Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Metaheuristic Algorithms for Optimization and Feature Selection in Cloud Data Classification Using Convolutional Neural Network

Nandita Goyal *

*Corresponding Author, M. Tech., Assistant Professor, Department of Information Technology, Ajay Kumar Garg Engineering College, Ghaziabad, India. E-mail: nanditagoyal@gmail.com

Munesh Chandra Trivedi

Associate Professor, Department of Computer Science and Engineering National Institute of Technology, Agartala, India. E-mail: munesh.trivedi@gmail.com

Abstract

Cloud Computing has drastically simplified the management of IT resources by introducing the concept of resource pooling. It has led to a tremendous improvement in infrastructure planning. The major goals of cloud computing include maximization of computing resources with minimization of cost. But the truth is that everything has a price and cloud computing is no different. With Cloud computing, several security concerns need to be addressed. Cloud forensics plays a vital role in addressing the security issues related to cloud computing by identifying, collecting, and studying digital evidence in the cloud environment.

This study aims to explore the concept of cloud forensics by applying optimization for feature selection before the classification of data on the cloud side. The data is classified as malicious and non-malicious using a convolutional neural network. The proposed system makes a comparison of models with and without feature selection algorithms before applying the data to CNN. A comparison of different metaheuristics algorithms-Particle Swarm Optimization, Shuffled Frog Leap Optimization, and Firefly algorithm for feature optimization are done based on convergence rate and efficiency.

Keywords: Feature selection, Classification, Cloud computing, Metaheuristic Algorithm, Convolution Neural network

Journal of Information Technology Management, 2023, Vol. 15, Issue 3, pp. 99-112
Published by the University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2023.93627
Article Type: Research Paper

Received: April 03, 2023
Received in revised form: June 13, 2023
Accepted: July 20, 2023
Published online: August 26, 2023

Introduction

These days Cloud computing is one of the most widely known and used computing paradigms used by a huge number of users all around the world. Cloud computing is one of the rising wireless storage technologies, which may suffer from attacks like other technologies. An effort has been made in the field of development of computing technologies by scholars and companies. There has been a vast development in the fields of cloud computing, edge computing, fog computing, mobile computing, and the Internet of Things (IoT) for the execution of several tasks simultaneously with low cost and time. Cloud computing offers many benefits to users in terms of services like on-demand services, elasticity pay-per-use, etc. But at the same time, it is also accompanied by several security threats or issues (Mishra et al., 2021). For years, researchers have been working around cloud computing open security issues. The major challenge has always been to understand the aspects that are important to designing a secure system in a cloud environment. Though the cloud is efficient and elastic by nature in offering services, the host system in the cloud is highly prone to intrusions. Therefore, the detection of Data intrusion that is stored in the cloud is an important aspect that requires the attention of researchers and industry (Dutta et al., 2018; Singh et al., 2018).

This paper focuses on the use of feature selection algorithms before the classification of cloud forensic data more specifically cloud hosts as malicious and non-malicious. The host system is the major component in the cloud that stores all the information gathered from the users into virtual servers as well as performs the execution tasks for the user applications. Therefore, the host system is highly prone to intrusions (Martini & Choo, 2012). For classification, we use a Convolutional Neural Network. For the selection of optimal features from datasets, we have used particle swarm optimization (PSO), firefly optimization (FFA), and shuffled frog leaping algorithm (SFLA) and then compared the performance of classification for these algorithms as well as without applying any feature selection algorithm. The work is implemented on the MATLAB platform and analysis is performed on two different datasets, the Microsoft Malware prediction database, and the Host list from the GitHub database.

Literature Review

This section gives the recent methods that are used for optimization and feature selection for machine learning algorithms.

Oliveira et al. (2015) proposed that optimization is a part of Machine learning (ML). Almost all ML problems are reduced to optimization problems. This work discusses and gives a glimpse of various metaheuristic optimization techniques that have been used in recent times in machine learning. The classification criteria can be used for the meta-heuristics, in terms of the features that follow in the research, memory feature, type of neighbor holding used, or the number of current solutions made from one iteration to the next.

Sun et al. (2019) provide an idea about the principles and progress of the most commonly used optimization methods from the perspective of machine learning. The various approaches discussed are of great significance and can offer guidance for developments in the area of optimization and machine learning research.

