratory KDD CUP 99 data sets and the background traffic. KDD CUP 99 data sets include four types of network attacks [22], DoS, R2L, U2R and PROBE. The types and quantities of attack category are shown as follows:

Background traffic is the real flow captured in the network, containing a large number of multimedia packets. In the experiment, the background traffic is sent by the first computer. As the attacker, the second computer uses Lincoln Laboratory KDD CUP 99 data set and IDS Informer to generate attack traffic. Both mixed traffic are sent to test NIDS installed on the third computer, as is shown in Figure 3.

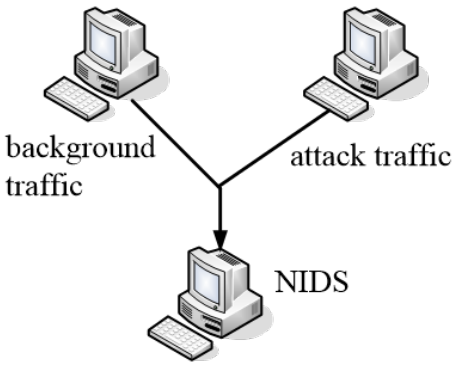

Figure 3: Experimental environment *** Please redraw this figure. Part of the icons (pictures) maybe have copyright problem. (Please don't use the commercial picture or icons). Please use a "rectangle with text" to replace the icons. ***

In order to complete each test successfully, we add the program to the preprocessor (spp_stream4 file) of NIDS for record the danger coefficient of multimedia packets which is selected into different threads. In addition, we improve the transmission speed of the mixed traffic so as to obtain experimental results in the case of packet loss.

All kinds of multimedia packet information in background traffic are shown in **Table 2.**

Figure 4-6 are analyses of the background traffic. As can be seen from the figure, the number of documents such as ASP and ASPX is the largest, exceeding 1200. On the total amount of data, the sum of JS files is the largest, reaching 3MKB. On the average detection length, the SWF file is the longest, with an average of more than 200KB.

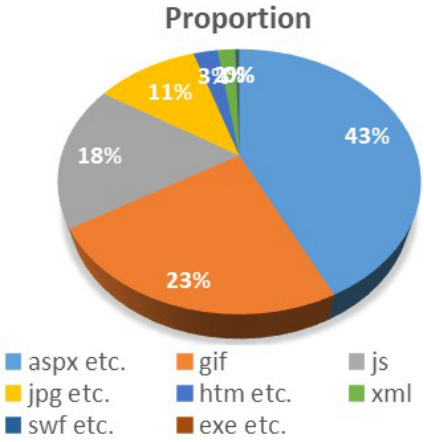

Figure 4: The proportion of different types of multimedia files

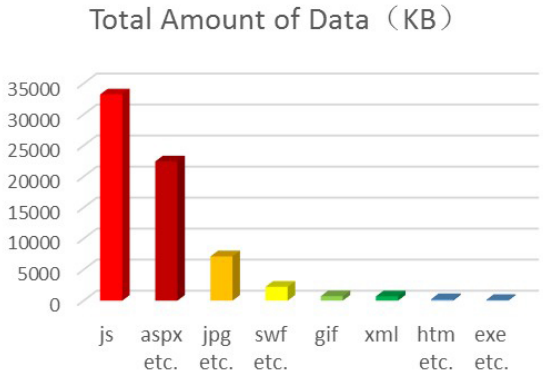

Figure 5: The total amount of data of different types of multimedia files

6.2 The Sum of Danger Coefficient in Different Threads

The first experiment is to compare differences in the sum of danger coefficients of the multimedia packets which are selected into each thread at the same time slice when using different methods.

As can be seen from Figure 7, the sum of danger coefficients of the multimedia packets in each thread has obviously increased after using the simple GA-based multithreading solution. The reason for this improvement is that these multimedia packets are selected randomly into each thread and its danger coefficient is not a consideration before using the GA-based solution. Therefore, the sum of danger coefficients is low and random. Nevertheless this value is relatively higher and stable after using the GA-based solution. This can also be shown by the sample variance of results. According to the following

Table 2: All kinds of multimedia packet information in background traffic

MIME Type	File Type	number	Total Amount of Data(KB)	Average Length(KB)	Risk factor
application/octet-stream	exe bin rar etc.	3	121	40	3.0
x-javascript	Js	545	33245	61	2.5
text/html	htm html hts etc.	81	243	3	1.6
application/x-asap etc.	asp aspx jsp etc.	1320	22440	17	1.6
text/xml application/xml	xml	59	708	12	1.3
image/jpeg	Jpz jpg jpeg	340	7140	21	1.5
image/gif	gif	728	728	1	1.5
x-shockwave-flash	swf swfi	10	2250	225	1.8

