Hancitor malware recognition using swarm intelligent technique

Laheeb M. Ibrahim[1], Maisirreem Atheeed Kamal[2], AbdulSattar A. Al-Alusi[3]
[1,2]Department of Software Engineering, College of Computer Science & Math, University of Mosul, Mosul, Iraq
[3]College of Security and Global Studies, American University in the Emirates, Dubai, United Arab Emirates

Article Info

Article history:

Received Jun 6, 2020
Revised Mar 9, 2021
Accepted Apr 18, 2021

Keywords:

Artificial intelligent technique
Artificial bee colony algorithm
Gray wolves optimization
Hancitor malware
Malware recognition
Swarm intelligence

ABSTRACT

Malware is a global risk rife designed to destroy computer systems without the owner's knowledge. It is still regarded as the most popular threat that attacks computer systems. Early recognition of unknown malware remains a problem. swarm intelligence (SI), usually customer societies, communicate locally with their domain and with each other. Clients use very simple rules of behavior and the interactions between them lead to smart appearance, noticeable, individual behavior and optimized solution of problem and SI has been successfully applied in many fields, especially for malware ion tasks. SI also saves a considerable amount of time and enhances the precision of the malware recognition system. This paper introduces a malware recognition system for Hancitor malware using the gray wolf optimization (GWO) algorithm and artificial bee colony (ABC) algorithm, which can effectively recognize Hancitor in networks.

Corresponding Author:

Laheeb M. Ibrahim
Department of Software Engineering
College of Computer Science & Math
University of Mosul
Aljameia, Mosul, Iraq
Email: laheeb_alzubaidy321966@uomosul.edu.iq

1. INTRODUCTION

Malware is recognized as a program created to disable computer operation and access to private computer systems. When the malware is implemented, the malicious program designer takes advantage of the benefit of accessing the computer systems of the infected device, and collects personal information and everything that the device contains without the consent of the computer owner. Currently, malware is used to steal important commercial and banking information. It is usually used widely against government websites, corporate sites, and banks to collect protected information or disrupt its operation in general. Malware is usually used against individuals to obtain personal information, such as bank card numbers. The different types of malwares create damages due to their removal difficulty upon installation on the prey's machine. The severity of the software ranges from minor inconvenience to irreparable harm that requires reformatting the hard drive.

Malware is considered a great danger and threats stand up to the world of the Internet and computer networks today and these malwares usually come in various forms such as Trojan horse, Virus, Worm, Botnet, Spyware, and Adware [1]. Fire Eye reported that 47% of organizations had encountered breaches of the malware accident safety. Malware is constantly growing in size, diversity, and speed. Thus, malware has become complex and uses new and advanced methods to infect computers and smart phones [1].

Many techniques were used in malware classification and recognition [2]. A deep neural network uses for malware dynamic behavior recognition, in which deep neural network could recognise future malware through the generative adversarial network implementation. A Swarm intelligent technique used for recognition of malware as in [3], [4] where particle swarm optimization (PSO) is uses to build a system to recognise malware, in these studies the researcher's refinement the rate of recognition with PSO. Vinod and Dhanya [5] using PSO in the recognition of the algorithm and to improve the recognition rate. Vinod and Dhanya [5] recognize malware using statistical approach. Another approach for recognising malware used as in [6], where the researcher used a data mining method. The other researchers used machine learning for recognising malware [7]-[11].

Another study used a hybrid approach which enhances the performance to recognise unknown malware, recognizer proposed in [12]-[17], as [12], where it suggests a malware recognition system for Android system using concept of hybrid intelligent depended on support vector machine (SVM) with evolutionary algorithms (genetic algorithm (GA) and PSO) to enhance malware recognition, which is respectively referred to as Droid-HESVMGA and Droid-HESVMPSO, to increase the precision rate to recognize malware. The methods based on naive-bayes, SVM, and decision tree used as recognisers, are exhaustibles that boost decision [13]. Tree is a top method used as a recogniser of malware. A malware recognition method has been proposed using image processing methods, which depicts malware binary as gray scale images [14]. A K-nearest neighbor technique with Euclidean distance method is used for malware recognition. Firdausi *et al.* [15] proofed a concept of using hybrid intelligent to recognise malware based on 5 recognisers i.e., k-nearest neighbors (kNN), naive bayes, J48 decision tree, SVM, and multilayer perceptron (MLP) neural network. Santos *et al*. [16] built OPEM system, which used 4 algorithms as recognisers, these methods are Decision Tree, kNN, Bayesian network, and SVM to recognise unknown malware, a similar work is done by [17] to recognise the malware using SVM, IB1, DT and RF. Anderson *et al.* [18] proposed a method which used support vector machine recognizer.