Emary et al. (2015) use the Firefly algorithm for feature selection. The proposed methodology is tested on eighteen data sets. The results discussed in the paper show that the firefly algorithm is better for feature search as compared to other methods like particle swarm optimization (PSO) and genetic algorithm (GA).

A feature selection method based on the Firefly algorithm has been suggested by Selvi et al. (2017) to improve big data analysis. The suggested technique was tested on a huge Twitter data set. The effectiveness of the proposed system was proven.

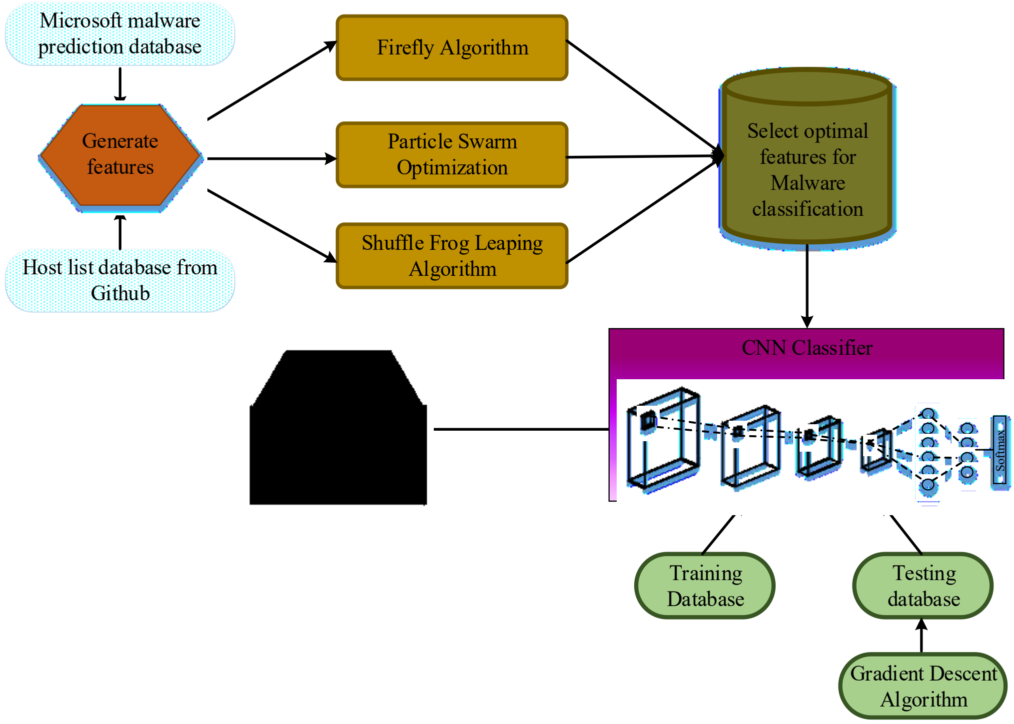
Abdelsalam et al. (2018) in their paper discuss a study on multi-objective (Zhang et al., 2020) particle swarm optimization (PSO) for feature selection. Two PSO-based feature selection algorithms have been studied. In the first algorithm, the idea of non-dominated sorting into PSO for feature selection problems has been presented. In the second algorithm, the ideas of crowding, mutation, and dominance to PSO to search for the Pareto front solutions have been discussed. Finally, a comparison is made for two algorithms based on conventional feature selection methods, a single objective feature selection method, a two-stage feature selection algorithm, and three well-known evolutionary multi-objective algorithms on twelve benchmark datasets.

Xue et al. (2012) in this paper present a new approach for designing fuzzy rule-based classifiers. For feature selection of classifiers, a continuous shuffled frog-leaping algorithm is applied. On a set of constructed classifiers, the optimal classifier is selected in terms of the accuracy and the number of features used, using the statistical Akaike informational criterion.

Methodology

In this paper, we have used the Convolutional Neural Network (Kilincer et al., 2021) to classify data from two datasets: the Microsoft malware prediction database and the Host list database from Github. We classified with and without applying feature selection algorithms. Feature selection algorithms applied in the work include Particle Swarm Optimization (PSO), Firefly algorithm (FFA and Shuffled Frog Leap Optimization (SFLA). The performance of the models is evaluated in terms of F-measure, Recall, Precision, and accuracy. Fig. 1 shows the overall proposed methodology for feature selection and classification.