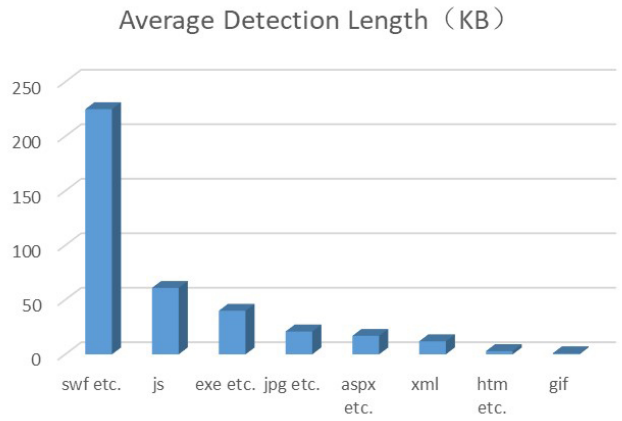

Figure 6: Average detection length for different types of multimedia files

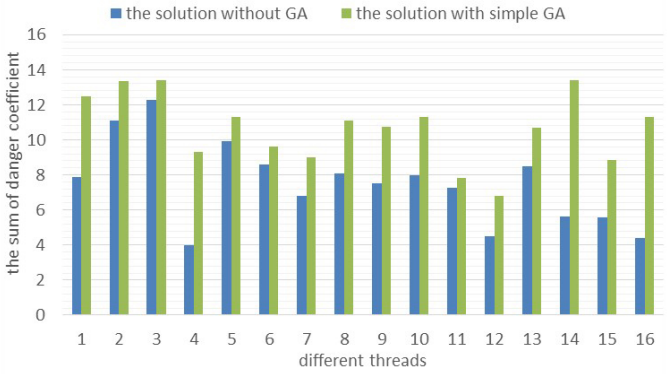

Figure 7: The differences in the sum of danger coefficient between using GA and not using GA

Figure 8: The differences in the sum of danger coefficient between using simple GA and hybrid GA

calculation formula of sample variance, sample variance is 6.5 before improvement, while it is 4.2 after improvement.

$$S^2 = \frac{\sum_{i=1}^{n} (x_i - E(x))^2}{n-1} \quad (6)$$

Figure 8 shows the differences in the sum of danger coefficient between using simple GA and hybrid GA. As you can see in the Figure 8, their difference is not obvious. In 69% of the threads, the sum of danger coefficient caused by using hybrid GA is more than that caused by simple GA.

6.3 The Packet Loss Rate

The second experiment is to compare the packet loss rate in three different situations.

As can be seen from Figure 9, after using simple GA or hybrid GA solutions, the packet loss rate increases slightly compared with that before improvement. The main reason for this problem is the high time complexity of GA. In addition, it is also found that the packet loss rate of the hybrid GA solution is lower than that of the simple GA solution. The main reason is that some invalid chromosomes would be generated in initialization and evolutionary process of the simple GA solution. In the distribution scheme represented by these invalid chromosomes, the sum of load of the multimedia packets in a thread will exceed the load of this thread. This leads to lower operation efficiency and increased packet loss rate of NIDS. Nevertheless, the advantage is that Solutions using hybrid GA can maintain packet loss rates within 6% higher than those without GA.

6.4 The Detection Number of Different Types of Multimedia Packets

The third experiment is to compare differences between the detection numbers of various types of multimedia packets with three different solution in the case of packet

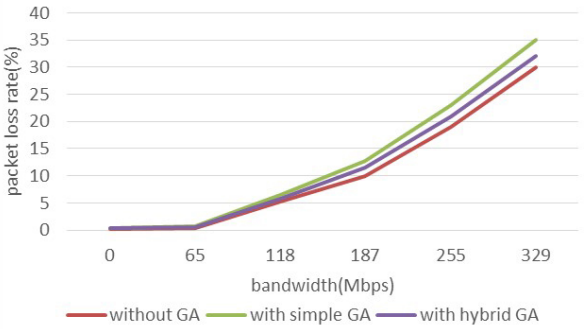

Figure 9: The packet loss rate caused by three different solutions

loss.

