International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I8P222

Volume 70 Issue 8, 216-223, August 2022

Original Article

Elephant Herd Optimization with Weighted Extreme Learning Machine based PDF Malware Detection and Classification Model

[1]P. Pandi Chandran, [2]Hema Rajini. N and [3]M. Jeyakarthic

[1]*Department of Computer and Information Science, Faculty of Science, Annamalai University, Chidambaram, 608002, India*
[2]*Department of CSE, Alagappa Chettiar Government College of Engineering and Technology, Karaikudi, 630003, India*
[3]*Department of Computer and Information Science, Annamalai University, Chidambaram, 608002, India*
[1]pandichandranresearch@gmail.com

Received: 17 June 2022 Revised: 11 August 2022 Accepted: 17 August 2022 Published: 22 August 2022

Abstract *- Portable Document Format (PDF) is widely utilized for document exchange and distribution because of its high portability and universal usage. Benign users and attackers leverage the format's adaptable and flexible nature to utilize different vulnerabilities, overcome security limitations, and then convert the PDF format into one of the foremost malicious code spread vectors. Investigation of the content in the malicious PDF for extracting major features plays a vital role in the automated identification of new attacks. This study develops an Elephant Herd Optimization with Weighted Extreme Learning Machine (EHO-WELM) based PDF malware detection and classification model. The presented EHO-WELM model mainly aims to determine the existence of PDF malware. For attaining this, the EHO-WELM model pre-processes in two ways: categorical encoding and null value removal. In addition, the pre-processed data are passed into the WELM model to identify and classify PDF Malware. For determining the weight values of the WELM model, the EHO algorithm is applied to improve the classifier efficacy, showing the work's novelty. The simulation analysis of the EHO-WELM model on the benchmark dataset implied superior outcomes over the existing approaches.*

Keywords *- Malware detection, Weighted extreme learning machine, PDF Malware, Elephant herd optimization, Parameter tuning.*

1. Introduction

Portable Document Format, widely called PDF, was introduced in 1993 as a de-facto format for dissemination and file exchange [1]. The extensive adaption of this document is because of its inherent flexibility and portable nature. PDF files might have different media (pictures, text), and implanted documents or code would be executed and interpreted through the analysis software. This final capability makes the PDF adaptable to many traditional necessities [2]. Notwithstanding the number of possibilities and the complexity the file format provides, the end user treats PDF files as immutable, plain, and static documents, with no knowledge that the reader software displays the outcome of implementing possibly complicated programs. Recently, malicious actors have exploited the absence of alertness, in conjunction with the existence of susceptibilities in conventional PDF readers [3], to make PDF a very effective direction for malware dissemination. Malware is malicious software utilized by breaching a computer system security policy regarding the availability, confidentiality, and integrity of information [4]. Malware is of dissimilar kinds: spyware, virus, adware, trapdoor, Trojan horse, etc. Based on the method of imposing a threat to the scheme. It could purposely remove, add, or change programs to damage the system's essential function and attain the identity without permission [5, 6].

Malware classification and detection is the key challenge in the cyber security field. A signature-based method is efficient against recognized malware but unsuccessful against unknown and advanced malware [7, 8]. Wang et al. [9] presented a common structure, so-called AdvAndMal comprising a 2layer network for adversarial trained for generating adversarial instances and enhancing the efficiency of classifications from Android malware recognition and family classifier. The adversarial instance generation layer was collected from the conditional generative adversarial network (GAN) named pix2pix, creating malware variations for extending the classifications training set. The malware classifier layer was trained by RGB image visualization in the order of system call.

Sethi et al. [10] presented a structure for detecting and classifying distinct files (for instance, exe, pdf, php, and so on) as benign and malicious, utilizing 2 level classification such as Macro (to detect malware) and Micro (to the classifier of malware files as Ad-ware, Trojan, Spyware, and so on). This solution utilizes Cuckoo Sandbox to create static and dynamic analysis reports by implementing the instance files from virtual environments. Gao et al. [11] present a new method for Android malware recognition and familial classifier dependent upon Graph Convolutional Network (GCN). The common idea is to map apps and Android APIs to a huge heterogeneous graph, altering the original issue as to the node classifier task.

Roseline et al. [12] utilize the visualization method in which malware has been demonstrated as a 2D image and presents a robust ML-related anti-malware solution. The presented method was dependent upon a layered ensemble method which simulates the key features of DL approaches however executes superior to the latter. The presented method is not needed hyperparameter tuning or back-propagation (BP) and works with decreased method difficulty. Reddy et al. [13] presented a new approach which exploits community recognition property and social network analysis models. The presented technique depended on the system call graph attained by removing the system call established from implementing malware files.