Figure 1.
Workflow of the proposed methodology



Feature Selection Algorithms

Firefly algorithm for feature selection

Step 1: Initializing the number of features $f(A)$ to detect the malfunction in cloud computing.

Step 2: Initializing the features' performance by considering the equation $f(A) = \{ \textit{best features} \}$. Define the coefficient ω and value λ, which denotes the attraction towards the best feature in the equation $f(A) = \{ \textit{best features} \}$.

Step 3: Evaluate the effectiveness of the features by equation $i\alpha 1/s^2$. The comparison between features is obtained from the equation $y_{t+1} = y_t + \lambda_0 e^{-\omega s^2} + \sigma\xi$.

The gravitation towards the best feature (λ) is given by:

$\lambda = \lambda_0 e^{-\omega s^2}$. Where λ0=λ at 0, ω is the coefficient. The flow of feature 'i' towards more potency feature 'j' is determined by $y_i^{t+1} = y_i^t + \lambda_0 e^{-\omega s^2} y_{ij}(y_j^t - y_i^t) + \sigma_t \xi_i^t$. Where 'i' is the best feature, 'a' is a difference in characteristics of various features, λ denotes attraction towards best features, 'i' denotes features, and 't' denotes iterations. ξ are random features which range from -0.5 to 0.5, σ is the randomization parameter. Whereas $\lambda_0 e^{-\omega s^2} y_{ij} (y_j^t - y_i^t)$ representing the characteristics differences between features. Between two features 'j' is better than 'i' then feature 'i' is omitted.

Step 4: If i=1(for all the features) and j=1 (for all the features that are going to be compared) then compare whether 'j' has better performance than 'i' then hold feature 'j' for the next level.

Step 5: Increase the number of features for execution with the new best features.

that

Step 7: Rank the best features based on the final step of execution and hold the current best features or otherwise repeat the process.

Step 8: The fitness function is updated by equation $\omega = 1/r^m$. The characteristics variation between any two features is given by:

$$r_{ij} = \left\| y_i - y_j \right\| = \sum_{k=1}^{n} (y_{i,j} - y_{j,k})^2$$

The algorithm aims to detect the malicious function in cloud computing by selecting the appropriate feature.

The unwanted features are eliminated based on the performance of each feature in the detection of malware in cloud computing. This study offers a new fitness function which is given by:

$fitness = \sigma \frac{F_{total} F_{leftout}}{F_{total}} + \frac{\lambda}{A}$, where $\lambda + \sigma = 1$, F_{total} is the total number of features, $F_{leftout}$ and the number of features left out.

Particle Swarm Optimization

PSO in feature selection of cloud computing offers the best position to feature based on the existing flow position of the feature. Each feature has different flow characteristics thus PSO is useful to extract the best performed features among the group of features. The best feature is selected in the set of features given in the database as $X = \{x1, x2, x3.......xn\}$ and the range of the features depends upon function $f(x)$. Where x is feature and $f(x)$ is the fitness function which is given by:

$$fitness\ function f(x) = \{best\ feature\} \quad (1)$$

The position movement of each feature to get the best feature is given by:

$$V_{ab}^{t+1} = \alpha V_{ab}^{t} + k_1 h_1^t\ (Obest_{ab} - X_{ab}^{t}) + k_2 h_2^t (Pbest - X_{ab}^{t}) \quad (2)$$

$$X_{ab}^{t+1} = X_{ab}^{t} + V_{ab}^{t+1} \quad (3)$$

Where V is a flow of feature, α is coefficient, X_{ab}^{t} is the current position of the feature, $xbest$ is pre-best the position of the feature, $ybest$ is the new updated position of features, h_1, h_2 are random parameters with [0, 1] range, k_1, k_2 are accelerating constant, $a = \{1,2,3\ldots p\}, b = \{1,2,\ldots n\}$ Equation (2) denotes newly updated feature equation in which a previous number of features are multiplied with 'α'. If α=1, then the feature's selection is fully influenced by the previous number of features, to reduce this 'α' should be in $0 < \alpha \leq 1$.

The first step in the process of detecting a malfunction in cloud computing using the particle swarm optimization algorithm is initializing the number of features to identify the malfunction. After the first step of execution, update the flow and position of features. If the difference in the position of the flow of features is increased, then the value of (Obest_ab-X_ab^t) gets increased. Hence this term increases the attraction towards the best results which is the best feature. Remove the unwanted features that have unique characteristics in the null value. If the current results are better than the previous one, then assign all the new features to detect the malware. Increase the number of iterations until you get the best feature.

