Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.029946
Article

Tech Science Press

Optimal Deep Belief Network Enabled Malware Detection and Classification Model

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

[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
*Corresponding Author: P. Pandi Chandran. Email: pandi.chandran@gmail.com

Received: 15 March 2022; Accepted: 20 April 2022

Abstract: Cybercrime has increased considerably in recent times by creating new methods of stealing, changing, and destroying data in daily lives. Portable Document Format (PDF) has been traditionally utilized as a popular way of spreading malware. The recent advances of machine learning (ML) and deep learning (DL) models are utilized to detect and classify malware. With this motivation, this study focuses on the design of mayfly optimization with a deep belief network for PDF malware detection and classification (MFODBN-MDC) technique. The major intention of the MFODBN-MDC technique is for identifying and classifying the presence of malware exist in the PDFs. The proposed MFODBN-MDC method derives a new MFO algorithm for the optimal selection of feature subsets. In addition, Adamax optimizer with the DBN model is used for PDF malware detection and classification. The design of the MFO algorithm to select features and Adamax based hyperparameter tuning for PDF malware detection and classification demonstrates the novelty of the work. For demonstrating the improved outcomes of the MFODBN-MDC model, a wide range of simulations are executed, and the results are assessed in various aspects. The comparison study highlighted the enhanced outcomes of the MFODBN-MDC model over the existing techniques with maximum precision, recall, and F1 score of 97.42%, 97.33%, and 97.33%, respectively.

Keywords: PDF malware; data classification; security; deep learning; feature selection; metaheuristics

1 Introduction

Portable Document Format (PDF) is a very popular and trusted extension, while Adobe Reader is the most commonly used program for opening this type of file. This factor encourages attackers to seek and research for vulnerability and new ways of making exploits that implement random code when opened with this software. PDF document is more commonly utilized to launch attacks by cybercriminal [1]. A PDF document with exciting topics is transferred to the target, and once the document is opened, specific vulnerability in the software configuration or implementation is exploited for launching the next level of attacks [2]. For instance, direct implementation of native executable (when code was embedded in the PDF document itself), injection of code into an operational process or even downloading binary from the internet and later executing them [3,4].

The malware detection method is categorized into signature and behavior methods [5]. Now, signature-based malware detector effectively works with formerly known malware that has been detected previously by anti-malware vendors. To address this challenge, utilize machine learning (ML) techniques and heuristic analysis that provide high recognition performance [6]. Based on available data, the conventional method in the malware detection field depended on signature analysis [7], which is unacceptable to detect unknown computer viruses. To sustain the appropriate security level, users were forced to timely and constantly upgrade antivirus databases.

The ML technique for malware classification has utilized a wide range of information for learning discriminative functions that can distinguish benign and malicious software. Few common data sources [8] have been studied, including entropy measures on the binary, dynamic system call traces disassembled files, binary files, control flow graphs, and dynamic instruction traces. In the last few years, several attempts have been made to develop a classifier with the malware feature. Data mining and ML methods are utilized for developing smart malware classification and detection techniques [9]. The Deep neural network (DNN) has attained considerable achievement in various applications, particularly in computer vision. Even though the deep learning (DL) model is effective, they have some limitations in real-time detection tasks, particularly in security domain [10]. With the flow of zero-day and unlabeled malware, the recognition accuracy using DL is also lower. This deep model requires a high computational overhead and is very intricate. They also need a considerable amount of hyperparameters, and improved performance can be accomplished by tuning them properly.

This study presents a mayfly optimization with a deep belief network for PDF malware detection and classification (MFODBN-MDC). The proposed MFODBN-MDC model primarily undergoes two stages of pre-processing, namely categorical encoding and null value removal. Moreover, the MFODBN-MDC technique derives an MFO algorithm for optimal selection of feature subsets. Furthermore, Adamax optimizer with the DBN model is used for PDF malware detection and classification. At last, the hyperparameter tuning of the DBN model takes place using the Adamax optimizer. For exhibiting the better performance of the MFODBN-MDC model, a wide range of simulations were executed and the results were evaluated under numerous aspects.

The rest of the paper is organized as follows. Section 2 offers a detailed literature review and Section 3 discusses the proposed model. Then, Section 4 provides experimental validation and Section 5 draws the conclusions.