Malware classification and recognition using swarm intelligent technique are significant areas in the recognition of malicious applications. The method used by malware recognize experts depends on the problem and dataset regardless of the categorical or numerical output data, therefore swarm intelligent techniques can be used for recognition, forecasting, and estimation malwares [3]. In this paper swarm intelligent (SI) algorithms are used to recognise Hancitor malware because SI are familiar in recognition and classification applications due to, they depended on simple idea and being accurate and easy concepts to implement, do not require progressive information and it is often used to solve a wide range of problems that cover many applications [19]. SI algorithms are divided in two categories depending on its type; the first type is the insect-based category for example, ant colony optimization (ACO), artificial bee colony (ABC) etc. The second category is animal-based algorithms which include PSO, artificial fish, and grey wolf optimization (GWO) etc [19], [20]. Two of swarm intelligent algorithms to recognising Hancitor malware uses in this paper depend on its categories, first for animal-based algorithms is GWO algorithm and second for insect-based category is ABC algorithm. GWO is used in this paper to recognize Hanictor malware due to the advantages of GWO by maintaining information on the search space overcome the course of iterations. The works in [19]-[21] uses memory to store the best solution obtained, contains some parameters to adjust and implemented in easy way [19] and GWO have capabilities to solve optimization problems [20]-[24] through the social hierarchy of GWO.

The second SI algorithm used to identify Hancitor malware is the ABC algorithm where ABC algorithm requires a minimum level of understanding of the problem area as it does not require complex training data as the bee recruiter better updates himself with the attribute correlation and update directly on the performance of the classification category than the knowledge of the waggle dance [25], [26]. Therefore, these types of procedures certainly have a greater potential in improving classification accuracy. In an ABC data classification, it can be a mimic behavior of insects to find the best food source, and build an ideal nest structure. The bees distribute the workload among themselves, which does not classify the data incorrectly and are homogeneous spectrum and spectral interference. Dancing behavior aids in optimal design. The Waggle dance is one of the mechanisms for sharing the existing food source, which indicates a good candidate for developing a new smart search for the optimal solution [25], [26]. The rest of the paper is arranged in the following style; in section two is theoretical background explores malware recognition techniques and Hancitor malware and its danger, while in section three the swarm intelligent techniques used in malware recognition, GWO and ABC algorithms are explained. Section four presents the proposed model, followed by the results of the comparison in section five, in section six the conclusion and future work.

2. MALWARE RECOGNITON TECHNIQUES

Hancitor sometimes called Tordal and Chanitor. The malware has been around since 2014. Hanictor attack to infect users' devices is malicious spam campaigns; Hancitor mostly gets into devices with Microsoft Office files. Once the user downloads and opens the malicious file, the malware either uses the lure to trick the victim into enabling macros or uses an exploit. After that Hancitor will be either downloaded from the C2 server or dropped from an Office file. The next step is its execution during which the malware downloads the main payload, usually a Trojan such as Pony, Vawtrak, or DELoader. Hancitor method of infecting the victim's machine using many ways, one of these ways is using .DOC attachments taking advantage of Microsoft's dynamic data exchange (DDE) technique. The user must first download the file and then activate macros, ignoring multiple security warnings [27]. Malware authors use lures to trick users into doing that. Some phishing emails contain an invoice or a fake payment related document, trying to make the user download it. In addition, attackers provide instruction to enable macros. If the user complies, malicious macros will download Hancitor or it will be dropped from the document. In some malspam campaigns, Hancitor was delivered to victims with .RTF documents which used an exploit to run the PowerShell command which downloaded the loader to the computer. Another way of Hancitor to infecting the victim's machine is by using Excel spreadsheets as a trap document since December 17, 2018, the executable file (Hanictor) was instilled in Excel spreadsheets, then Hancitor declining to a vulnerable Windows host after opening the spreadsheet in Excel and enabling macros on January 28, 2019. However, the Hancitor campaign changed its decoy document on February 5, 2019. This campaign went back to using Word documents instead of Excel spreadsheets. The Hancitor executable was retrieved from a web server hosted on the same IP address (but a different domain) as the initial Word document after enabling macros. The Hancitor infections on February 5, 2019, is shown in Figure 1.