Shuffled Frog Leaping Optimization

Step 1: Initializing the total number of features that are going to be used in the detection of malware. The number of features is denoted in the equation $S_i = (J_1, J_2, J_3 \ldots\ldots Jv)$. The q^{th} feature is represented by $Jq = (J_q^1, J_q^2, \ldots\ldots J_q^z)$ for d-dimensional problems. Then compute the performance value of f(i) for each J(t).

Step 2: Examine the characteristics of each feature performance by equation $B = \{S(i), f(i), i = 1,2,\ldots.S$ and grouping all the features. The step size is denoted by the equation $d = p \times n$.

Step 3: Sort the features in descending order according to their performance. For each step of execution, compare the best feature in a group with the worst feature.

Step 4: If the results from the above process are satisfied, then renew the worst feature and shuffle all the features, then check whether the resulting feature is best, or otherwise go to the next step.

Step 4: The sub-class is used to concentrate more on the best features, and to eliminate the unwanted poorly performed feature. On the sub-class process, the best-performed features are considered. The unwanted feature is replaced by introducing a random feature (R).

Step 5: The variation in each feature is obtained from the equation $M^k = f(k + n(q-1)) \quad q = 1,2,3,......p, \quad k = 1,2,3...n$. Categories the best and worst performing features and hold the current best feature for the next step.

Step 6: Evaluate the current best features with the group of best features, then evaluate the individual features randomly. Shuffle all the features then validate the best features.

Step 7: If the resulting features are having the best performance, then stop the process or otherwise repeat the process.

Step 8: The newly determined best features were obtained by using the equation $q_j = \frac{2(p+1-q)}{p(p+1)} \quad q = 1,2,...p$ which denotes features with the best performance having the highest probability, $q = \frac{2}{p+1}$ is considered for the subclass, and worst performing having the lowest probability $q_n = \frac{2}{p(p+1)}$. In the subclass, features are selected randomly from a total number of frogs (n) to form a subclass array of features C and record the performance. In the Z-dimensional problem, let F_{best} is the best feature and F_{worst} is the worst feature and the global best one is F_{global} the feature selection is renewed F_{new}.

$$Z_i = \begin{cases} \min\{\text{int}[r(F_{best} - F_{new})], Z_{\max}\} & r(F_{best} - F_{worst}) \geq 0 \\ \max\{\text{int}[r(F_{best} - F_{new})], -Z_{\max}\} & r(F_{best} - F_{worst}) \leq 0 \end{cases} \tag{4}$$

Where ‘int (F)’ is the roundness of F. ‘r’ is the random number range of (0-1) and $Z_{\max}$ denotes maximum step size allocation. The resulting feature is obtained by the following expression:

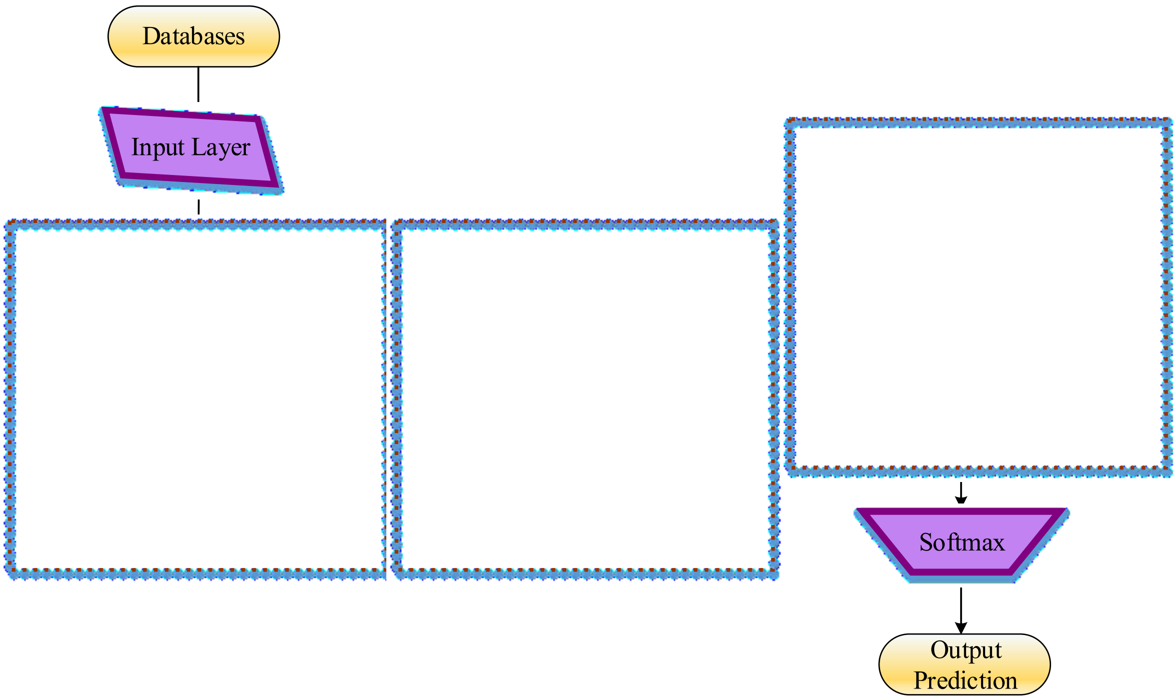
$$F_{new\vartheta} = Z_i + F_{new} \tag{5}$$