2 Literature Review

Corum et al. [11] introduced a learning-based model for identifying PDF malware with processing and image processing methods. The PDF file is initially transformed into grayscale image through the image visualization technique. Next, the image feature represents the visual features of malware and benign PDF files are removed. Lastly, a learning algorithm is employed for creating the classification method to categorize a PDF file as malevolent or benign. Sethi et al. [12] proposed an ML based malware analysis method for accurate and efficient malware classification and detection. Furthermore, we proposed feature selection and extraction modules that extract features from the report and select the essential feature to ensure higher accuracy at a minimal computational cost. We use a distinct ML method for fine-grained classification and accurate detection.

The researchers in [13] proposed the trusted architecture to identify unknown malware in Linux virtual machine (VM) cloud-environment. The presented method obtains volatile memory dump from the examined

VM by enquiring about the hypervisor in a reliable way and overpowering malware capability for evading detection and the security mechanism. We use the ML algorithm to leverage informative traces (171 features) from distinct portions of the VM volatile memory. Li et al. [14] developed an evasion mechanism-based feature-vector generative adversarial network (fvGAN) for attacking a learning enabled malware classification. The proposed method was commonly employed in real-time fake image generation. Damaševičius et al. [15] proposed an ensemble classifier-based method for detecting malware. Initially, it is implemented by a convolution neural network (CNN) and stacked ensemble of dense (FC), then it is implemented by a meta-learner. For a meta learner, we compare and explore fourteen classifiers.

In [16], a new malware detection scheme based on a two-phase artificial neural network (ANN) is presented. The presented method is tested on the 'Malimg' dataset comprising of visual depiction of malware family. Here, some significant image features are extracted. According to this feature, the ANN was trained. Next, the ANN is utilized for detecting and classifying other data samples. Shhadat et al. [17] examined the ML algorithm utilized in unknown malware detection. The study proposes a feature set using RF to minimize the amount of features. Various ML methods are employed on a standard dataset in this experiment. Roy et al. [18] aim is to develop a DL-based detector DeepRan for ransomware earlier classification and recognition. The presented method employs an attention based bidirectional long short term memory (Bi-LSTM) with fully connected (FC) layer for modelling normalcy of host in an operating enterprise scheme and detecting anomalous activities from a massive amount of ambient host logging information gathered from bare metal server. The researchers in [19] presented Deep-Hook, a trusted architecture for detecting unknown malware in Linux-based cloud environment. The memory dump is converted as to visual image that is investigated by a CNN based classification.

3 The Proposed Model

In this study, a new MFODBN-MDC model has been developed for the identification and classification of PDF malware. The proposed MFODBN-MDC model involves three stages of operations such as pre-processing, MFO based feature subset selection, DBN classification, and Adamax hyperparameter optimization. Fig. 1 illustrates the overall process of the MFODBN-MDC technique.

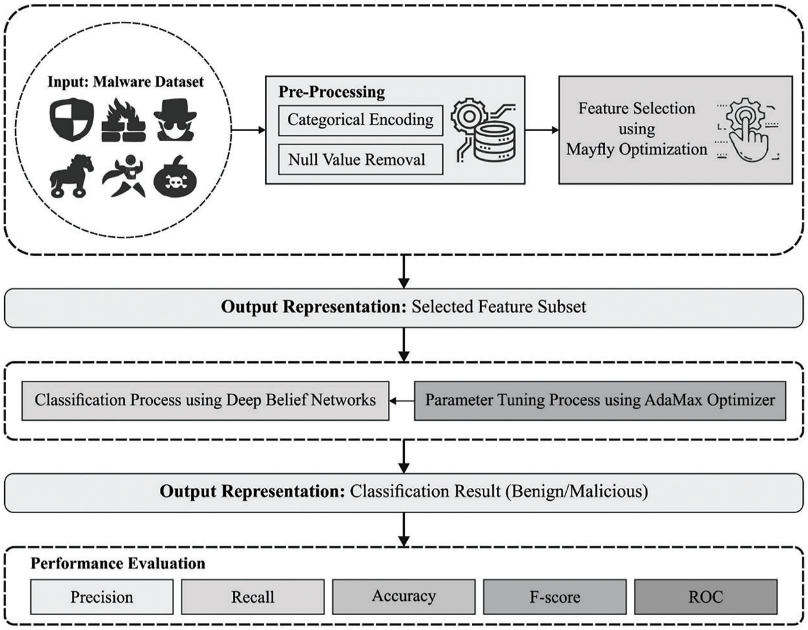

Input: Malware Dataset
Pre-Processing
Categorical Encoding
Null Value Removal
Feature Selection using Mayfly Optimization
Output Representation: Selected Feature Subset
Classification Process using Deep Belief Networks
Parameter Tuning Process using AdaMax Optimizer
Output Representation: Classification Result (Benign/Malicious)
Performance Evaluation
Precision
Recall
Accuracy
F-score
ROC

