International Journal of Safety and Security Engineering
Vol. 12, No. 1, February, 2022, pp. 39-45
Journal homepage: http://iieta.org/journals/ijsse

Machine Learning-Based Malware Software Detection Based on Adaptive Gradient Support Vector Regression

Lavanya Bharathi*, Shanthi Chandrabose

Department of Computer Science, Vels Institute of Science Technology and Advanced Studies, Pallavaram, Chennai 600117, Tamil Nadu, India

Corresponding Author Email: lavanya84tvm@gmail.com

https://doi.org/10.18280/ijsse.120105

Received: 30 November 2021
Accepted: 15 January 2022

Keywords:
malware software detection, Adaptive Gradient Support Vector Regression (AGSVR), Adaptive Normalized Data Analysis (ANDA), Adaptive Static Feature Analysis (ASFA), machine learning

ABSTRACT

Malware Software detection is one of the key steps in developing the anti-malware software in computer systems. In the existing system, malware detection had been performed inefficiently with poor detection accuracy. The previous methods were not efficient enough to detect malware in terms of low efficiency, low overhead, and poor security. The proposed method uses the Machine learning approaches for Malware software detection based on the Adaptive Gradient Support Vector Regression (AGSVR) to overcome these issues. Initially, the pre-processing stage reduces the imbalanced data and missing values based on the Adaptive Normalized Data Analysis (ANDA) using the specified dataset. Secondly, features extracted from the pre-processing stage are used for the training and testing of dataset using the Adaptive Static Feature Analysis (ASFA) algorithm. Each selected feature value is extracted and stored with the associated category of specified dataset. Absolute rights are established based on the values assigned to the Malware software detection system. Finally, the analysis of the selected features is done using the classification based on the training and testing of malware data. The classification is based on the Adaptive Gradient Support Vector Regression (AGSVR) algorithm. Recognition is an approach to mutual identification that is useful for distinguishing between malicious and non-malicious applications. Then, the extracted information is used to classify malicious and benign applications that use machine learning-based AGSVR classification algorithm. The simulation results show the improved sensitivity and specificity, reduced error rate, high accuracy and reduced time complexity in the proposed method which is better than the previous method.

1. INTRODUCTION

Malware is a virus program that is create4d in order to infect the computer system. Malware detection is one of the biggest challenges to protect the important data in our computer system. Malicious software show millions of samples and one of the non-profit claims is about 1.5B of malware models. The largest sample collection to avoid detection, and other technologies is done mainly through falling hashes of polymorphic malware variants, mutations, semen, malware examples, and common signatures used by anti-virus detection. The scale of the malware set is unlikely to slow 1M / day and then expands at about 400K / day. As a result, industry and academia are interested in doing malware analysis [1]. Since the variant detected is an increasingly serious threat data set, the research community has put a considerable amount to understand the system.

This malware application is designed to allow the reuse of malware detection code. Without direct pairwise comparison, the proposed method can calculate the similarity between large malware sample data set. The signature of the blocks that has been extracted from a malware sample is used to create variant malware signatures which allow the creation of derivative data sets from other signatures. Therefore, the source of malware is also due to the reuse of detection method and a conventional mutation detection tool of block-level code. The experiment results in malware while maintaining the ability to explore the reuse approach show that have the advantage in comparison time and the accuracy of detection of malware variants.

A malicious software program developed by a computer system is infected by cyber-attacks that exploit security vulnerabilities. These extraordinary financial and political motives reward the owner of the malware as they have many network computers, so they can compromise by consuming more energy to achieve their malicious goals. Malware is deployed on a computer infected by cyber-attacks, and it is a malicious software program. This software and malware are political to obtain a financial reward that has been created at an alarming rate. This malware, infected with the entire network's security, will be sent to obtain sensitive information. Systems that are affected by this software and malware are called a bot [2-5]. Malware programmers write a BOT or proxy program to use as a technology virus in the various regions on the Internet and install the automaton in the infected computer system.