Examine the achieved feature $f(F_{new\vartheta})$, if getting the best result then replace F_{new} by $F_{new\vartheta}$. Random feature ‘R’ is generated in the feasible range to replace the features that have undesirable performance and compute f(R) set $f(R) = f(F_{new9})$ and $F_{new9} = R$. Whether 'in < n' and 'iM < M' then increase both by one. The iteration would not end until the satisfied output was obtained. After all the searches were completed within the features, then the halt conditions were implemented by the following criteria such as; while the optimum solution is obtained or there is no improvement in consecutive steps of the process and the predefined number of iterations were executed.

Convolutional Neural Network

We apply the Convolutional Neural Network (CNN) for classifying the data (Albawi et al., 2017) before and after applying three feature selection algorithms: FFA, PSO, and SFLA.

Figure 2.
Structure of CNN



CNN is used with two convolutional and pooling layers, two fully connected layers along with the input and output layer is used, as shown in (Figure 2). The activation function used is ReLU for making the negatives nullify and at the end, SoftMax function is applied which will map the values to 0 and 1. The gradient descent method is used to minimize the error.

Results

The CNN model has been evaluated without applying any feature selection algorithm and then with three feature selection algorithms: FFA, PSO, and SFLA.

(i) Evaluations of CNN without feature selection

The CNN model has been evaluated with the datasets separately without using the feature selection algorithms. This evaluation is done to identify the effectiveness of the feature selection algorithms in malicious node detection. The results of the evaluations on different datasets are presented (Figure 3) and (Figure 4) below:

Figure 3.

Performance of CNN for the Microsoft malware prediction database without feature selection

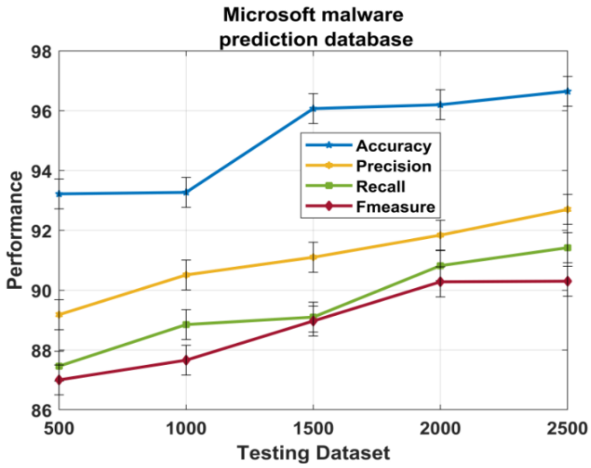
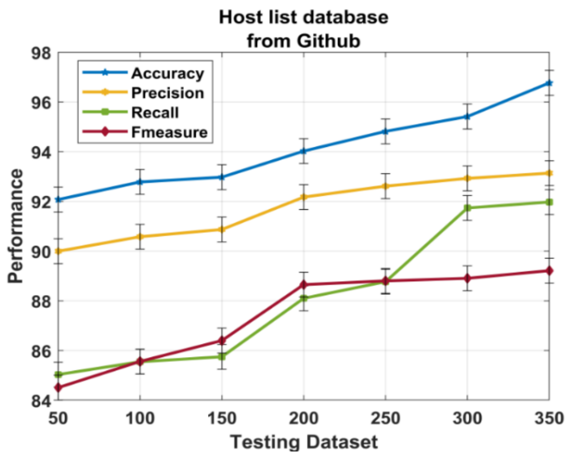


Figure 4.