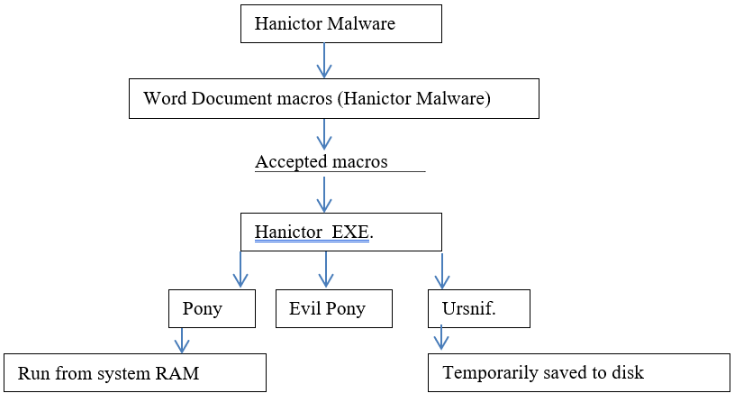

Figure 1. Hancitor malware infection on February 5, 2019

When Hancitor initially infects a system, it sends a POST request to its command and control (C&C) server with information on the infected system. Figure 2 show the Hancitor malware Infected system.

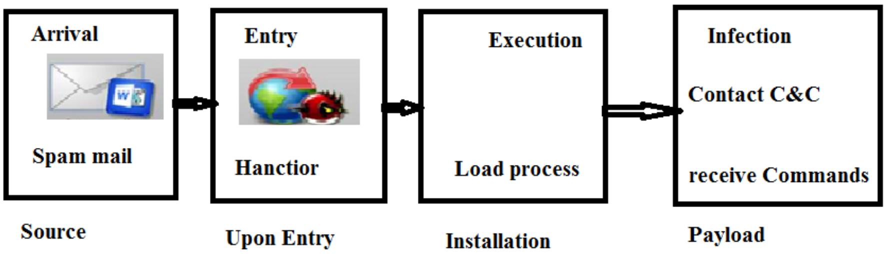

Figure 2. Hancitor malware

3. WHY SWARM INTELLIGENCE (SI)

SI is a branch of artificial intelligence (AI) branches and is specified by Gerardo Beni and Jing Wang in 1989. There are many purposes accountable for using SI algorithms based on flexibility, ease of use, speed in implementation, versatility, the self-learning capability and adaptability to external variations. Also, SI used to solve nonlinear problems in real-world applications in sciences, engineering, in bioinformatics applications, recognition and classification technologies, and network security [19]-[21].

3.1. Grey wolf optimization (GWO)

GWO algorithm is suggested by [20], [22], [23] relying on social hierarchy and hunting habits for gray wolves to find victims. There are four hierarchy types of individuals in Community of GWO these types are alpha, beta, delta, and omega based on their fitness. Where the research process is done by designing a model to mimic the hunting behavior of gray wolves through searching for prey, attacking the prey and covering exploitation. In GWO algorithm the hunting method helps to determine the location of prey [20]. The mathematical simulation of the GWO is explained in [28], and the GWO algorithm for Hancitor infection malware recognition is presented follows.

GWO algorithm [20]

```
Initialize the parameters a, A, C
Initialize GWO population GWOi (i = 1, 2... m)
For Each search agent must calculate the fitness for it
  GWOα= best search agent
  GWOβ= second best search agent
  GWOδ= third best search agent
  While (i < Maximum No. of iterations)
    For each search agent
        Upgrade the position of the current search agent by
        GWO (i + 1) = $(GWO_1 + GWO_2 + {}^{\rightarrow} GWO_3) / 3$
    End for
    Upgrade a, A, and C
    Compute the fitness of all search agents
    Upgrade GWO α, GWO β, and GWO δ
    i++
  End while
  Return GWO α
```

3.2. Artificial bee colony (ABC)

The ABC algorithm is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm, proposed by Derviş Karaboğa in 2005, the colony into the ABC algorithm contain from three types of bees: employed bees, onlookers and scouts. In ABC they are simulated to only one artificial bee for each food source, where the numbers of bees used in the colony are equal to the number of food sources around the hive. In ABC, employed bees go to their food source, return to the hive and dance in this region. Employed bees whose food source has been abandoned become scouts and begin to search for a new source of food. Onlookers watch the dances of employed bees and choose food sources according to the dances.

ABC differ from another swarm intelligence algorithms based on the situation that the potential solutions are appear by the food sources, not the individuals in the population. The quality of the potential solution is presented as a fitness value; the fitness value is calculated by the value of the objective function of the problem. In the ABC algorithm onlookers and employed bees carry out the exploitation process in the search space, while the scouts control the exploration process. The phases of ABC algorithm and mathematical equations are in [29]. Pseudo-code of the ABC algorithm for constrained optimization problems [30] is:

- Initialize the population of solutions and evaluate the population, where xi (i = 1, 2, SN) is a D-dimensional vector present the solution, SN represent the size of the population.
- Initialize value of cycle by 1
- Do while cycle not equal MCN
- Calculate new solutions for the employed bees, as in (1)

$$v_{i,j} = \begin{cases} x_{i,j} + \emptyset_{i,j} \times \left(x_{i,j} - x_{k,j}\right), R_j < \mathrm{MR} \\ \quad x_{i,j}\ otherwise \end{cases} \tag{1}$$

Where k is a random number (1 - SN/2) and different from *i*.