Detecting the signature malware in any system is the unique challenges to grant permissions and limited users with limited resources [6-10]. It provides a unique opportunity to connect to the metadata required for each application. The work aims to detect malicious applications based on machine learning techniques using real world android applications. Our system will extract relevant application features with the help of well-known feature selection techniques. We consider different categories of android application in the system and investigate the API calls and permission mode of application.

Deployment of applications on mobile devices is part of the overall malware detection and prevention process. Is used to detect the installation of the application and to uninstall restrictions and malicious application of the interruption of other applications that include the Android application. The main contribution is as follows. 1) Using a minimum of system resources that does not require the application to run the active detection of malicious application of lightweight deployment. Since there is no escalation, it has the authority 2) to prevent malicious software. A system has been used to confirm the uninstallation of the root privileges of users and applications; however, the deployed application detects default permissions. 3) Reaction of malware protection function can respond to install and update broadcasting system. It is not required to scan all of the downloaded items; it has a chance to run; regardless of any other authority that shall be obtained before it is installed, it still will be able to remove the malware.

Proposed Adaptive Gradient Support Vector Regression (AGSVR) can deploy the actual on a device of the plurality of malware detector, real-time malware detection framework. Adaptive Gradient Support Vector Regression (AGSVR) is designed as part of the operating system as it has two modules for monitoring and testing. A plurality of sub-detectors can be installed to improve the safety of the system. Sub detector monitors the use of the analysis in a monitoring module for detecting run-time information for malware. They report their analysis results to the Adaptive Gradient Support Vector Regression (AGSVR) detector. The detector is decided at the time to mark the application as malware. A framework like Adaptive Gradient Support Vector Regression (AGSVR) can help third parties through the application market, and users publish their detection technology.

The rest of paper is organized as follows: section 2 describes the pros and cons of the existing solutions available for malware detection. Section 3 describes about the materials and methods used in the work. Section 4 discusses the results of the proposed work by comparing it other state-of-the-art methods. Section 5 conclude the work.

1.1 Problem statement

Malwares are created by the adversaries in order to harm the computer system. This malware is the biggest challenges for many organizations to safeguard their computing resources. Due to this, many solutions have been emerged to detect and eliminate malware from the system. Despite many security frameworks available for removing malwares, still it is highly challenging to cyber space. Many researchers have provided various mechanisms for malware detection and many anti-malware software are available. But still malware attacks happen in a large number. Most of the research available for malware detection are based on machine learning strategies. Some of the common machine learning algorithms used in the existing works are decision tree, Convolutional Neural Network, SVM, regression etc., However, all the algorithms only focused on basic security features. There are no broad ways for the efficient detection of malware. In this work, we apply AFSVR algorithm to classify the malicious and benign patterns. First, we perform feature selection using ANDA, then the relevant features are extracted using ASFA.

2. RELATED WORKS

Malware is becoming a serious threat to more and more embedded systems, such traditional software solutions such as anti-virus and patch program is still evolving, has been very successful in the defense advanced malicious program. Li et al. [11] proposed a real time protection for malware detection. The work recognizes the signatures of malware using past behaviors and samples with the help machine learning approaches. Authors compare the proposed work with other works in terms of accuracy, detection rate, and energy. The work claims the proposed system is better than other works. However, the work done for feature extraction will increase the computational overhead. Kim et al. [12], proposed a security framework to detect malware in android applications. The work uses multi model deep learning techniques for malware detection. The work also uses multiple features to classify the samples and only relevant feature are extracted. The performance evaluation of the work is done by comparing the detection accuracy of proposed work with existing models. The work improved the detection accuracy in detecting the malware programs. However, the framework does not consider dynamic features for classification process. Hence, the performance of the detection in static features are questionable. Ma et al. [13], proposed a detection system for malware in android applications. The authors constructed a control flow graph using FlOWDROID to perform taint analysis. Then the work extracts the information with two stages training stage and detection stage. Standard metrics were considered to evaluate the performance of the work. Authors claims that the detection accuracy is improved with 98.5% classification accuracy than the other available work. However, the work considers only two classifications to detect the malware in applications which cannot guarantee the improvement in the proposed work. Li et al. [14] performs malware detection based on domain generation algorithm involving machine learning techniques. There are two phases in the algorithm classification phase and clustering phase. The work can effectively cluster the malicious domain from the available domains and performs efficient detection. However, it is very difficult to analyze the performance of the work in variable length domain.