For analyzing the inherent features of distinct malware families, features can be removed following community and social network property and utilized in the classifier. Though several models are available in the literature, there is a need to improve the PDF malware classification performance. In addition, the parameter tuning of the ML models becomes essential to improve the overall detection efficiency.

This study develops an Elephant Herd Optimization with Weighted Extreme Learning Machine (EHO-WELM) based PDF malware detection and classification model. The presented EHO-WELM model undergoes pre-processing in two ways: categorical encoding and null value removal. In addition, the pre-processed data are passed into the WELM method to identify and classify PDF Malware. For determining the weight values of the WELM model, the EHO technique was enforced to improve the classifier efficacy. The simulation analysis of the EHO-WELM method on the benchmark dataset implied superior outcomes over the existing approaches.

2. The Proposed Model

In this study, a new EHO-WELM model has been developed to detect and classify PDF malware. The presented EHO-WELM model aims to determine the existence of PDF malware. The EHO-WELM model encompasses three subprocesses: data preprocessing, WELM classification, and EHO-based parameter optimization. Fig. 1 shows the workflow of the EHO-WELM model.

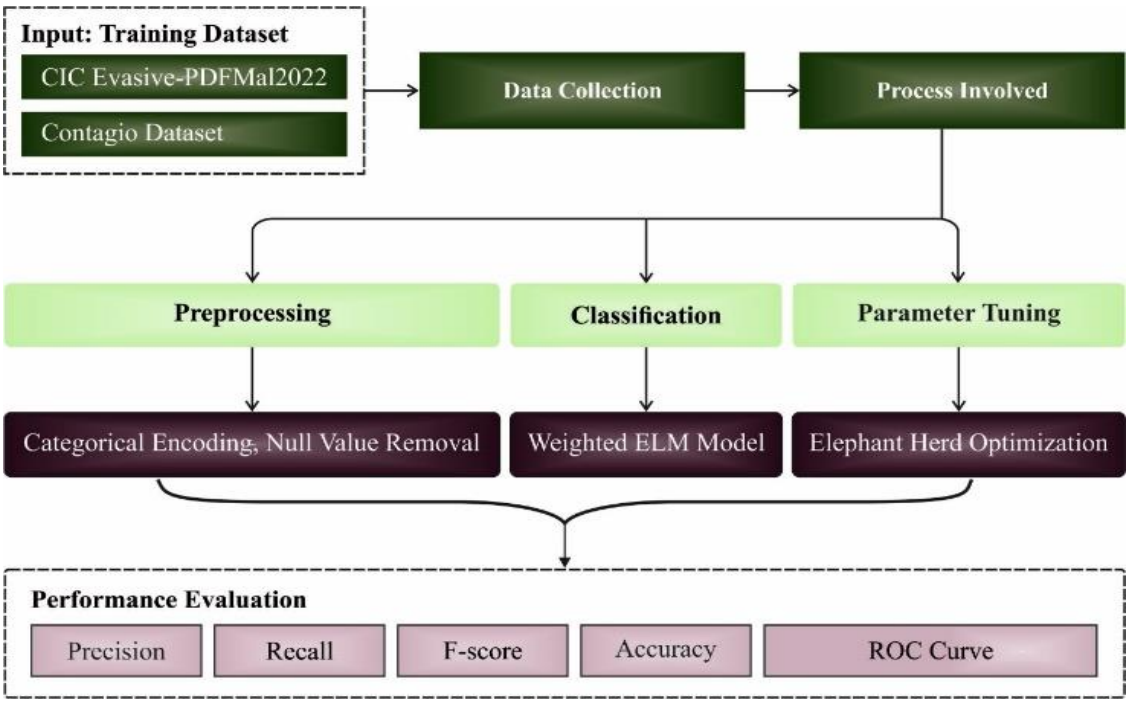

Fig. 1 Workflow of EHO-WELM model

2.1. Data Pre-processing

The preliminary level of the EHO-WELM model is the data preprocessing stage, which comprises two levels: categorical encoding and null value removal. At the initial stage, the categorical values undergo encoding into numerical values. Next, the null values which are available in the dataset are removed. In the next stage, the classification of PDF documents takes place using the WELM model.