Figure 1: Overall process of MFODBN-MDC technique

3.1 Data Pre-processing

At the initial stage, the input data is pre-processed in two stages of operations such as categorical encoding and null value removal. Firstly, the categorical values are encoded into numerical values. Secondly, the null values that exist in the dataset are removed.

3.2 Design of MFO Based Feature Selection Approach

Next to data pre-processing, the MFO algorithm is utilized for the effective choice of the features involved in it [19]. MFO algorithm was proposed by imitating the group behavior of MF, especially the mating behavior. Initially, the mayfly (MF) is classified into male and female populations. In other words, each MF is arbitrarily scattered in a d-dimension space, and it can be taken into account as candidate solutions using the expression of $= (\xi_1,\ \xi_2,\ \cdots,\ \xi_d)$. Next, the velocity vector represents the modified in location is determined by $y = (\varpi_1,\ \varpi_2,\ \cdots,\ \varpi_d)$.

Movement of male MF: ξ_I^t denotes the location of $i-th$ male MF at time t, and ϖ_{Imale}^{t+1} denotes the velocity that is added to ξ_I^t for changing the location of $i-th$ individuals. The $t+1$ location of the male MF $\xi(t+1)$ is formulated as follows

$$\xi_i^{t+1} = \xi_i^t + \varpi_{i,male}^{t+1} \tag{1}$$

The velocity of $i-th$ MF at $j-th$ dimension as follows:

$$\varpi_{ij,male}^{t+1} = \varpi_{ij,male}^t + a_1 exp\left(-\beta p_p^2\right) * \left(pbest_{ij} - \xi_{ij}^t\right) + a_2 exp\left(-\beta p_g^2\right) * \left(gbest_j - \xi_{ij}^t\right) \tag{2}$$

Whereas ξ_{ij}^t and $\varpi_{i,j\ male}^t$ denotes the location and velocity of i^{th} MF at j^{th} dimension, correspondingly. $a_i(i = 1, 2)$ signifies the positive attraction constant that responds to the rule of social and cognitive mechanisms. $pbest_{ij}$ and $gbest_j$ denotes the local and global optimum locations, correspondingly. β signifies a fixed visibility co-efficient that limits the visibility of individuals to other individuals. p^p and p^g denotes the cartesian distance in i^{th} MF to the local and global optimum solutions, correspondingly. To minimize problem, the local optimum *values* $pbest_{ij}$ and optimum global values $gbest_j$ is estimated by the following equation

$$pbest_i = \begin{cases} \xi_i^{t+1},\ if\ \{\phi_{1.....c}(\xi_i^{r+1})\} \prec \{\phi(pbest_i)\} \\ is\ kept\ the\ same,\ otherwise \end{cases} \tag{3}$$

$$\begin{aligned} &gbest \in \{pbest_1, pbest_2, \cdots, pbest_N, |(\phi_{1.....c}(cbest)\} \\ &= dominate\ \{\{\phi_{1.....c}(pbest_1)\}, \{\phi_{1.....c}(pbest_2)\}, \cdots\ \{\phi_{1.....c}(pbest_N)\}\} \end{aligned} \tag{4}$$

Whereas $\phi_1, \ldots, c : \mathbb{R}^n \to \mathbb{R}$ characterizes the objective function. The optimum MF in the population continually execute an up-and-down nuptial dance to guarantee the efficient process, viz., velocity of the optimum MF should always be changed as follows:

$$\varpi_{ij,\ male}^{t+1} = \varpi_{ij,\ male}^t + d * p \tag{5}$$

In which d characterizes the nuptial dance coefficient, p indicates an arbitrary value within $[-1, 1]$, and $\varpi_{ij,\ male}^t$ indicates the location of the $i-th$ male MF at the $j-th$ parameter.