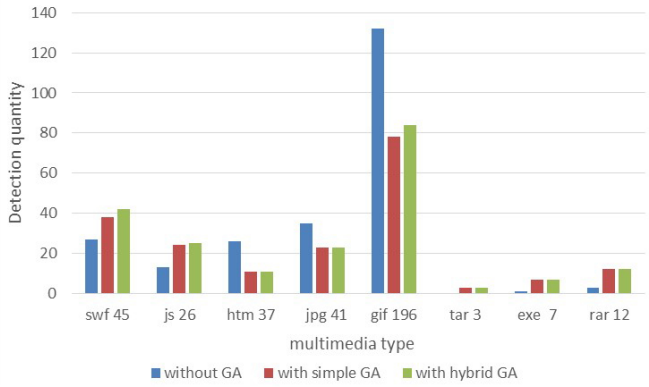

Figure 10: The detection number of different types of multimedia packets

As shown in Figure 10, when using the solution without GA, because all multimedia packets have been selected according to the time sequence they are captured, for all multimedia types, the more the numbers, the more the detection number, such as gif. While after using the solution with simple GA, because NIDS select multimedia packets according to its danger coefficient, the more danger, the more the detection number. For example, the detection number of exe type has increased by 6 times, all exe file are detected. On the other hand, the detection number of multimedia packets with lower danger coefficient has decreased. For instance, the detection number of gif type decreases by 41%. If the solution with hybrid GA is adopted, this trend of improvement will continue. Because the hybrid GA solution is superior to simple GA in controlling thread load capacity, the detection number of multimedia types with large number is slightly higher than that of simple GA.

6.5 The Detection Rate of Dangerous Incident

The fourth experiment is used to test the detection rate for dangerous incident by using three different solutions. The types and numbers of attack flow as shown in Table 3.

These experiments show that, when the network traffic exceeds the processing capacity of NIDS, the limited processing power of NIDS can be focused on the more dangerous multimedia packets by using the multithreading solution.

Table 3: The types and numbers of attack flow

Attack Type	DoS	R2L	U2R	PROBE	Total
Numbers	229853	16137	228	4166	250384

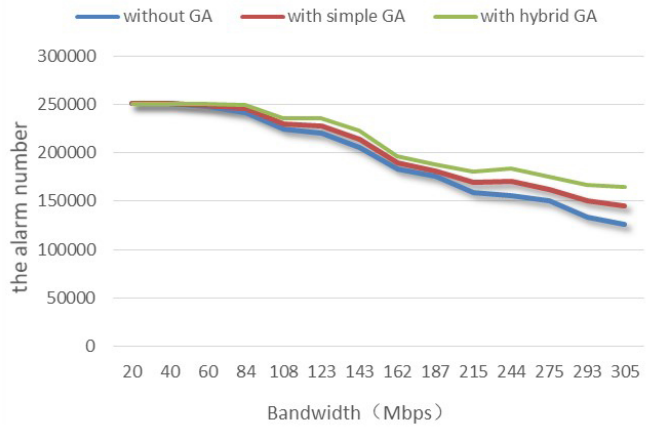

Figure 11: The detection rate using three different solutions

7 Conclusion and Future Work

With the high development and the wide application of network technology, network invasion is becoming an increasingly serious problem for network engineers and managers. Intrusion detection becomes a critical component of network security administration.

This paper addresses the performance challenges of NIDS in high-speed networks by proposing the multithreading solution based on genetic algorithm. By using this solution, when the network traffic is too great and omission is inevitable, NIDS can choose more dangerous multimedia packets for processing within the maximum processing capacity of different threads. Various experiments have shown that the solution can effectively improve the detection number of dangerous multimedia packets.

As for future work, firstly, I will apply this solution to non-multimedia files, such as bat type. Secondly, I will continue to compare the advantages and disadvantage of the solution proposed in the paper with those of other experiments to achieve more objective evaluations in the higher speed network environment.

Acknowledgment

This work is supported by Shaanxi Science and Technology Project (2019KRM153), Xi'an Science and Technol-

ogy Bureau (201805030YD8CG14(8)), Xi'an Beilin District Science and Technology Bureau (GX1708), Shaanxi Education Science Project(SGH18H089).

References

[1] Amrita, K. K. Ravulakollu, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers, *International Journal of Network Security*, vol. 20, no. 1, pp. 41-55, 2018.

[2] A. M. V. Bharathy, A. M. Basha, "A multi-class classification MCLP model with particle swarm optimization for network intrusion detection," *S?dhan?*, vol. 42, no. 5, pp. 631–640, 2017.

[3] E. M. Boujnouni, M. Jedra, "New intrusion detection system based on support vector domain description with information gain metric," *International Journal of Network Security*, vol. 20, no. 1, pp. 25-34, 2018.

[4] A. M. Chandrashekhar, K. Raghuveer, "Amalgamation of K-means clustering algorithm with standard MLP and SVM based neural networks to implement network intrusion detection system," *Parasitology*, vol. 114, no. 2, pp. 159-73, 2014.