2.2. WELM-based classification

At this stage, the pre-processed data are passed into the WELM model to identify and classify PDF Malware. ELM was utilized to classify the balanced dataset in WELM was employed to classify the imbalanced data set. Therefore, this section explains the establishment of WELM [14]. The structure of ELM is given in Fig. 2. The trained data set has N separate samples $(x_i, z_i), i = 1,2, \dots, N$. The single hidden layer (HL) NN with L HL node is established as:

$$\sum_{i=1}^{L} \beta_i \cdot l\left(w_i \cdot x_j + b_i\right) = z_j, j = 1, \dots, N \tag{2}$$

In which w_i represents the single HL input weighted, $l()$ determines the activation function, β_i implies the outcome weight, and b_i denotes the single HL bias. At this point, Eq. (2) is showcased as:

$$S\beta = T \tag{3}$$

In which S demonstrates the single HL outcome matrix [15]:

$$S(w_1, \dots, w_L, b_1, \dots, b_L, x_1, \dots, x_N) = \begin{pmatrix} l(w_L \cdot x_1 + b_1) & \cdots & l(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ l(w_L \cdot x_N + b_1) & \cdots & l(w_L \cdot x_N + b_L) \end{pmatrix}_{N \times L} \tag{4}$$

As per the Karush-Kuhn-Tucker model, a Lagrangian factor has been presented to transform ELM's training into a dual problem. The outcome weight β is computed as:

$$\beta = S^T \left(\frac{1}{C} + SS^T\right)^{-1} T \tag{5}$$

In which C demonstrates the regularized co-efficient. So, the resultant function of the ELM classifier method was showcased as:

$$F(x) = s(x)S^T \left(\frac{1}{C} + SS^T\right)^{-1} T = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^{\mathrm{T}} \left(\frac{1}{C} + \chi\right)^{-1} T \tag{6}$$

In which χ defines the kernel matrix that is calculated as:

$$\chi = SS^T = s(x_i)s(x_j) = K(x_i, x_j) \tag{7}$$

It can be apparent in (5) that the HL feature map $s(x)$ has independent of the outcome ELM classifier, and the classification outcome was related to kernel function (KF) $K(x, y)$. Then, the $K(x, y)$ executes the inner product. Thus the HL node count is no result on the output. Hence, the ELM is not needed to set the input weighted and offset of HL. The KF of KELM is RBF function, as demonstrated under:

$$K(x, y) = \exp\left(-\gamma \|x - y\|^2\right). \tag{8}$$

Therefore, the KELM classifier was implemented with 2 parameters, KF parameter γ and penalty parameter C. According to the ELM benefits, the WELM with assigned weighted for many instances accomplished the imbalanced classifier problem. Its outcome function has been calculated as:

$$F(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^{\mathrm{T}} \left(\frac{1}{C} + W\chi\right)^{-1} WT, \tag{9}$$

$$W = diag\ (w_{ii}), i = 1,2, \dots, N \tag{10}$$

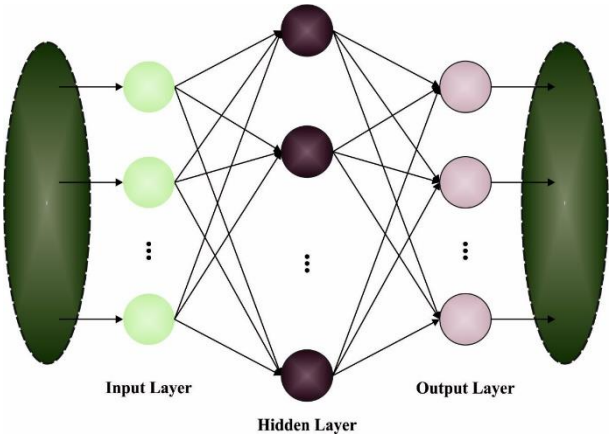

Fig. 2 Structure of ELM

In which W defines the weighted matrix. The WELM has 2 weighted approaches, i.e.,

$$w_{ii} = \frac{1}{\#(z_i)}, \tag{11}$$

$$w_{ii} = \begin{cases} \frac{0618}{\#(z_i)}, if\, z_i > \overline{z}_i \\ \frac{1}{\#(z_i)}, otherwise \end{cases} \tag{12}$$

whereas $\#(z_i)$ stands for the count of instances going to class $z_i, i = 1, \ldots, m$. m refer to the count of classes. $\overline{z}_i$ refers to the average of total instances of each class. In WELM, the regularized co-efficient C and bandwidth of RBF kernel γ becomes vital for determining the efficacy of the implementation of WELM.

