COMPUTATIONAL RESEARCH PROGRESS IN APPLIED SCIENCE & ENGINEERING (CRPASE)

CRPASE: TRANSACTIONS OF
ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING

Journal homepage: http://www.crpase.com

CRPASE: Transactions of Electrical, Electronic and Computer Engineering 7 (1) Article ID: 2320, 1–4, March 2021 ISSN 2423-4591

Research Article

Provide A Way To Improve Malware Detection Using Backup Vector Machine And Cuckoo Algorithm

Reza Molaee Fard*
Department Of Computer Engineering - Dezful Branch of Azad University, Khuzestan, Iran

Keywords

Malware Detection,
Support Vector Machine,
Cuckoo Algorithm

Abstract

Today, in spite of the existing malware, the number of which is increasing day by day, it seems necessary to have a solution that can deal with this malware and can detect these malware. This study provides a way to improve the detection of malicious malware that causes damage to the software system. By first categorizing our data using a backup vector machine, we then optimize the data obtained using the cuckoo algorithm so that we can increase the accuracy of the data. The results of this study are the result of the remarkable accuracy of the proposed method. So that the proposed method was able to achieve 99% accuracy compared to other existing methods, and it can be said that the proposed method worked well and was able to correctly detect malware to a large extent and improve system performance.

1. Introduction

Malware is generally referred to as software that causes any damage to a personal computer, server, or computer network. Malware is malicious code that is upgraded to harm a computer or computer network. The number of malware is growing rapidly, and this amount of growth has led computer security researchers to invent new ways to protect computers and networks. The rapid growth of malicious code has always been one step ahead of the efforts of security experts to provide solutions to detect and remove them from users' systems. Despite this growth, however, older methods of dealing with malicious code, including waiting for a number of computers to be infected, detecting malicious code, designing a counter, and then presenting it to users and customers, are a long and inefficient process. It is possible for damage by malicious codes. Ways to detect malware and fix it in a timely manner can be used to deal with these malicious threats that cause the system to become infected. Reproduction and dissemination of malicious code may have adverse effects on various types of ordinary users, businesses, and governments using computer systems. For example, if a copy of malicious code infiltrates a computer connected to a network, it could result in loss, unauthorized use, or a large amount of alteration. As a result, users will be unsure of the accuracy of the information on the network. Once logged in, your malware can attack things like sending spam emails, stealing information and hosting account passwords. Malware can use a variety of different methods and techniques to execute itself. For example, some of them use your system as a victim to carry out hacking operations on other systems, some of them collect users' personal information such as bank account number, password and usernames, and may even cause Destroy users in the system. With the prevalence of this type of attack, malicious activity has entered a new phase in which malicious code, instead of infecting computers, seeks to steal users' personal information in order to steal and defraud them. Accordingly, the number and harmful effects are increasing day by day. In order to identify these malicious malware, in this article, a new method is presented in order to populate and then improve the results. The proposed method is that first, after collecting a database containing a number of healthy files and a number of malicious files, to identify malicious malware from healthy software, the backup machine method is used for this segmentation, then using a meta-algorithm.

* Corresponding Author: Reza Molaee Fard
E-mail address: Rezamolae4@gmail.com

Received: 12 December 2020; Revised: 15 February 2021; Accepted: 13 March 2021

Please cite this article as: R. Molaee Fard, Provide A Way To Improve Malware Detection Using Backup Vector Machine And Cuckoo Algorithm, Computational Research Progress in Applied Science & Engineering, CRPASE: Transactions of Electrical, Electronic and Computer Engineering 7 (2021) 1–4, Article ID: 2320.

Cuckoo tries to optimize and improve the results. The results of this study indicate a 99% correct diagnosis of the proposed method.