[5] N. Cleetus, K. A. Dhanya, "Multi-objective particle swarm optimization in intrusion detection," *Procedia Computer Science*, vol. 60, no. 1, pp. 714-721, Mar. 2015.

[6] Y. Danane, T. Parvat, "Intrusion detection system using fuzzy genetic algorithm," in *International Conference on Pervasive Computing (ICPC'15)*, 2015. ISBN: 978-1-4799-6272-3.

[7] M. Ghaffari, N. Ghadiri, "Ambiguity-driven fuzzy C-means clustering: How to detect uncertain clustered records," *Applied Soft Computing*, pp. 1-12, 2016.

[8] D. Y. Harvey, M. D. Todd, "Automated feature design for numeric sequence classification by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 1, 2014.

[9] E. Hodo, X. Bellekens, A. Hamilton, *et al.*, "Threat analysis of IoT networks using artificial neural network intrusion detection system," *Tetrahedron Letters*, vol. 42, no. 39, pp. 6865-6867, 2017.

[10] R. Kondaiah, B. Sathyanarayana, "Trust factor and fuzzy firefly integrated particle swarm optimization based intrusion detection and prevention system for secure routing of MANET," *International Journal of Computer Networks & Communications*, vol. 10, no. 1, pp. 13-33, 2018.

[11] W. Li, Z. M. Yang, Y. P. Chan, *et al.*, "A clustering algorithm oriented to intrusion detection," in *IEEE International Conference on Computational Science and Engineering*, pp. 862-865, 2017.

[12] I. V. M. D. Lima, J. A. Degaspari and J. B. M. Sobral, "Intrusion detection through artificial neural networks," in *Network Operations and Management Symposium*, pp. 867-870, 2008.

[13] A. Nezarat, "Distributed intrusion detection system based on mixed cooperative and non-cooperative game theoretical model," *International Journal of Network Security*, vol. 20, no. 1, pp. 56-64, 2018.

[14] S. Rastegari, "Evolving statistical rulesets for network intrusion detection", *Applied Soft Computing*, vol. 33, pp. 348-359, Mar. 2015.

[15] R. Vijayanand, D. Devaraj, B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304-314, 2018.

[16] X. Zhao, C. Wang, "The improvements to snort intrusion detection system," *Journal of Xi'an Polytechnic University*, vol. 21, no. 6, pp. 859-863, Nov. 2007.

[17] X. Zhao, "Optimization of dynamic programming to the multimedia packets processing method for network intrusion detection system," *International Journal of Security and Its Applications*, vol. 9, no. 11, pp. 35-46, 2015.

[18] X. Zhao, "Research on a structure of the multimedia list oriented network intrusion detection system," *International Journal of Security and Its Applications*, vol. 10, no. 12, pp. 53-68, 2016.

[19] X. Zhao, "Dynamic self-adapting multimedia data processing method based on snort," *Computer System Application*, vol. 20, no. 4, pp. 211-213, 2011.

[20] X. Zhao, "The optimization research of the multimedia packets processing method in NIDS with 0/1 knapsack problem," *International Journal of Network Security*, vol. 17, no. 3, pp. 351-356, 2015.

[21] O. Marques, P. Baillargeon, "A multimedia traffic classification scheme for intrusion detection systems," in *International Conference on Information Technology and Applications*, pp. 496-501, 2005.

[22] S. Zander, G. ArmitagePractical, "Machine learning based multimedia traffic classification for distributed Qos management," *Local Computer Networks*, pp. 399-406, 2011.

[23] X. Y. Zhang, "Research of intrusion detection system dataset-KDD CUP99," *Computer Engineering and Design*, vol. 31, no. 22, pp. 4809-4812, Jan. 2010.

Biography

Xu Zhao is an associate professor in the School of Computer Science, Xi'an Polytechnic University, Shannxi, China. He received the M.S. degree from Xi?an Electronic Technology University, Xi?an City, Shannxi Province, China in 2007. He has developed several methods to deal with multimedia packets for network intrusion detection systems and is currently working on new optimization method with the help of artificial intelligence. He has some projects in research supported by provincial funds. His research interest is Network Security.

Guangqiu Huang is a professor and PhD supervisor at

Xi'an University Of Architecture And Technology. His major research interests include information security. He has completed 48 important scientific research projects.

Mr Reza Mousoli is a School Director of Stakeholder Engagement of School of Law, Criminal Justice and Computing of Canterbury Christ Church University of UK. His research Interests include Cyber Security, e-safety, Privacy and Confidentially.