2.3. EHO-based Parameter Optimization

For determining the weight values of the WELM method, the EHO algorithm can be applied to improve the classifier efficacy [16]. EHO approach largely depends on the nature of elephants, recently introduced for global optimization. This approach doesn't take advantage of the previous individuals in the upgrade procedure. Once the useful information in the preceding individual is fully employed in the optimization method, the solution quality would be significantly amended.

2.4. Clan Updating Operator

The updated principles of elementary EHO are proposed [16]. Assume an elephant clan that is represented by ci. Next, the forthcoming location of the elephant, j, in the clan is exploited by the following equation:

$$x_{new,ci,j} = x_{ci,j} + \alpha \times \left(x_{best,ci} - x_{ci,j}\right) \times r, \quad (13)$$

Where$x_{ci,j}$ denotes the preceding location of elephant j in the clan, and$x_{new,ci,j}$ signifies the upgraded position. $x_{best,ci}$ symbolizes the matriarch of clan ci; she characterizes the fitting elephant in the clan. The scaling factor $\alpha \in [0,1]$ is employed in estimating the matriarch of ci on $x_{ci,j}$. $r \in [0,1]$ belongs to stochastic distribution that can provide the development of population diversity. It is noted that $x_{ci,j} = x_{best,ci}$That denotes that a matriarch in a clan couldn't be exploited. This situation is removed by upgrading the location of the fitting elephant through the following equation:

$$x_{new,ci,j} = \beta \times x_{center,ci}, \quad (14)$$

Now $x_{center,ci}$ on $x_{new,ci}$, is preserved by $\beta \in [0,1]$. The information acquired from each individual in clan ci is employed for evolving a novel individual $x_{new,ci,j}$. The in-between part of clan ci, $x_{center,ci}$, can be described for d-th dimensional vector by D calculation where D indicates the dimension vector as follows:

$$x_{center,ci,d} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j,d} \quad (15)$$

Here, $1 \le d \le D$ shows the d-th dimensional vector, n_{ci} indicates the individual in ci, and $x_{ci,j,d}$ indicates dth dimensional vector of individual $x_{ci,j}$.

2.5. Separating Operator

Generally, the ME leaves the family member and lives alone when they get mature. The isolation procedure can be labelled by appropriate splitting operators in solving the optimization issue. For improving the searching capability of EHO, assume an elephant with poor fitness for executing the separation operator for each generation in the following.

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times rand \quad (16)$$

Now x_{max} and x_{min} denotes the upper and lower bounds, respectively. $x_{worst,ci}$ indicates the poor individual elephant in clan ci. $rand \in [0,1]$ implies a kind of uniform and stochastic distribution from [0,1] as employed in current works.

Algorithm 1: Elephant Herd Optimization (EHO)

Begin
Step 1: Initialization.
Fix the generation counter $t = 1$.
Began the population P of NP elephant individuals at random by uniform distribution in the searching region.
Fix the amount of elephants $nKEL$, the maximal generation $MaxGen$, the scale factor α and β, the amount of clan $nClan$, and the total of elephants for the ci-th clan n_{ci}.
Step 2: Fitness calculation.
Evaluate elephant individuals based on their location.
Step 3: While $t < MaxGen$:
Establish the elephant individual based on its fitness.
Protect the $nKEL$ elephant individual. Perform the clan update operator as shown in Algorithm 1.
Perform the separation operator as shown in Algorithm 2.
Describe the population as per the recently upgraded position.
Exchange the poor elephant with $nKEL$ optimal one.
Upgrade the generation counter, $t = t + 1$.
Step 4: End while
Step 5: Output the optimal solution.
End.

3. Performance Validation

In this section, the PDF malware detection and classification outcomes of the EHO-WELM model are tested using CIC Evasive-PDFMal2022 dataset [17] and Contagio Dataset [18]. The former dataset includes 32 attributes, whereas the latter dataset includes 136 datasets.

Fig. 3 establishes the training accuracy (TA) and validation accuracy (VA) offered by the EHO-WELM model on the CIC Evasive-PDFMal2022 dataset. The results reported that the EHO-WELM model had accomplished

maximum TA and VA with increased epochs. It is also revealed that the VA is considerably higher than the TA.

Fig. 4 confirms the training loss (TL) and validation loss (VL) gained by the EHO-WELM model on CIC Evasive-PDFMal2022 dataset. The figure labelled that the RDADL-HIC model has presented decreased TL and VL with an increase in epoch count. It is noticeable that the VL is lower compared to TL.