Performance of CNN for the host list database without feature selection



(ii) Evaluations of CNN with feature selection

All those extracted features are given to the CNN-based classifier for the accurate prediction of malicious hosts on the cloud. The performance parameters like true positive rate (TPR) as well as false positive rates (FPR) are given in (Figure 5) and (Figure 6), malicious host count is shown in (Figure 7), f-measure in (Figure 8), precision in (Figure 9) and recall values are shown in (Figure 10)

Figure 5.
false positive rate

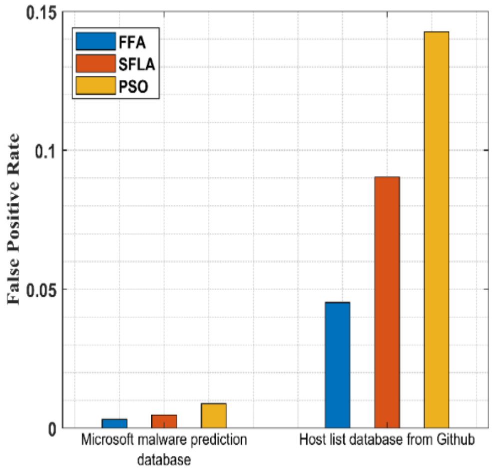


Figure 6.
true positive rate

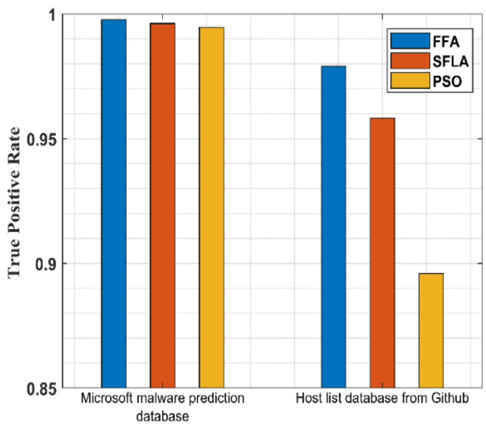


Figure 7.

Malicious host count for a) Microsoft malware prediction database b) Host list database from GitHub

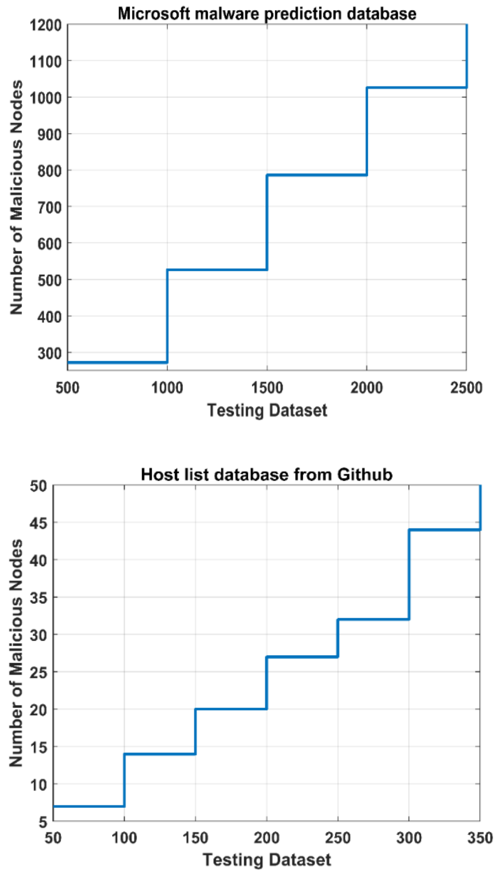


Figure 8.