- Calculate selection operation depends on Deb's method [31].
- Constraint violations (CV), as in (2) and calculate the probability values (pi) for xi using fitness of the solutions, as (3).

$$CV = \sum_{g_j > 0} g_j\ (x) + \sum_{q+1}^{m} h_j\ (x) \tag{2}$$

$$Pi = \begin{cases} 0.5 + \left(\frac{fitness_{\ i}}{\sum_{i=1}^{SN} fitness_{\ i}}\right) \times 0.5\ if\ solution\ is\ feasible \\ \left(1 - \frac{CV}{\sum_{i=1}^{SN} CV}\right) \times 0.5\ if\ solution\ is\ infeasible \end{cases} \tag{3}$$

- Product a new solution vij, as in (1) for each onlooker bee in the neighbourhood of the solution selected depending on pi and evaluate it
- Calculate selection operation the value between (υi - xi) depended on Deb’s method
- Use “limit” parameter for the scout to decide the disused solutions. if they exist, replace them with new randomly produced solutions, as in (4)

$$x_{ij}\ = lb_j + rand(0,1) \times \left(ub_j - \ lb_j\right) \tag{4}$$

- Store the best solution completed up till now
- cycle = cycle+1
- End do

4. PROPOSED MODEL

In this paper a recogniser system is designed to recognise Hancitor traffic from normal traffic, to allow the network administrator to make the appropriate decision, See Figure 3. In proposed model a Hancitor method of infecting the victim's machine is using .DOC attachments taking advantage of Microsoft’s dynamic data exchange (DDE) technique, netflow traffic analyzer (NTA) tool Ver. 9 used to collect network traffics by capturing them and obtaining the data used in the proposed model. The captured traffics (normal or Hancitor malware traffics) are then stored in the Hancitor data file. The collected traffics are used for the selection of some features related to the underlying network traffic to select the following attributes (Source IP, Destination IP, Protocol, Timeline, and Length), see Table 1. After the selection of attributes, the monitored traffics are used as input to the recogniser (GWO and ABC algorithms) to recognise the traffic into Hancitor or normal traffic. GWO and ABC swarm intelligent algorithms are used to recognise Hancitor traffics on the network using the attributes. The underlying conveyances and the parameters used in the tests are significant. The parameters of the gray-wolf optimizer are chosen after running a few tests to obtain satisfactory outcomes. GWO was initialized with:

- Population of GWO: Number of gray wolves = 20.
- Maximum iteration number = 150.
- The values of a = (2 to 0)

The fitness function as given by Euclidean distance, as in (5) is calculated to have best solution after first iteration as “α-wolf”, and “β” and “δ” wolves, and the second and third best solution are β and δ

$$D = \sqrt{\sum_{i=1}^{n}(q_i - \ p_i)\ \ ^2} \tag{5}$$

The parameters of the ABC algorithm are chosen after running a few tests to obtain satisfactory outcomes. ABC was initialized with:

- Population size of ABC: Employed bees = 30.
- Maximum iteration number = 100.
- Limite = 30

The fitness function as given by Euclidean distance, as in (5) is calculated to have best solution after first iteration.

To measure the performance of the GWO and ABC algorithms to recognize Hancitor malware two equations are used. [4], [16], these equations are Accuracy of Recognition packets used to calculate the percentage of correctly classified packets (normal or Hancitor packet) among the total number of packets is computed, as in (6) and false alarm rate (FAR) where FAR is referred to as the false positive rate (FPR) or sensitivity, as in (7):

$$ACC = \frac{TN+TP}{TN+FP+FN+FP} \tag{6}$$

$$FAR = \frac{FP}{TN+FP} \quad (7)$$

where true positive (TP): correctly identified Hancitor malware; true negative (TN): incorrectly identified Hancitor malware; false positive (FP): correctly rejected Hancitor malware; and false negative (FN): incorrectly rejected Hancitor malware.