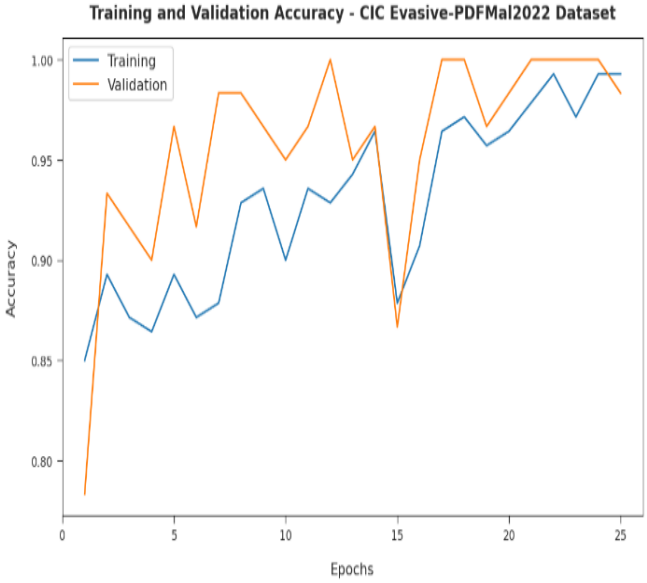

Fig. 3 TA and VA of EHO-WELM model on CIC Evasive-PDFMal2022 dataset

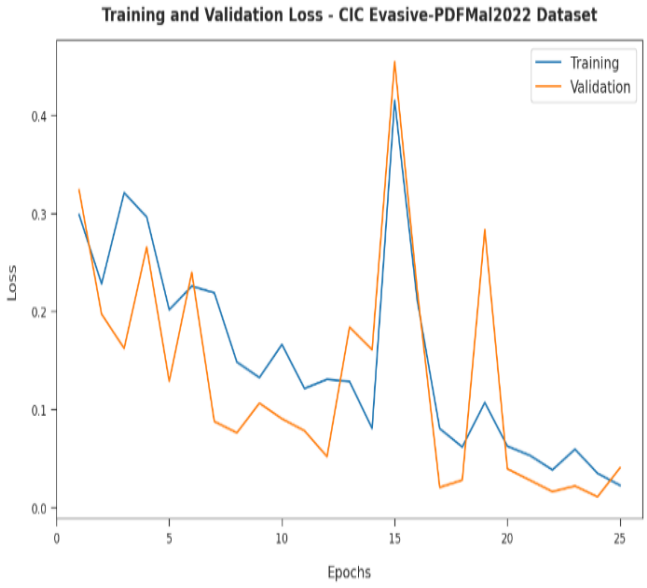

Fig. 4 TL and VL of EHO-WELM model on CIC Evasive-PDFMal2022 dataset

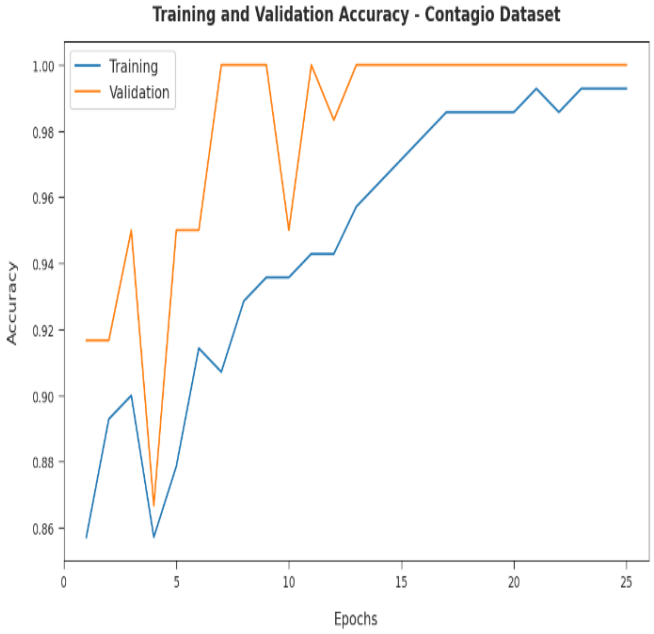

Fig. 5 TA and VA of EHO-WELM model on Contagio Dataset

Fig. 5 establishes the TA and VA offered by the EHO-WELM model on the test Contagio Dataset. The results stated that the EHO-WELM model has demonstrated improved TA and VA with increased epochs. It is also revealed that the VA is considerably higher than the TA.