F-Measure for Microsoft and Github database

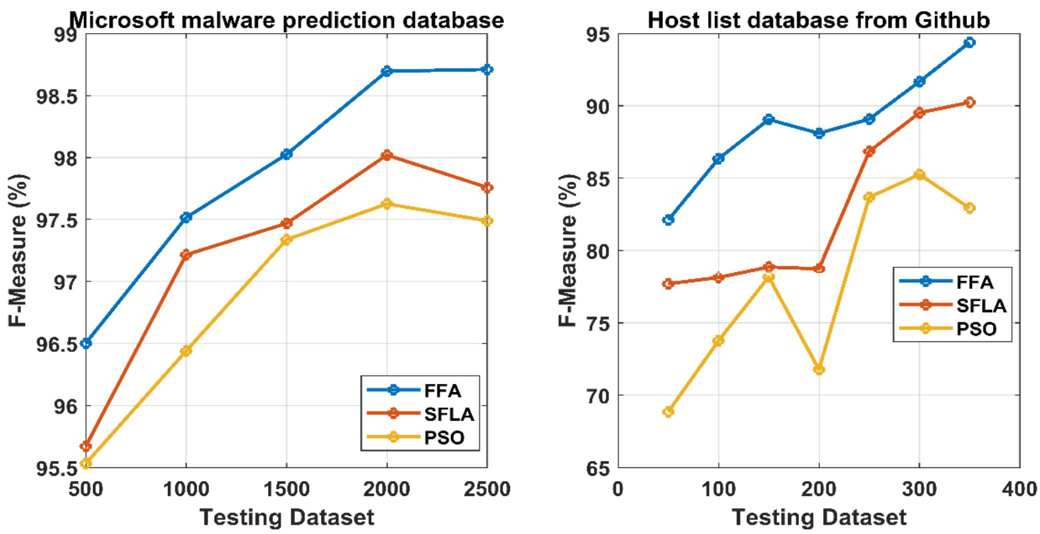


Figure 9.

Precision for Microsoft and GitHub database

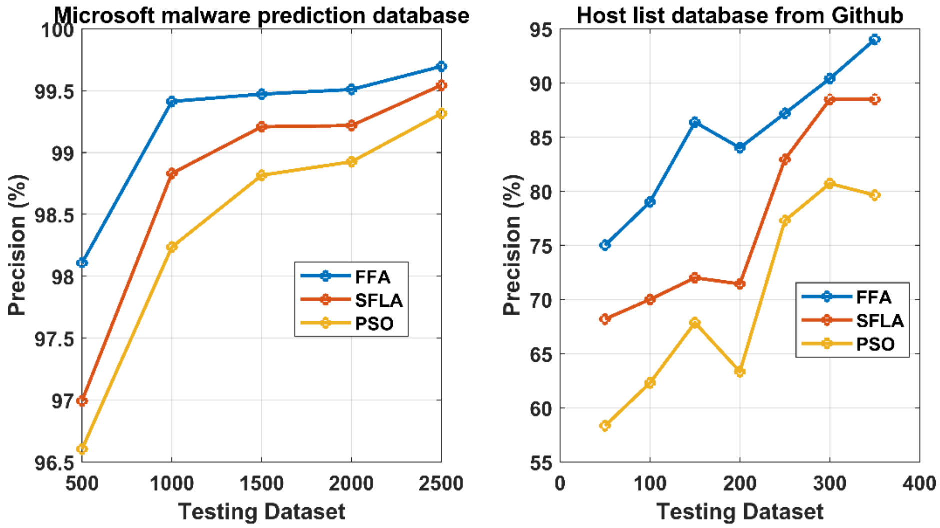


Figure 10.