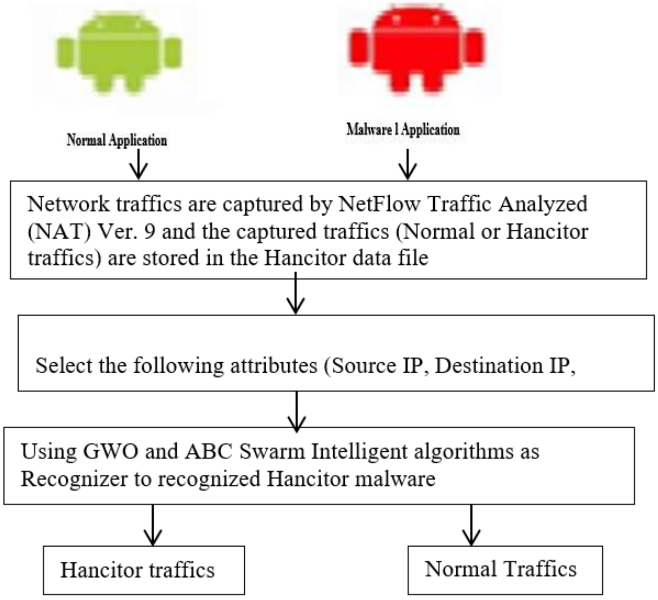
Normal Application
Malware l Application
Network traffics are captured by NetFlow Traffic Analyzed (NAT) Ver. 9 and the captured traffics (Normal or Hancitor traffics) are stored in the Hancitor data file
Select the following attributes (Source IP, Destination IP,
Using GWO and ABC Swarm Intelligent algorithms as Recognizer to recognized Hancitor malware
Hancitor traffics
Normal Traffics

Figure 3. Proposed model

Table 1. Derived statistic attributes from the captured file

Feature name	Feature contents
Length	Size of the whole packet include header, trailer and data that send on the packet
Snapshot(length)	The amount of data for each frame that actually captured by the network capturing tool and stored into the capture file
Last packet elapse	Capture time and duration of the last packet
Packet	The number of protocol packet from the total capture packet
Time span/s	Is the time between the first and last packet
Average App	Average app: information about NetFlow Traffic Analyzer hardware.
Average size	The average size of the header on the packet
Bytes	The number of protocol bytes from the total capture packets
Average Byte/s	The average number of protocol bytes from the total capture packets
Average Bits/s	The average bandwidth of this protocol in relation to the capture time

5. EXPERIMENTAL RESULT

The proposed recognition model for Hancitor malware was implemented in MATLAB version R2015a. The proposed model performed GWO and ABC algorithms on the set of packets stored in the Hancitor data file to recognise Hancitor traffics. Two experiments were conducted in the proposed work based on two types of packets:

- Packets based on statistic attributes (Length, Packet size limit, Elapsed, Packets, Time span(s), Average ppsmm Average packet size (B), Bytes, Average bytes/s, Average bits/s), see Table 2, the size of packets based on statistic attributes are 3000 packets.
- Packets based on IPv4 characteristics, see Tables 3 and 4, the size of packets based on IPv4 characteristics are 3000 packets.

Table 2. Packets based on statistic attributes from the captured traffic (normal and Hancitor packet)

Items	Hancitor traffic	Normal traffic
Length	1272kb	1073 MB
Packet size limit	65535byte	4096 bytes
Elapsed	01:26:09	00:35:16
Packets	6489	4198011
Time span, s	5169.785	2116.140
Average pps	1.3	1983.8
Average packet size, B	180	240
Bytes	1168236	1006573119
Average bytes/s	225	475 k
Average bits/s	1807k	3805 k

Table 3. Derived attributes from the captured file using IPv4 characteristics (normal packet)

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
All Addresses	1286	2.9907	100%	6.9900	0.000
64.4.23.143	1	0.0023	0.08%	0.0100	0.250
224.0.0.10	3	0.0070	0.23%	0.0200	0.210
192.168.28.254	10	0.0233	0.78%	0.0800	0.120
192.168.28.202	42	0.0977	3.27%	0.2800	0.120
192.168.27.253	80	0.1860	6.22%	0.3100	0.150
192.168.27.25	7	0.0163	0.54%	0.0300	0.220
192.168.27.203	2	0.0047	0.16%	0.0200	0.320
192.168.27.202	5	0.0116	0.39%	0.0300	0.320
192.168.27.152	3	0.0070	0.23%	0.0100	0.000
192.168.27.103	2	0.0047	0.16%	0.0100	0.000
192.168.27.102	6	0.0140	0.47%	0.0300	0.220
192.168.27.101	6	0.0140	0.47%	0.0400	0.240
192.168.27.100	179	0.4163	13.92%	0.4600	0.310
192.168.26.254	1	0.0023	0.08%	0.0100	0.100
192.168.26.253	8	0.0186	0.62%	0.0400	0.240
192.168.26.252	8	0.0186	0.62%	0.0400	0.240
192.168.26.25	4	0.0093	0.31%	0.0400	0.230
192.168.26.203	6	0.0140	0.47%	0.0400	0.240
192.168.26.202	2	0.0047	0.16%	0.0200	0.310

Table 4. Derived attributes from the captured file using IPv4 characteristics (Hancitor packet)