Fig. 6 depicts the TL and VL gained by the EHO-WELM model on the test Contagio Dataset. The figure labelled that the RDADL-HIC model has presented decreased TL and VL with an increase in epoch count. It is noticeable that the VL is lower compared to TL.

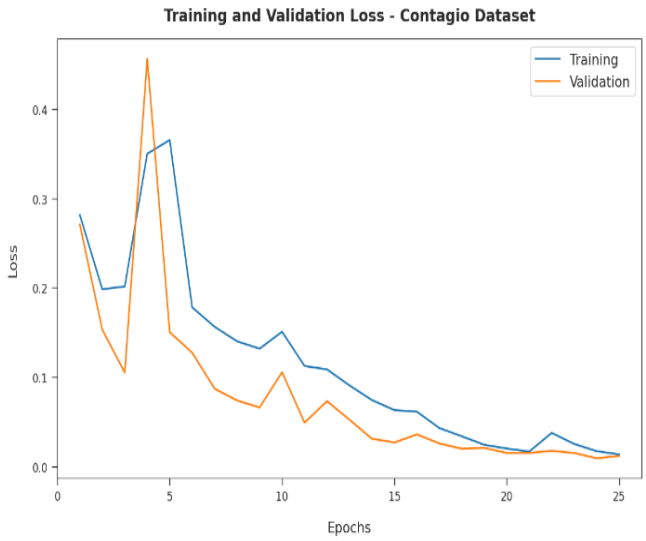

Fig. 6 TL and VL of EHO-WELM model on Contagio Dataset

Table 1 provides a detailed comparative malware classification outcomes of the EHO-WELM model with the WELM model on two datasets. Fig. 7 portrays a brief comparative investigation of the EHO-WELM technique with the standard WELM model on the Contagio Dataset.

Table 1. Comparative results of EHO-WELM with WELM models

Measures	CIC Evasive-PDFMal2022		Contagio Dataset	
	WELM	EHO-WELM	WELM	EHO-WELM
Accuracy	89.69	93.91	92.04	96.94
Precision	90.26	93.16	90.55	96.82
Recall	90.52	91.85	90.53	96.88
F1-Score	88.54	92.46	93.35	96.93
AUC-Score	90.66	92.56	93.25	96.99

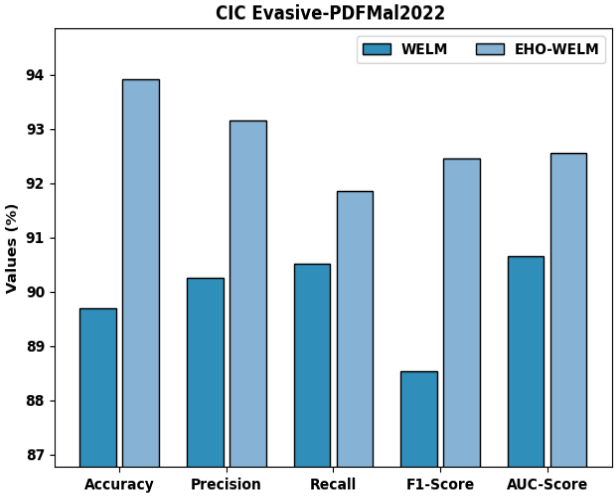

Fig. 7 Comparative results of EHO-WELM with WELM models on CIC Evasive-PDFMal2022 dataset

The obtained values denoted that the EHO-WELM model has the capability of reaching enhanced PDF malware detection and classification outcomes with increased $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC_{score} of 96.94%, 96.82%, 96.88%, 96.93%, and 96.99% respectively. Moreover, the standard WELM model has reached decreased PDR malware detection outcomes with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC_{score} of 92.04%, 90.55%, 90.53%, 93.35%, and 93.25% respectively.