2. Related Work

Alazab et al, In a 2020 paper Provided a way to identify malware in mobile applications. The researchers proposed an effective classification model of a combination of permission requests and API calls. Because Android apps use a large number of APIs, three different grouping strategies were used to select the most valuable API calls. To maximize the possibility of detecting malicious Android applications. The experimental results obtained from this research using a real malware data set containing a number of Android applications indicate that the proposed method in this research is effective in identifying mobile malware. And can help significantly in malware and program analysis [1].

Alzaylaee et al, In a 2020 article, Presented a way to detect malware on Android operating systems. The researchers proposed a method called DL-Droid. This system is a deep learning system to identify malicious Android applications through dynamic analysis using appropriate input generation. Experiments with more than 3,000 applications (malware and software) on real devices showed improved malware detection results. The results show that the DL-Droid can achieve a maximum of 97.8% detection rate (only with dynamic features) and 99.6% detection rate (with static + dynamic features) respectively, which can perform better than conventional traditional techniques [2].

Darabian et al, In their 2020 paper, Presented a method for detecting malware. In this study, the researchers used sequential pattern extraction techniques to detect the repetitive coding sequence of malicious programs. Maximum recurring coding sequence patterns can be used to distinguish malicious from benign programs, then MFPs are classified as a feature based on KNN, SVM, MLP, and decision tree. The results of this study indicate that 99% of malware is detected [3].

Taheri et al, In their 2020 paper Presented a way to improve the detection of malicious malware in applications. In their paper, the researchers presented four identification methods using the Hamming distance to find similarities between specimens that are first neighbors, all nearest neighbors, nearest neighbors, and nearest neighbors based on the K-medoid. This method detects malicious Android applications. It can also sound an alarm and can prevent the spread of malware. Research shows that the proposed method is 90% accurate [4].

Pektas and Acarman , In their 2020 paper, presented a way to increase the detection accuracy of malware. In their research, the researchers used the API call diagram as a diagram of all possible execution paths that a malware could track at runtime. The embedding of API call diagrams, which has become a set of Mac-sized numerical vector features, is introduced into the deep neural network. The similarity is then taught for each binary function and tested effectively. The similarity is then taught for each binary function and tested effectively. This research also focuses on maximizing network performance by evaluating different embedded algorithms and adjusting various network configuration parameters to ensure the best over-combination of parameters and achieving the highest statistical metric value. The results of the evaluation of this proposed method indicate a 98% accuracy of the proposed method [5].

3. Proposed Method

In the proposed method, a new method is introduced to identify and detect malicious malware among applications. Then we optimize the results to increase the accuracy of the system. The proposed method is that we first provide the database with a database of more than 10,000 applications, including both malicious and healthy applications. Then, using the pattern recognition methods and using the backup vector machine, we try to give a suitable algorithm to this algorithm and then classify the programs from the two directions of malicious malware and correct applications. After identifying and categorizing the patterns using the cuckoo hyperbole algorithm, we try to improve the results obtained from the research and try to increase the accuracy of the system to a considerable extent so that the proposed method can justifiably remove the malware from the software Recognize.

4. Support Vector Machine Classification Method

In the proposed method, we must first find a way to discover and categorize the patterns. To be able to separate malware from applications through this template and category [6]. We use SVM to do this. The backup vector machine works by assuming that we have the data set $\{(x_1, c_1), (x_2, c_2), \dots, (x_n . c_n)\}$And we want to divide them into two classes $c_i = \{-1,1\}$ Each x_i is a p dimension vector of real numbers, which are actually the variables that represent software behavior [7]. Linear classification methods try to separate data by constructing a superficial. The backup vector machine classification method, which is one of the linear classification methods, finds the best surface cloud that separates the data related to the two classes with maximum margin [8] [9] . In order to better understand the content, the following figure shows an image of a data set belonging to two classes, in which the backup vector machine method selects the best super-surface to separate them. In the backup vector machine method, the input vectors are mapped to a multidimensional space [10]. After that, a surface cloud will be created that will separate the input vectors with the maximum possible distance [11] [12]. This super-surface is called the super-surface with the maximum separating boundary. As shown below, two parallel surface clouds will be constructed on either side of the surface with a maximum separating boundary that separate the data for the two classes in such a way that no data is placed at the boundary between these two surface surfaces [13]. A surface cloud with a maximum separating boundary is a surface cloud that maximizes the distance between two parallel surface clouds. It is assumed that the greater the separator boundary or, in fact, the distance between two parallel surface clouds, the smaller the classification error [14] [15].