Li et al. [15] proposed a classification method for malware detection based on factorization machine. The work evaluates the performance based on DREBIN and AMD datasets. The work achieves 100% precision score and 99% precision score on AMD datasets.

Machine learning-based solutions have been used for the automatic detection of malware on the successful Android mobile phone [16]. However, the machine learning model, a powerful hostile embodiment made by carefully perturbing the selection to the normal input, is missing. Gong et al. [17], discuss about robust detection methods to identify malware attacks. As part of this work, authors perform machine learning based detection to identify the malware program. The authors analyzed the challenges in ML based detection schemes with collaborative approach in real time environment. The work concludes that the machine learning approach is more effective than any other approaches in performing efficient classification.

From the above analysis, we identified the performance of static detection model and dynamic detection mode. We understand that the static detection model performs well in malware detection process. However, it cannot cope up with the dynamic nature of classification approach using machine learning strategies. Dynamic approach for malware detection can effectively detect malware at runtime but it cannot be run in smart phone directly. Distribution characteristics of malware are mostly observed in previous works of detection of detecting Android malware machines to the assumption that no change over time. The work also identified that the chances of vulnerability are found in permission model of android applications. In this work, we perform efficient feature extraction based on AGSVR and ANDA.M Then, training and testing is done for the extracted features using ASFA. The work then uses the features in machine learning algorithms to perform the detection process efficiently.

3. MATERIALS AND METHOD

With the rapid advancement of data technology, the rapid growth of malicious code has become one of the biggest cyber security threats. The efficiency of a malware detection system depends on the characteristics of the malware to effectively differentiate how analytics technologies extract it. There are different ways to set up your analytical environment with different static and dynamic tools. The proposed Adaptive Gradient Support Vector Regression (AGSVR) method is used to categorize the malware software detection machine learning methods to low complexity and high speed. Malware detection techniques are provided using machine learning algorithms.

There are various challenges in developing malware classifiers. Finally, effective malware detection systems are to be created by dealing with software issues of malware detection.

Figure 1 shows the Malicious software detection classified into the proposed method using the Adaptive gradient support vector Regression (AGSVR) algorithm. In the pre-processing stage, the imbalanced data is handled using the Adaptive Normalized Data Analysis (ANDA)and in the second stage, the features are extracted using the Adaptive Static Feature Analysis (ASFA)and classified as Malicious or Non malicious.

3.1 Preprocessing using Adaptive Normalized Data Analysis (ANDA)

Feature selection is the important task in the preprocessing stage. Only relevant and suitable features need to be considered to detect malware programs efficiently [18-20]. There are several factors to consider when creating a login feature package from an Android app. Using machine learning algorithms to detect malware for Android, a good feature set of training is necessary for machine learning algorithms. It includes data pre-processing, data product pre-processing, cleaning, normalization, modification data etc. Its selection is the final training set. A single array of data would be fine if each set of pre-processing algorithms has better performance, but this would not happen. Therefore, the Adaptive Normalized Data Analysis (ANDA) algorithm data set is presented for each step because pre-processing facilitates the reach of the best performance data set. The work uses dataset from https://ocslab.hksecurity.net/andro-autopsy. There are 2 lakhs android applications collected from the dataset. We then classify the dataset in to malware applications and benign applications. The work considers different categories of android application in the system and investigate the API calls and permission mode of application.