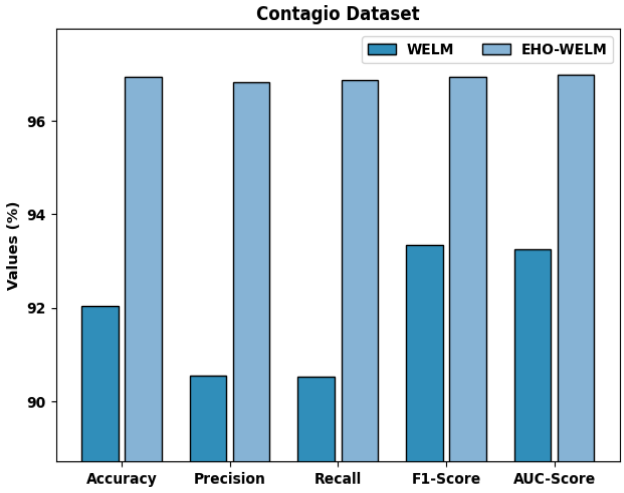

Fig. 8 Comparative results of EHO-WELM with WELM models on Contagio Dataset

Fig. 8 reports a brief comparative study of the EHO-WELM model with the standard WELM model on the CIC Evasive-PDFMal2022 dataset. The experimental outcomes implied the EHO-WELM method can attain improved PDF malware detection and classification outcomes with maximum $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC_{score} of 93.91%, 93.16%, 91.85%, 92.46%, and 92.56% correspondingly. In parallel, the standard WELM method has resulted in lowering PDR malware detection outcomes with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC_{score} of 89.69%, 90.26%, 90.52%, 88.54%, and 90.06% respectively.

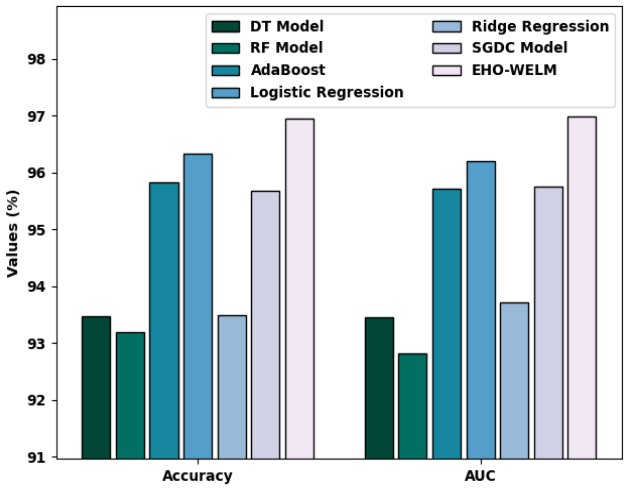

Fig. 9 Comparative study of EHO-WELM model with recent models-I

The detailed comparative analysis was exhibited in Figs to assure the enhanced performance of the EHO-WELM method. 9-10 [19]. The outcomes exhibited that the DT, RF, and ridge regression methods resulted in lower classification outcomes.

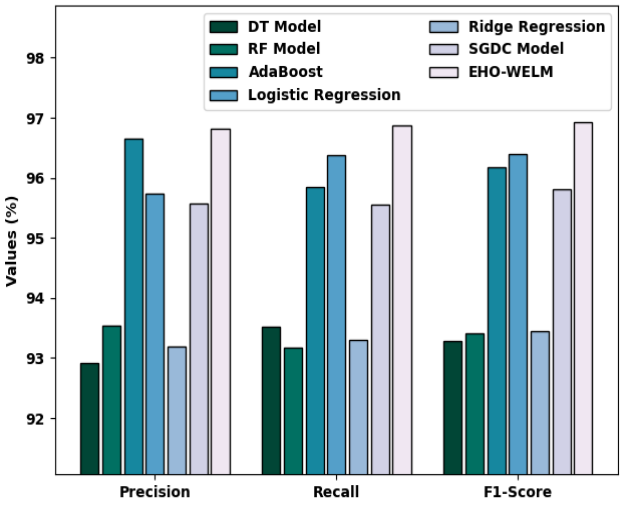

Fig. 10 Comparative study of EHO-WELM model with recent models-I

The AdaBoost and SGDC models have reached a slightly enhanced classification outcome. Moreover, the LR model has accomplished reasonable performance over the other models. At last, the EHO-WELM model has demonstrated superior PDF malware classification performance with $accu_y$ of 96.94%, $prec_n$ of 96.82%, $reca_l$ of 96.88%, $F1_{score}$ of 96.93%, and AUC of 96.99%. These results and discussion reported the effective outcomes of the EHO-WELM model on the detection and classification of PDF malware.

4. Conclusion

This study has developed new EHO-WELM-based PDF malware detection and classification model. The presented EHO-WELM model mainly aims to determine the existence of PDF malware. For attaining this, the EHO-WELM model pre-processes in two ways: categorical encoding and null value removal. In addition, the pre-processed data are passed into the WELM model to identify and classify PDF Malware. For determining the weight values of the WELM model, the EHO algorithm is applied to improve the classifier efficacy. The simulation analysis of the EHO-WELM model on the benchmark dataset implied superior outcomes over the existing approaches. In future, feature selection and feature reduction methods can be developed to boost PDF malware detection efficiency.