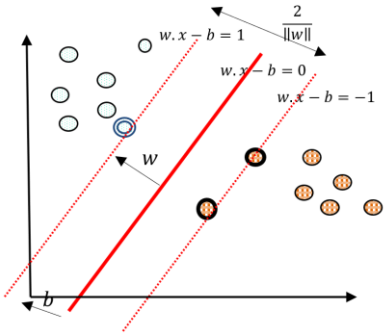

Figure 1. Support Vector Machine

5. Use The Cuckoo Algorithm To Improve Tthe Results

In the next step, we have to optimize the patterns obtained from the previous step to improve the results and increase the accuracy. The proposed optimization method is to use the cuckoo algorithm.

In the next step, we will optimize the node energy using the cuckoo meta-heuristic algorithm. The cuckoo algorithm is one of the newest and most powerful evolutionary optimization methods. This algorithm uses the lifestyle of a bird named Face. This algorithm starts with an initial population. This population of cuckoos has a number of eggs that they will lay in the nest of a number of host birds. Some of these eggs are more similar to the eggs of the host bird, more likely to be the eggs of the host bird, more likely to grow into an adult cuckoo. Other eggs identified by the host bird are destroyed. The amount of eggs grown indicates the suitability of the nests in that area. The more eggs that can live and survive in an area, the more profit will be made to that area; Therefore, the situation in which the largest number of eggs are saved will be a parameter that the cuckoos intend to optimize [16] [17] [18].

- To solve an optimization problem, it is necessary to form the values of the problem variables in the form of an array.

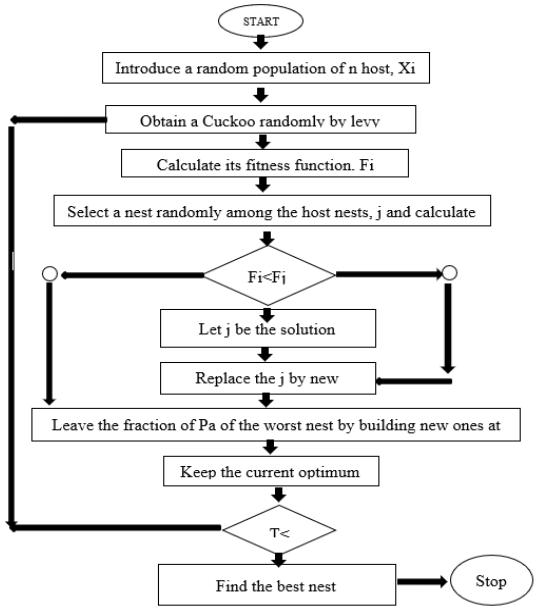

Figure 2. Shows The Framework Of The Cuckoo Algorithm.

- In the cuckoo algorithm, the array is called Habitat.
- In the next N_{var} optimization problem, a Habitat will be an array var $1 \times_{var}$ that represents the current living position of the cuckoos. This array is defined as follows.

$$\boldsymbol{abitat} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, .\,, \dots, \boldsymbol{x_{Nvar}}] \quad (1)$$

- The appropriateness of the current Habitat is obtained by evaluating (f_p) in the Habitat, therefore:

$$\boldsymbol{Profit} = \boldsymbol{f_p}\ \boldsymbol{habitat} = \boldsymbol{f_p}(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, .\,, \dots, \boldsymbol{x_{Nvar}}) \quad (2)$$

- As can be seen, the cuckoo algorithm is an algorithm that maximizes the profit function.
- To use the cuckoo algorithm to solve the minimization problems, it is enough to multiply a negative sign in the cost function.
- To start the optimization algorithm, we generate a Habitat matrix with size $N_{pop} \times N_{var}$.
- A number of random eggs are then assigned to each of these habitats.
- In nature, each cuckoo lays between 5 and 20 eggs. These numbers are used as the upper and lower limits of each cuckoo egg allocation in different iterations.