The input database contains a call array of worms, viruses, and Trojan horses. At this point, the original file is executed after extracting an array of attributes from the executable. Attributes that there are datasets to extract and then extract each data after loading the dataset. Sample processing is also done to categorize worms, viruses, Trojan horses, or data types as usual. We apply the feature selection in the dataset using ANDA and the performance of the proposed algorithm was examined based on confusion matrix. The procedure for the proposed method in preprocessing is given in algorithm 1.

```
Algorithm 1: Adaptive Normalized Data Analysis (ANDA)
Input: Initialize the data
Output: Reduced Missing values
Step 1: ID- Input the dataset
Step 2: ED- Extract the dataset
For a= 0: ID do
        ED ← Separate each data;
End for;
Step 3: RPCA → using for selecting and computing the threshold values.
Step 4: Normalization scales features
```

$$\text{Normalization } X^2 \frac{x-\min(x)}{\max(x)-\min(x)} \tag{1}$$

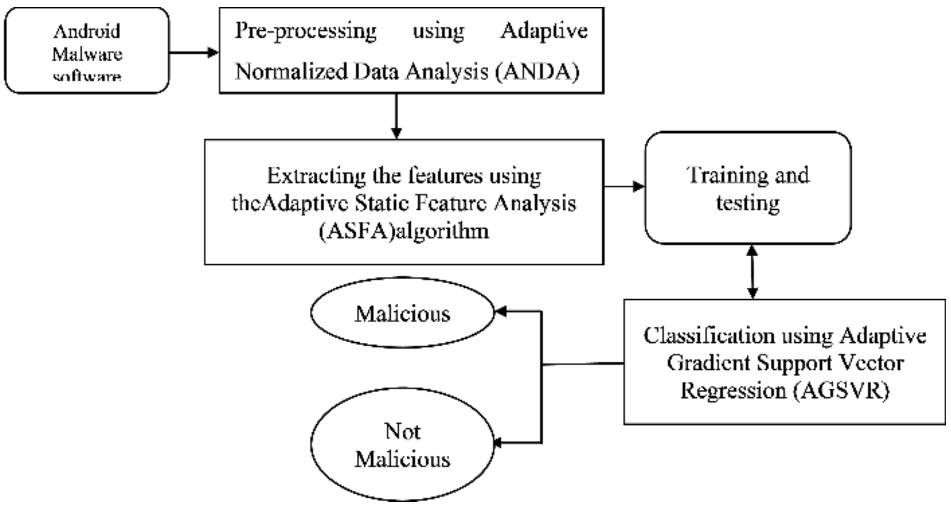
Android Malware software
Pre-processing using Adaptive Normalized Data Analysis (ANDA)
Extracting the features using theAdaptive Static Feature Analysis (ASFA)algorithm
Training and testing
Malicious
Classification using Adaptive Gradient Support Vector Regression (AGSVR)
Not Malicious

Figure 1. Proposed architecture for malicious software detection

```
Step 4: CD ← Categorizing of data
    For a = 0; ED do
    If ED.a == Worm
        CD = Worm;
    Else if ED. a = = Trojan
        CD = Trojan;
    Else if ED. a = = Virus
        CD = Virus;
    Else
        CD = Normal;
    End if;
End for;
```

where, ANDA-Adaptive Normalized Data Analysis, in the pre-processing algorithm, uses the malware dataset to handle the missing values and imbalanced data. Training and testing data facilitate computing and selecting the instance for the dataset attribute values.

3.2 Extracting the features for Adaptive Static Feature Analysis (ASFA) algorithm

Extracting the features of information such as effective data, filters, and process names is very important as they are the most important binary files. The information can be used to detect malware software systems. First, it is necessary to explain the relationship between data and feature selection, which begins to function due to avoidable confusion. The process of machine learning covers pre-processing rather than choosing the data feature. Facility selection is a component of data pre-processing used to improve machine learning skills by removing inappropriate or redundant features using the Adaptive Static Feature Analysis (ASFA) algorithm. Feature Extraction aims to reduce the number of features in a database by creating new features (and not excluding original features). Features from these new lower boxes can summarize the information contained in the original package of features. In this way, a version of these original features can be created for both the original data. Steps for ASFA feature extraction is given in algorithm 2.