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
All Addresses	6489	0.0013	100%	0.3800	266.055
82.196.9.45	283	0.0001	4.36%	0.0200	607.306
8.8.8.8	280	0.0001	4.31%	0.0300	3148.741
54.243.123.39	11	0.0 000	0.17%	0.0400	251.282
51.255.48.78	423	0.0001	6.52%	0.0200	499.704
47.52.45.178	378	0.0001	5.83%	0.0800	858.004
47.254.199.192	347	0.0001	5.35%	0.0900	54.971
46.235.47.59	361	0.0001	5.56%	0.3800	266.055
213.136.85.253	700	0.0001	10.79%	0.0200	347.922
207.148.83.241	280	0.0001	4.31%	0.0200	579.307
193.183.98.66	283	0.0001	4.36%	0.0200	578.944
192.71.245.208	283	0.0001	4.36%	0.0200	778.826
191.101.20.16	460	0.0001	7.09%	0.0600	262.657
178.17.170.179	338	0.0001	5.21%	0.0400	1155.896
159.89.249.249	283	0.0001	4.36%	0.0200	606.800
142.4.205.47	700	0.0001	10.79%	0.0200	335.869
111.67.20.8	283	0.0001	4.36%	0.0200	580.004
103.236.162.119	356	0.0001	5.49%	0.0600	582.692
10.12.6.101	6489	0.0013	100.00%	0.3800	266.055
10.12.6.1	440	0.0001	6.78%	0.0600	747.343

Table 5. Explains the accuracy of the GWO and ABC algorithms considering recognition packets (normal or Hancitor packet) for the two types of data

Swarm Intelligent Algorithms	Packets	Total packets	False Alarm Rate%	Accuracy %
GWO	Packets based on statistic attributes	3000	4.2%	95.8%
	Packets based on IPv4 characteristics	3000	8%	92%
ABC	Packets based on statistic attributes	3000	2.8%	97.2%
	Packets based on IPv4 characteristics	3000	5.7%	94,3%

Table 5. Accuracy of GWO and ABC Algorithms for recognition Hancitor malware, After performing experiments on two types of data packets using the GWO algorithm to recognise Hancitor malware, the Experimental result shows that using data based on attributes is better than using data based on IPv4 characteristics to recognise Hancitor malware, see Figure 4, and also after performing experiments using ABC algorithm on two types of data packets to recognise Hancitor malware, the Experimental result also show that using data based on attributes is better than using data based on IPv4 characteristics to recognise Hancitor malware, see Figure 5.

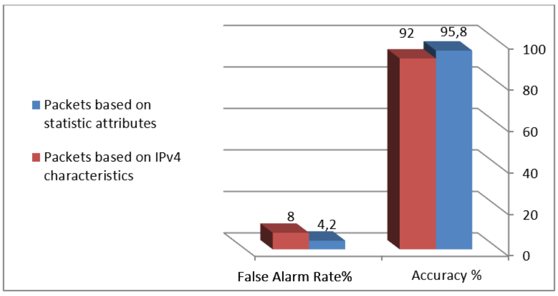

Figure 4. Recognition accuracy result for Hancitor using the GWO algorithm

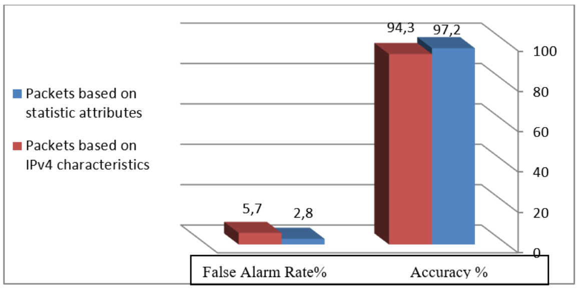

Figure 5. Recognition accuracy result for Hancitor using the ABC algorithm

From Table 5 and Figure 6, we note that the accuracy rate of the recognition of Hancitor malware using ABC algorithm is better than GWO algorithm in the two types of data Packets based on statistic attributes and Packets based on IPv4 characteristics, and also the False Alarm Rate in ABC algorithm is lower than GWO for two types of data. ABC algorithm is better than GWO algorithm to recognise Hancitor malware in spite of our use of an equal number of packets and the same data because ABC algorithm does not require a complex data training, whether it is simple or complex training data, but rather requires an understanding of the minimum understanding of the problem area that helped in accuracy and speed of discrimination.