Another real cuckoo habit is to lay their eggs in a certain range.

$$\boldsymbol{ELR} = \boldsymbol{a} \times \frac{\boldsymbol{Number\ of\ current\ cuckoos\ eggs}}{\boldsymbol{Total\ number\ of\ eggs}} \times (\boldsymbol{Var_{hi}} - \boldsymbol{Var_{low}}) \quad (3)$$

6. Evaluate The Proposed Method

The most important factor in evaluating malware detection methods is the use of metrics. To evaluate the accuracy of the proposed algorithm, a comparison was made between the SVM algorithm and other methods such as RBF, MLP and KNN, which the proposed method was able to obtain more accuracy than other algorithms. Also, an evaluation was performed between the cuckoo meta-algorithm and other existing algorithms such as the gray wolf algorithm, the firefly algorithm and the PSO algorithm, in which the proposed algorithm could be more accurate.

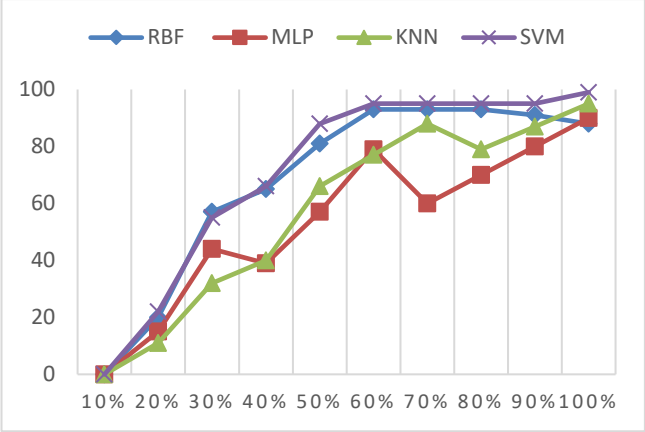

Figure 3. Diagram comparing the accuracy of the Proposed method with other methods

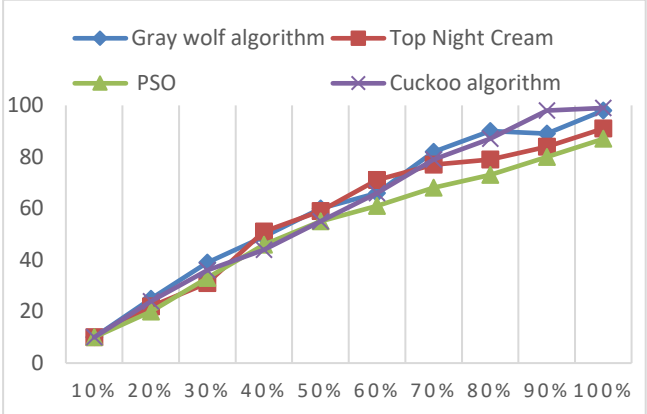

Figure 4. Diagram comparing the accuracy of the Proposed method with other methods

7. Conclusion

Malware is undoubtedly one of the most important security threats to information technology and will continue to be so. Over the years, from simple malware to more advanced threats such as advanced viruses such as today's advanced viruses have always been one of the most important reasons for security incidents. Malware is code snippets written by programmers to infect the system without the owner's permission and to perform unwanted or malicious actions. To prevent this malicious malware, a way must be taken to deal with these threats. One way to do this is to use pattern detection algorithms to detect these malware from applications. In this research, pattern detection method and backup vector machine classification are used to identify malicious malware. Then we tried to improve the results using the cuckoo meta-heuristic algorithm. Evaluations of the proposed method showed a very high accuracy compared to other methods. The main criterion for comparing algorithms is the use of accuracy algorithm. In the obtained results and studies, the proposed method was able to achieve 99% accuracy and it can be said that this algorithm could achieve significant performance from other methods for detecting malware.