Algorithm 2: Feature Extraction using Adaptive Static Feature Analysis (ASFA)

```
Input: A = {Ax}, the set of all Dataset (n)
Output: B = {Bx,y}, set of feature in the dataset, the row vector features of the original data.
Step 1: Begin
Step 2: Using ALCA → for extracting the features in the original data set
        For x= 1 to n do
    Decompile the Ax
    Generate the extracting feature f (151-bit) from Manifest
    Generate the API features a (3262-bit) from sources;
        End for
    For y = 1 to 151 do
    B x, y = a y;
    End for
        For y = 152 to 3413 do
    B x, y = f y;
    End for
Step 3: πa is usually frequencies of the training set:
```

$$\Pi_a = \frac{No.of.sampleclassA}{Totalofsamples} \tag{2}$$

```
Step 4: End procedure
```

where, ALCA- Adaptive Linear Component Analysis, API-Application programing interface, n- Number of datasets, Ax – Input values, B x,y – features of row and column values is used for the dataset analysis of the original information to extract the features based on this algorithm. It helps in removing the inappropriate values and selecting the processing data in the Malware software dataset. The API ensures that this process is an application that uses the A path, so it can combine API with permissions to create more representative and comprehensive feature extraction that reflects the behavior of 3413 bits (151 extracting feature functions + 3262 API functions), which is much less complex and each malware The detection system can also be developed for each application.

3.3 Classification using Adaptive Gradient Support Vector Regression (AGSVR)

To theft sensitive information or unauthorized access to private networks, different malwares are created for different purposes. Malware software that attempts to perform malicious activities on the computer is a software program. It will detect any malware and notify of the action used by intrusion detection systems to prevent them. The Soft Computing Technology section detects the provided datasets that allow packets or files to come through the network. Here, the details are focused using Adaptive Gradient Support Vector Regression (AGSVR) with a high accuracy detection system in various forms such as Malware software system. The steps for AGSVR is given in algorithm 3. Most applications call their operating system various APIs, known to require the Android operating system. Once we compiled a small file into an Android application, we further scrutinized every small file included in our list, thus using SmsManager. send text message malicious API calls (or Wifi-Manager setWifi-Enabled) for each such application.

Algorithm 3: Adaptive Gradient Support Vector Regression (AGSVR)

```
Input: Malware software training and testing data
Output: Determine the calculated accuracy
Step 1: Handle SRC= Nt Open File;
Step 2: Handle section Handle + NtCreatesection(Section Handle);
If (QueryAttributesFile>TF)
While (Stopping condition is not met) do
    Implement AGSVR training step for each data values
```

$$\text{AGSVR} \rightarrow \text{Calculating weights } TF_{xy} = W_{x,y}/T_{x,y} \tag{3}$$

```
Implement AGSVR to classify the testing data values
```

$$TF_i = \log\frac{N}{Ni} + 1 \tag{4}$$

```
End While
End If
Return Accuracy
Step 3: End
```

where, AGSVR-Adaptive Gradient Support Vector Regression, W-weight values, TF-Time Frequency as in Eqns. (3) and (4)., analyzes the classification accuracy based on the malware software detection to the implementation of training and testing data analysis of the optimal values.

4. DISCUSSION

The proposed implementation results and performances have been tested using selected dataset and the selected features are trained using machine learning algorithm to identify the accuracy of Malware detection using the proposed method. Test case measurements are calculated by the true and false positions of the error rate performed during processing. The test results have been compared to the Adaptive Gradient Support Vector Regression (AGSVR) Method. The analysis is done based on Sensitivity, specificity, accuracy, Error Rate and Time complexity in the proposed system.