Using SI in recognise Hancitor Malware have big advantages, becuase SI presents similar intelligent collective behavior, where SI provides intelligent solutions to problems by the self-organization and communication between individuals in the swarm, and the seamless coordination of all individual activities does not require supervisor. GWO and ABC algorithms used to detect correctly a Hancitor malware in a fast and high recognition rate because GWO and ABC swarm intelligence algorithms have a big advantage, where ABC algorithm is a simplest swarm intelligence algorithm and delivers highly accurate results for optimization problems with levels ranging from simplicity to complexity, and it proved that the ABC algorithm is the best choice for solving Hanictor malware problems and can be applied to many applications, also GWO was identified to be sufficient competitive with other state-of-the-art met heuristic methods to recognised Hanictor malware, it achieves better performance.

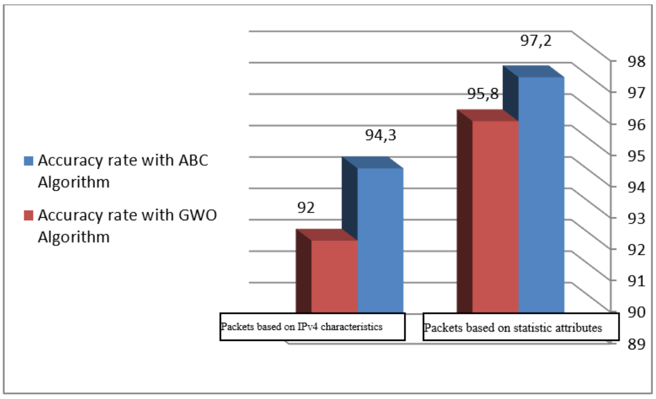

Figure 6. Recognition accuracy result for Hancitor using the GWO and ABC algorithms

6. CONCLUSION

A new model uses GWO and ABC to recognise Hancitor malware behaviors is proposed. It can protect users from Hancitor malware attacks. In this research GWO and ABC have ability to recognise correctly a Hancitor malware in a fast and precision recognition rate. ABC and GWO gives good results for recognize the existence of Hancitor in the network with accuracy of 79.2% by using ABC and 95.8% using GWO for data depend on static attributes, for the second type of data depend on IPv4 characteristics the recognition rate is 94.3% by using ABC and 92% by using GWO. The prediction of percentage of infection has good performance using ABC better than GWO with data depend on static attributes better than data depend on IPv4 characteristics.

REFERENCES

[1]. E. Gandotra, D. Bansal, and S. Sofat, “Malware Analysis and Classification: A Survey,” in *Journal of Information Security*, vol. 5, no. 2, pp. 56-64, 2014, doi: 10.4236/jis.2014.52006.

[2]. S. Lu *et al.*, “New Era of Deep learning Based Malware Intrusion: The Malware Detection and Prediction Based on Deep Learning,” *Computer Science, Cryptography and Security*, 2019. [online] Available: https://arxiv.org/abs/1907.08356

[3]. O.S. Adebayo and N. A. Aziz, “Improved Malware Detection Model with A priori Association Rule and Particle Swarm Optimization,” *Hindawi Security and Communication Networks*, vol. 2019, pp. 1-13, 2019, doi: 10.1155/2019/2850932.

[4]. T. S. Sobh, “Hybrid Swarm Intelligence and Artificial Neural Network for Mitigating Malware Effects,” *Recent Patents on Computer Science*, vol. 7, no. 1, pp. 38-53, 2014**.**

[5]. M. V. Varsha, P. Vinod and K. A. Dhanya, “Identification of Malicious Android App using Manifest and Op-code Features,” *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 2, pp. 125-138, 2017, doi: 10.1007/s11416-016-0277-z.

[6]. W. W. Cohen, “Fast Effective Rule Induction,” in *Proceedings of 12th International Conference on Machine Learning, San Francisco*, 1995, pp. 115-123, doi: 10.1016/B978-1-55860-377-6.50023-2.

[7]. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp.10-18, 2009, doi: 10.1145/1656274.1656278.

[8]. I. Santos, J. Nieves, and P. G. Bringas, “Semi-Supervised Learning for Unknown Malware Detection,” *International Symposium on Distributed Computing and Artificial Intelligence Advances in Intelligent and Soft Computing*, 2011, vol. 91, pp. 415-422, doi: 10.1007/978-3-642-19934-9_53.

[9]. M. Siddiqui, M. C. Wang, and J. Lee, “Detecting Internet Worms Using Data Mining Techniques,” *Journal of Systemic, Cybernetics and Informatics*, vol. 6, no. 6, pp. 48-53, 2009.

[10]. K. Rieck, P. Trinius, C. Willems, and T. Holz, “Automatic Analysis of Malware Behavior Using Machine Learning,” *Journal of Computer Security*, vol. 19, no. 4, pp. 639-668, 2011, doi: 10.3233/JCS-2010-0410.

[11]. T. Lee, “Behavioral Classification,” *Proceedings of the European Institute for Computer Antivirus Research Conference (EICAR’06),* 2006.