References

[1] M. Alazab, M. Alazab, A Shalaginov, A Mesleh, Intelligent mobile malware detection using permission requests and API calls, Future Generation Computer Systems 107 (2020) 509–521.

[2] MK.Alzaylaee, SY.Yerima, S. Sezer, DL-Droid: Deep learning based android malware detection using real devices, Computers & Security 89 (2020) 101663.

[3] H. Darabian, A. Dehghantanha, and K .wang R. Choo, An opcode-based technique for polymorphic Internet of Things malware detection, Concurrency and Computation: Practice and Experience 32 (2020) e5173.

[4] R .Taheri, M. Ghahramani, R. Javidan, M. Shojafar, Similarity-based Android malware detection using Hamming distance of static binary features, Future Generation Computer Systems 105 (2020) 230–247.

[5] A. Pektaş, T. Acarman, Deep learning for effective Android malware detection using API call graph embeddings, Soft Computing 24 (2020) 1027–1043.

[6] K. Shankar, SK. Lakshmanaprabu, D. Gupta. De Albuquerque. Optimal feature-based multi-kernel SVM approach for thyroid disease classification, The journal of supercomputing 76 (2020) 1128–1143.

[7] J. Sun, H. Li, H Fujita, B. Fu, W. Ai., Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting, Information Fusion 54 (2020) 128–144.

.[8] R. Hu, X. Zhu, Y. Zhu, J. Gan, Robust SVM with adaptive graph learning, World Wide Web 23 (2020) 1945–1968.

[9] AP. Gopi, R.NS Jyothi, VL Narayana, Classification of tweets data based on polarity using improved RBF kernel of SVM, International Journal of Information Technology (2020) 1–16.

[10] Y. Zhang, H. Yang, H Cui, Q. Chen, Comparison of the ability of ARIMA, WNN and SVM models for drought forecasting in the Sanjiang Plain, China, Natural Resources Research 29 (2020) 1447–1464.

[11] W. Fu, K. Shao, J. Tan, K. Wang, Fault diagnosis for rolling bearings based on composite multiscale fine-sorted dispersion entropy and SVM with hybrid mutation SCA-HHO algorithm optimization, IEEE Access 8 (2020) 13086–13104.

[12] M. Azimi-Pour, H. Eskandari-Naddaf, Linear and non-linear SVM prediction for fresh properties and compressive strength of high volume fly ash self-compacting concrete, Construction and Building Materials 230 (2020) 117021.

[13] G. Wu, R. Zheng, Y. Tian, D. Li. Joint Ranking SVM and Binary Relevance with robust Low-rank learning for multi-label classification, Neural Networks 122 (2020) 24–39.

[14] CL. Chowdhary, M. Mittal, PA. Pattanaik, Z. Marszalek, An efficient segmentation and classification system in medical images using intuitionist possibilistic fuzzy C-mean clustering and fuzzy SVM algorithm, Sensors 20 (2020) 3903.

[15] MA. Ganaie, M. Tanveer, PN. Suganthan Oblique decision tree ensemble via twin bounded SVM, Expert Systems with Applications 143 (2020) 113072.

[16] HR. Boveiri, An enhanced cuckoo optimization algorithm for task graph scheduling in cluster-computing systems." Soft Computing 24 (2020) 10075–10093.

[17] X. Cai, Y. Niu, S. Geng, J. Zhang, Z. Cui, An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search, Concurrency and Computation: Practice and Experience 32 (2020) e5478.

[17] M. İnci, A. Caliskan, Performance enhancement of energy extraction capability for fuel cell implementations with improved Cuckoo search algorithm, International Journal of Hydrogen Energy 45 19 (2020) 11309–11320.