We consider F-measure and accuracy to measure the performance of the proposed algorithm with other state of the algorithm such as:

Accuracy: Malware detection is done with the help of benign applications and malware applications. For all prediction steps, the proportion of correct predication is its accuracy. Accuracy is given by

$$Accuracy = \frac{X+Y}{\text{M classes}} \tag{5}$$

F-measure: The work also uses F-measure to evaluate precision and recall. These two methods can be used to compare the proposed model with other existing models. F-measure is given by

$$F - measure = \frac{2 * P + R}{P + R} \tag{6}$$

where, P and R are precision and recall respectively.

4.1 Experimental setup

To identify the performance of our proposed work, AGSVR is implement on different categories of android application extracted from the dataset. Then, the work identified the suitable detection model and compare the proposed model with other models such as GCRNC, KNN, Naïve Bayesian, and LCS. In the first part of experiment, feature selection is performed on different features to select relevant features for malware detection. Next, selected features are trained using machine learning approaches to analyze the performance with accuracy and F-measure. Table 1 shows the parameters used in the proposed model.

Table 1. Simulation parameters for the proposed system

Parameters used	Values processed
Input dataset	Malware software Dataset
simulation language	Python
Simulation tool	Anaconda
Number of data	1000
Trained data	700
Testing data	300

F-measure and Accuracy are used in evaluating the proposed method with other models such as LCS, GCRNC, Naïve Bayesian and KNN. The outcome of the experiment shows that the proposed method provides improved results compared to the previous method results.

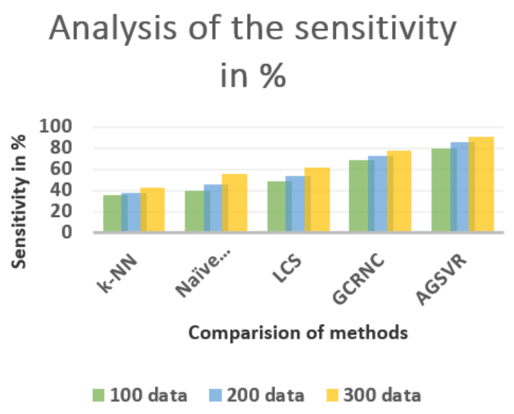

Figure 2. Analysis of the sensitivity

Figure 2 describes the Sensitivity performance of the proposed and existing methods the proposed Adaptive Gradient Support Vector Regression (AGSVR) improves the sensitivity up to 90.1%, which is better than the previous method of Gradient Conventional Recursive Neural Classifier (GCRNC).

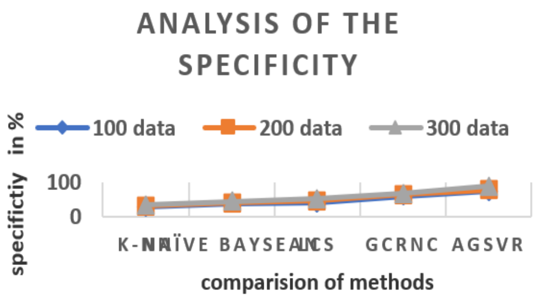

Figure 3. Analysis of the specificity

Figure 3 describes the Specificity performance of the proposed and existing methods; the proposed Adaptive Gradient Support Vector Regression (AGSVR) improves the specificity up to 88.2%, which is better than the previous method of Gradient Conventional Recursive Neural Classifier (GCRNC).

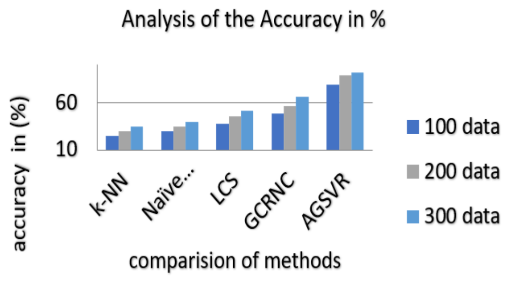

Figure 4. Analysis of the accuracy

Figure 4 describes the Accuracy performance of the proposed and existing methods. The proposed Adaptive Gradient Support Vector Regression (AGSVR) improves the accuracy up to 92.2%, which is better than the previous methodGradient Conventional Recursive Neural Classifier (GCRNC).