[12]. W. Ali, “Hybrid Intelligent Android Malware Detection Using Evolving Support Vector Machine Based on Genetic Algorithm and Particle Swarm Optimization,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 19, no. 9, pp. 15-28, 2019.

[13]. J. Z. Kolter, and M. A. Maloof, “Learning to Detect Malicious Executable in the Wild,” in *Proceedings of the* 10*th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 470-478, doi: 10.1145/1014052.1014105.

[14]. L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath “Malware Images: Visualization and Automatic Classification,” *Proceedings of the* 8*th International Symposium on Visualization for Cyber Security*, 2011, pp. 1-7, doi: 10.1145/2016904.2016908.

[15]. I. Firdausi, C. lim, A. Erwin and A. S. Nugroho, “Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection,” *2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies,* 2010, pp. 201-203, doi: 10.1109/ACT.2010.33.

[16]. I. Santos, J. Devesa, F. Brezo, J. Nieves, and P. G. Bringas “OPEM: A Static-Dynamic Approach for Machine Learning Based Malware Detection,” *Proceedings of International Conference CISIS’12-ICEUTE’12, Special Sessions Advances in Intelligent Systems and Computing*, 2013, vol. 189, pp. 271-280, doi: 10.1007/978-3-642-33018-6_28.

[17]. R. Islam, R. Tian, L. M. Batten, and S. Versteeg, “Classification of Malware Based on Integrated Static and Dynamic Features,” *Journal of Network and Computer Application*, vol. 36, no. 2, pp. 646-556, 2013, doi: 10.1016/j.jnca.2012.10.004.

[18]. B. Anderson, C. Storlie, ans T. Lane, “Improving Malware Classification: Bridging the Static/Dynamic Gap,” *Proceedings of* 5*th ACM Workshop on Security and Artificial Intelligence* (*AISec*), 2012, pp. 3-14, doi: 10.1145/2381896.2381900.

[19]. H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, “Grey wolf optimizer: a review of recent variants and applications,” *Neural Computing & Applications*, vol. 30, pp. 413-435, 2018, doi: 10.1007/s00521-017-3272-5.

[20]. S. Mirjalili, S. M. Mirjalili, and A. Lewis, “A. Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007

[21]. A. Chakraborty and A. K. Kar, “Swarm Intelligence, A Review of Algorithms,” *Nature-Inspired Computing and Optimization*, vol. 10, pp. 475-494, 2017, doi: 10.1007/978-3-319-50920-4_19.

[22]. R.-E. Precup, R.-C. David, A.-I. Szedlak-Stinean, E. M. Petriu, and F. Dragan, “An Easily Understandable Grey Wolf Optimizer and Its Application to Fuzzy Controller Tuning,” *Algorithms*, vol. 10, no. 2, pp. 1-15, 2017, doi: 10.3390/a10020068.

[23]. K. Murali and J. T., “Automated Image Enhancement Using Grey-Wolf Optimizer Algorithm,” *IIOAB Journal*, vol. 7, no. 3, pp. 77-84, 2016.

[24]. Y. Tan and Y. Shi, “Advances in Swarm Intelligence,” *10th International Conference, ICSI 2019, Chiang Mai, Thailand, July 26–30, 2019, Proceedings*, 2019.

[25]. C. Zhang, D. Ouyang, and J. Ning, “An artificial bee colony approach for clustering,” *Expert Syst. Appl*. vol. 37, no. 7, pp. 4761-4767, 2010, doi: 10.1016/j.eswa.2009.11.003.

[26]. C. Xu, H. Duan, and F. Liu, “Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning,” *Aerosp. Sci. Technol*., vol. 14, no. 8, pp. 535-541, 2010, doi: 10.1016/j.ast.2010.04.008.

[27]. Hancitor malspam and infection traffic from Tuesday 2019-02-05, February 6th, 2019. [Online]. Available: https://isc.sans.edu/forums/diary/Hancitor+malspam+and+infection+traffic+from+Tuesday+20190205/24616/

[28]. L. Qiang *et al.*, “An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis,” *Computational and Mathematical Methods in Medicine*, vol. 2017, pp. 1-15, 2017, doi: 10.1155/2017/9512741 020068.

[29]. S. Sharma and P. Bhambu, “Artificial Bee Colony Algorithm: A Survey,” *International Journal of Computer Applications*, vol. 149, no. 4, pp. 11-19, 2016.

[30]. D. Karaboga and B. Basturk, “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems,” *Proc. IFSA 2007, LNAI* 4529, 2007, pp. 789-798, doi: 10.1007/978-3-540-72950-1_77.

[31]. K. Deb, “An Efficient Constraint Handling Method for Genetic Algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311-338, 2000, doi: 10.1016/S0045-7825(99)00389-8.