Recall for Microsoft and GitHub database

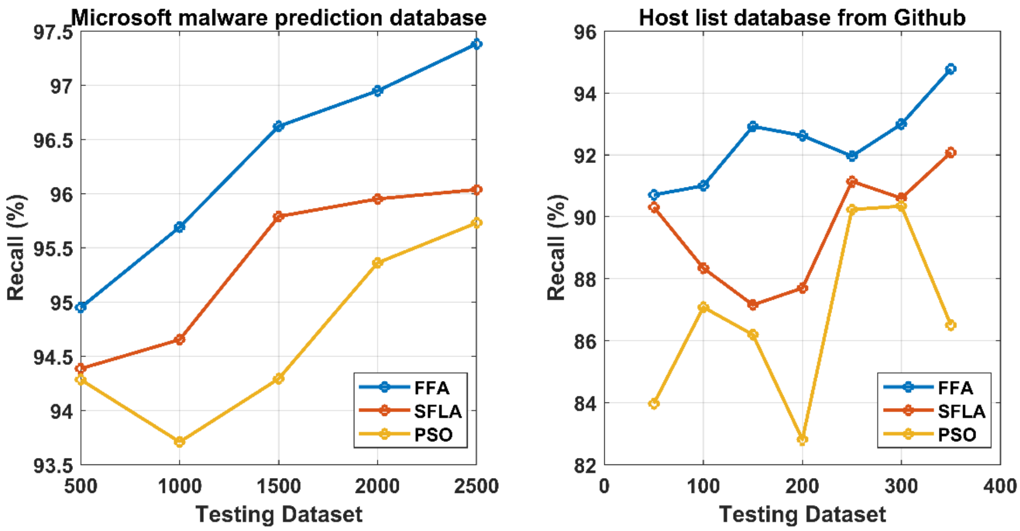


Table 1.

Accuracy of Malware prediction methods

Algorithm	Accuracy for Microsoft database	Accuracy for GitHub database
FFA	**99.84**	**99.09**
PSO	99.76	98.18
SFLA	99.64	96.67

(Table 1) gives the accuracy of the malware prediction system for the three various algorithms. From the tabulation, the FFA method has given the best result for malicious detection in cloud computing when used with the CNN classifier.

Conclusion

In this paper, the performance of the various optimization algorithms for the feature selection phase of malicious detection in the cloud environment is presented. The feature selection process of the FFA, PSO, and SFLA is studied, and from that it is proved that the FFA algorithm has shown better performance. Those algorithms are analyzed for malware detection by considering two sets of databases i.e. Microsoft malware prediction and Host list from GitHub database. By having a smaller number of optimal features on both databases, the malicious behavior is well predicted by the FFA algorithm with high accuracy. Meanwhile, the number of features selected by the other two algorithms is high and yields low accuracy in contrast to the FFA. After implementing the process on MATLAB, the accuracy of 99.84% and 99.09% are obtained for 2,500 and 330 testing data from the Microsoft malware prediction and Host list from the GitHub database respectively by the FFA method.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Abdelsalam, M., Krishnan, R., Huang, Y., & Sandhu, R. (2018, July). Malware detection in cloud infrastructures using convolutional neural networks. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 162-169). IEEE.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.

Datta, S., Santra, P., Majumder, K., & De, D. (2018). An automated malicious host recognition model in cloud forensics. In *Networking Communication and Data Knowledge Engineering* (pp. 61-71). Springer, Singapore

Emary, E., Zawbaa, H. M., Ghany, K. K. A., Hassanien, A. E., & Parv, B. (2015, September). Firefly optimization algorithm for feature selection. In *Proceedings of the 7th Balkan Conference on Informatics Conference* (pp. 1-7).

Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Computer Networks*, *188*, 107840.

Martini, B., & Choo, K. K. R. (2012). An integrated conceptual digital forensic framework for cloud computing. *Digital investigation*, *9*(2), 71-80.

Mishra, S., Sharma, S. K., & Alowaidi, M. A. (2021). Analysis of security issues of cloud-based web applications. *Journal of Ambient Intelligence and Humanized Computing*, *12*(7), 7051-7062.

Oliveira, P., Portela, F., Santos, M. F., Abelha, A., & Machado, J. (2015). Machine Learning: an overview of optimization techniques. *Recent Adv Comput Sci*.

Selvi, R. S., & Valarmathi, M. L. (2017). An improved firefly heuristics for efficient feature selection and its application in big data. *Biomedical Research*, *28*, S236-S241.

Singh, D. A. A. G., Priyadharshini, R., & Leavline, E. J. (2018). Cuckoo optimization-based intrusion detection system for cloud computing. *International Journal of Computer Network and Information Security*, *9*(11), 42.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, *50*(8), 3668-3681.

Xue, B., Zhang, M., & Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, *43*(6), 1656-1671.

Zhang, Z., Wen, J., Zhang, J., Cai, X., & Xie, L. (2020). A many objective-based feature selection model for anomaly detection in cloud environment. *IEEE Access*, *8*, 60218-60231

Bibliographic information of this paper for citing:

Goyal, Nandita & Chandra Trivedi, Munesh (2023). Metaheuristic Algorithms for Optimization and Feature Selection in Cloud Data Classification Using Convolutional Neural Network. *Journal of Information Technology Management*, 15 (3), 99-112. https://doi.org/ 10.22059/jitm.2023.93627