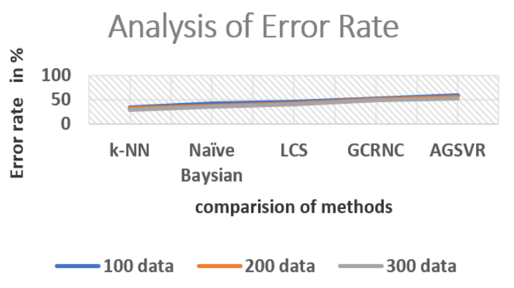

Figure 5. Analysis of the error rate

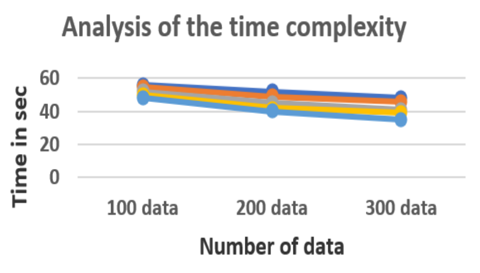

Figure 6. Analysis of the time complexity

Figure 5 describes the Error Rate performance of the proposed and existing methods. The proposed Adaptive Gradient Support Vector Regression (AGSVR) reduces the Error Rate score up to 52.4%, which is better than the previous method of Gradient Conventional Recursive Neural Classifier (GCRNC).

Figure 6 describes the performance time complexity of the proposed and existing methods. The proposed Adaptive Gradient Support Vector Regression (SVR) method reduces the time complexity level to 35sec which is better than the previous method of Gradient Conventional Recursive Neural Classifier (GCRNC).

5. CONCLUSION

Despite many proposed solutions, some challenges have still not been addressed, especially because of the rapidly evolving nature of malware. For complexity in emerging-malware issues with source code, code obfuscation-related difficulties are available and immediately they require special attention. In the previous methods, the number of samples, especially malware samples, is insufficient and the malware properties obtained are not yet represented. The Adaptive Gradient Support Vector Regression (AGSVR) method is applied to perform malware detection. The performance analysis is done by comparing the proposed model with state-of-the-art models. From the analysis, we observed that the reduced computational consumption ensures the enhancement of the proposed algorithm, which is higher in other models and traditional machine learning methods. In the simulation results, the proposed method AGSVR provides better accuracy compared to the previous methods. The proposed AGSVR improves the sensitivity to 90.1%, specificity to 88.2%, accuracy to 90.1%, reduces the error rate to 52.1% and reduces the complexity of time in 35Sec using the Malware software detection dataset.

REFERENCES

[1] Das, S., Liu, Y., Zhang, W., Chandramohan, M. (2015). Semantics-based online malware detection: Towards efficient real-time protection against malware. IEEE Transactions on Information Forensics and Security, 11(2): 289-302. https://doi.org/10.1109/TIFS.2015.2491300

[2] Roseline, S.A., Geetha, S., Kadry, S., Nam, Y. (2020). Intelligent vision-based malware detection and classification using deep random forest paradigm. IEEE Access, 8: 206303-206324. https://doi.org/10.1109/ACCESS.2020.3036491

[3] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Venkatraman, S. (2019). Robust intelligent malware detection using deep learning. IEEE Access, 7: 46717-46738. https://doi.org/10.1109/ACCESS.2019.2906934

[4] Dai, Y., Li, H., Qian, Y., Yang, R., Zheng, M. (2019). Smash: a malware detection method based on multi-feature ensemble learning. IEEE Access, 7: 112588-112597. https://doi.org/10.1109/ACCESS.2019.2934012

[5] Gu, J., Sun, B., Du, X., Wang, J., Zhuang, Y., Wang, Z. (2018). Consortium blockchain-based malware detection in mobile devices. IEEE Access, 6: 12118-12128. https://doi.org/10.1109/ACCESS.2018.2805783