Movement of female MF: Different from male MF that gathers in swarm, the female individual is towards the male individual to breed. The existing location and the velocity of $i-th$ female MF at time r are fixed to ψ_i^t and $\varpi_{i,\ female}^{t+1}$, correspondingly. Next, the $(t+1)-th$ location of the female MF is given by:

$$\psi_i^{t+1} = \psi_i^t + \varpi_{i,\ female}^{t+1} \tag{6}$$

During the optimization method of MFO algorithm, the attraction approach is determined by a deterministic system. Regarding the minimized problem, the velocity of i-th female MF at j-th parameter is estimated as follows

$$\varpi_{ij,\,female}^{t+1} = \begin{cases} \varpi_{ij}^t\, female\ + a_2 exp\left(-\beta p_{mf}^2\right) * \left(\xi_{ij}^{'} - \psi_{ij}^t\right),\ if\ \varnothing(\psi_i) > (p\left(\xi_j\right) \\ \varpi_{ij,\,female}^t\ + fl{*}p,\ if\ \varnothing(\psi_i) \le (p(\xi_i) \end{cases} \tag{7}$$

Whereas ψ_{Ij}^t and $\varpi_{ij,\,female}^t$ denotes the location and velocity of i^{th} female MF at j^{th} variable. p_{mf} means the cartesian distance from i^{th} male MF to i^{th} female MF. fl characterizes the arbitrary walk co-efficient.

Mating of MF: Two parents are carefully chosen in male as well as female populations, correspondingly. The mating rule depends on the mating of optimal male with optimal female, creating 2 offspring based on the following equations:

$$offspring\ 1 = \mu{*}male + (1-\mu){*}female \tag{8}$$

$$offspring2\ = \mu{*}female\ + (1-\mu){*}male \tag{9}$$

Here, males and females denote the male and female individuals of the preceding generation, and $\mu \in (0, 1)$ signifies an arbitrary number. The primary velocity of the individual in the present generation denotes 0. The MFO algorithm derives a fitness function using two parameters for the effective selection of features namely classification accuracy and number of chosen features. It can be derived as follows.

$$fitness\ = \omega_1 \times acc(classifier) + \omega_2 \times \left(1 - \frac{s}{p}\right) \tag{10}$$

where p signifies total number of features and s denotes the number of chosen features. Here, the value of ω_1 and ω_2 are 1 and 0.001, respectively, [17]. The $acc(classifer)$ represents the overall classifier accuracy attained by the DBN model that can be attained using Eq. (11):

$$acc(\text{classifier}) = \frac{n_c}{n_c + n_i} \times\ 100\% \tag{11}$$

where n_i and n_c represents the number of wrongly and properly classified samples respectively.

3.3 Process Involved in Optimal DBN Based Classification

For the identification and classification of PDF malware, the DBN model [20] is applied in this work. DBN is a multi-layered probabilistic model [20] that comprises multi-parameters for learning models. All the layers contain a simple undirected graph named restricted Boltzman machine (RBM). The RBM layer is of two kinds, that is visible layer and hidden layer. The hidden layer denotes the top layer, and visible layer represents the bottom layer. Fig. 2 illustrates the framework of DBN. An RBM encode the joint likelihood distribution through the energy function, where v denotes the visible data, h indicates the hidden data, w represents the weight, and $\theta = \left(w,\ b^{(v)},\ b^{(h)}\right)$. It can be expressed as follows.

$$E(v, h,\ \theta) = -\sum\nolimits_i \sum\nolimits_j w_{ij} v_i h_j - \sum\nolimits_i b_i^{(v)} v_i - \sum\nolimits_i b_j^{(h)} h_j \tag{12}$$

$$p(v,\ h|\theta) = \frac{\exp(-E(v, h : \theta)}{\Sigma_{v'} \Sigma_{h}, \exp(-E(v, h, \theta)}. \tag{13}$$