References

[1] Gibert, D., Mateu, C. and Planes, J, " the Rise of Machine Learning for Detection and Classification of Malware: Research Developments, Trends and Challenges," *Journal of Network and Computer Applications*, vol.153, P.102526, 2020.

[2] Komatwar, R. and Kokare, M, " A Survey on Malware Detection and Classification*," Journal of Applied Security Research*, vol.16, no.3, pp.390-420, 2021.

[3] Wang, J., Xue, Y., Liu, Y. and Tan, T.H, " Jsdc: A Hybrid Approach for Javascript Malware Detection and Classification," *in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pp.109-120, 2015.

[4] Vinayakumar, R., Soman, K.P. and Poornachandran, P, " Deep Android Malware Detection and Classification," *in 2017 International Conference on Advances in Computing, Communications and Informatics* (ICACCI), pp.1677-1683, 2017.

[5] Singh, J. and Singh, J, " A Survey on Machine Learning-Based Malware Detection in Executable Files," *Journal of Systems Architecture,* vol.112, pp.101861, 2021.

[6] Nishant Jakhar, Rainu Nandal, Kamaldeep, "Design of A Rule-Based Decisive Model for Optimizing the Load Balancing in A Smart Grid Environment," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 97-103, 2022.

[7] Miguel Fernández, Avid Roman-Gonzalez, "A Multi-Objective Approach to Modelling the Integrated Resource Selection and Operation Sequences Problem in Production System," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 51-56, 2022.

[8] T V Divya, Barnali Gupta Banik, "An Integrated Cycle GAN and PEGASUS to Generate Synthetic Data for Detection of Fake News," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 57-70, 2022.

[9] Wang, C., Zhang, L., Zhao, K., Ding, X. and Wang, X, " Advandmal: Adversarial Training for Android Malware Detection and Family Classification," Symmetry, vol.13, no.6, pp.1081, 2021.

[10] Sethi, K., Chaudhary, S.K., Tripathy, B.K. and Bera, P, " A Novel Malware Analysis Framework for Malware Detection and Classification Using Machine Learning Approach," *in Proceedings of the 19th International Conference on Distributed Computing and Networking*, pp. 1-4, 2018.

[11] Gao, H., Cheng, S. and Zhang, W, " Gdroid: Android Malware Detection and Classification with Graph Convolutional Network," *Computers & Security*, vol.106, P.102264, 2021.

[12] Roseline, S.A., Geetha, S., Kadry, S. and Nam, Y, "Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm," *IEEE Access*, vol.8, pp.206303-206324, 2020.

[13] Reddy, V., Kolli, N. and Balakrishnan, N, " Malware Detection and Classification Using Community Detection and Social Network Analysis," *Journal of Computer Virology and Hacking Techniques*, vol.17, no.4, pp.333-346, 2021.

[14] Zong, W., Huang, G.B. and Chen, Y, " Weighted Extreme Learning Machine for Imbalance Learning," *Neurocomputing*, vol., 101, pp.229-242, 2013.

[15] Yu, H., Yang, X., Zheng, S. and Sun, C, " Active Learning From Imbalanced Data: A Solution of Online Weighted Extreme Learning Machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no.4, pp.1088-1103, 2018.

[16] Wang, G.G., Deb, S. and Coelho, L.D.S, " Elephant Herding Optimization," *in 2015 3rd International Symposium on Computational and Business Intelligence* (ISCBI), pp. 1-5, 2015.

[17] Https://Www.Unb.Ca/Cic/Datasets/Pdfmal-2022.Html

[18] Https://Github.Com/Srndic/Mimicus/Tree/Master/Data

[19] Damaševi˘cius, R.; Ven˘ckauskas, A.; Toldinas, J.; Grigaliunas, Š, " Ensemble-Based ¯ Classification Using Neural Networks and Machine Learning Models for Windows PE Malware Detection. Electronics," vol.10, pp.485, 2021. Https:// Doi.Org/10.3390/Electronics10040485

[20] Modalavalasa Hari Krishna, Dr.Makkena Madhavi Latha *"Intelligent Parameter Tuning Using Segmented Recursive Reinforcement Learning Algorithm"* International Journal of Engineering Trends and Technology 68.6(2020):1-8.