[6] Euh, S., Lee, H., Kim, D., Hwang, D. (2020). Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems. IEEE Access, 8: 76796-76808. https://doi.org/10.1109/ACCESS.2020.2986014

[7] Pan, Y., Ge, X., Fang, C., Fan, Y. (2020). A systematic literature review of android malware detection using static analysis. IEEE Access, 8: 116363-116379. https://doi.org/10.1109/ACCESS.2020.3002842

[8] Caviglione, L., Choraś, M., Corona, I., Janicki, A., Mazurczyk, W., Pawlicki, M., Wasielewska, K. (2020). Tight arms race: Overview of current malware threats and trends in their detection. IEEE Access, 9: 5371-5396. https://doi.org/10.1109/ACCESS.2020.3048319

[9] Fang, Z., Wang, J., Geng, J., Kan, X. (2019). Feature selection for malware detection based on reinforcement learning. IEEE Access, 7: 176177-176187. https://doi.org/10.1109/ACCESS.2019.2957429

[10] Martins, N., Cruz, J.M., Cruz, T., Abreu, P.H. (2020). Adversarial machine learning applied to intrusion and malware scenarios: A systematic review. IEEE Access, 8: 35403-35419. https://doi.org/10.1109/ACCESS.2020.2974752

[11] Li, J., Sun, L., Yan, Q., Li, Z., Srisa-An, W., Ye, H. (2018). Significant permission identification for machine-learning-based android malware detection. IEEE Transactions on Industrial Informatics, 14(7): 3216-3225. https://doi.org/10.1109/TII.2017.2789219

[12] Kim, T., Kang, B., Rho, M., Sezer, S., Im, E.G. (2018). A multimodal deep learning method for android malware detection using various features. IEEE Transactions on Information Forensics and Security, 14(3): 773-788. https://doi.org/10.1109/TIFS.2018.2866319

[13] Ma, Z., Ge, H., Liu, Y., Zhao, M., Ma, J. (2019). A combination method for android malware detection based on control flow graphs and machine learning algorithms. IEEE Access, 7: 21235-21245. https://doi.org/10.1109/ACCESS.2019.2896003

[14] Li, Y., Xiong, K., Chin, T., Hu, C. (2019). A machine learning framework for domain generation algorithm-

based malware detection. IEEE Access, 7: 32765-32782. https://doi.org/10.1109/ACCESS.2019.2891588

[15] Li, C., Mills, K., Niu, D., Zhu, R., Zhang, H., Kinawi, H. (2019). Android malware detection based on factorization machine. IEEE Access, 7: 184008-184019. https://doi.org/10.1109/ACCESS.2019.2958927

[16] Chen, X., Li, C., Wang, D., et al. (2019). Android HIV: A study of repackaging malware for evading machine-learning detection. IEEE Transactions on Information Forensics and Security, 15: 987-1001. https://doi.org/10.1109/TIFS.2019.2932228

[17] Gong, L., Lin, H., Li, Z., Qian, F., Li, Y., Ma, X., Liu, Y. (2020). Systematically landing machine learning onto market-scale mobile malware detection. IEEE Transactions on Parallel and Distributed Systems, 32(7): 1615-1628. https://doi.org/10.1109/TPDS.2020.3046092

[18] Zhang, H., Luo, S., Zhang, Y., Pan, L. (2019). An efficient Android malware detection system based on method-level behavioral semantic analysis. IEEE Access, 7: 69246-69256. https://doi.org/10.1109/ACCESS.2019.2919796

[19] Narayanan, A., Chandramohan, M., Chen, L., Liu, Y. (2017). Context-aware, adaptive, and scalable android malware detection through online learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(3): 157-175. https://doi.org/10.1109/TETCI.2017.2699220

[20] Aslan, Ö., Yilmaz, A.A. (2021). A new malware classification framework based on deep learning algorithms. IEEE Access, 9: 87936-87951. https://doi.org/10.1109/ACCESS.2021.3089586