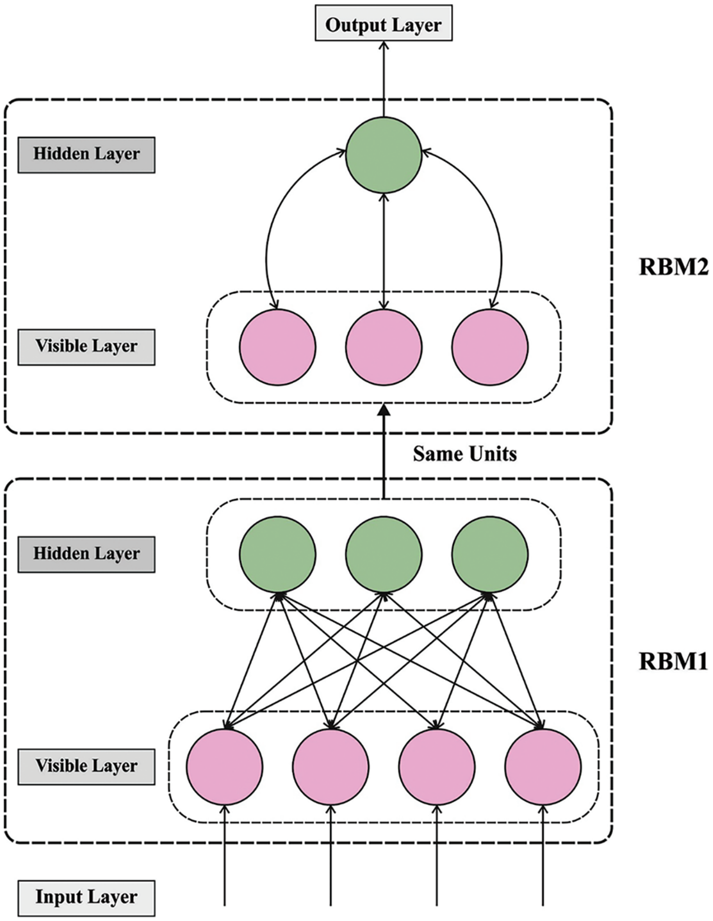


Figure 2: DBN architecture

This rule can be derived to upgrade the primary state; thus, each update gives a low energy state and eventually settles into equilibrium. Now, $\sigma(x) = 1/(1 + exp(-x))$, whereas the sigmoid function is detected as follows:

$$p(v_i = 1|h,\ \theta) = \sigma\left(\sum\nolimits_j w_{ij}h_j + b_i^{(v)}\right) \tag{14}$$

$$p(h_i = 1|v,\ \theta) = \sigma\left(\sum\nolimits_i w_{ij}v_i + b_j^{(h)}\right) \tag{15}$$

The visible layer is offered with the input data for training the RBM. Now, the learning is to adopt the variable θ thus the likelihood distribution becomes maximally analogous to the true value implies that it maximizes the log-probability of observed data. The contrastive divergence (CD) samples the value for each hidden layer and the present input gives a whole sample $(v_{data},\ h_{data})$. It can be attained the sample from the model as $(v_{model},\ h_{model})$. The weight is upgraded as follows

$$\Delta w_{ij} = \eta\left(v_{i,\ data}h_{j,data} - v_{i,model}h_{j,\ model}\right). \tag{16}$$

In order to effectually modify the hyperparameter values of the DBN model, the Adamax optimizer is utilized [21]. Adamax is a variant of Adam dependent upon the infinity norm. Here, the update rules for separate weight measure the gradient inversely proportionate to *a* (scaled) L^2 norm of the present and previous gradients. Then, generalize the L^2 norm-based updating rules to L^p norm-based updating rules. This variant becomes arithmetically unstable for larger p. But, in the special case [22], we consider $p \rightarrow \infty$, which emerges as a stable and simple approach. Update biased first moment estimation:

$$m_t \leftarrow \beta_1 . m_{t-1} + (1 - \beta_1) \tag{17}$$

Update the exponentially weighted infinity norm:

$$u_t \leftarrow \max\ (\beta_2 . u_{t-1},\ |g_t|) \tag{18}$$

Update parameter:

$$\theta_t \leftarrow \theta_{t-1} - \left(\frac{\alpha}{1 - \beta_1^t}\right) . m_t / u_t \tag{19}$$

The default setting for the tested ML problem is $\alpha = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

4 Performance Validation

In this section, the experimental validation of the MFODBN-MDC model is carried out using the CIC Evasive-PDFMal2022 (https://www.unb.ca/cic/datasets/pdfmal-2022.html) and Contagio (https://github.com/srndic/mimicus/tree/master/data) datasets. The first CIC Evasive-PDFMal2022 dataset includes 31 features with two classes, namely Benign and Malicious. The proposed technique has chosen a set of 15 features. Similarly, the Contagio dataset includes 136 features, and the proposed model has selected a collection of 40 features.

Fig. 3 showcases the convergence analysis of the MFODBN-MDC model on the two datasets, namely CIC Evasive-PDFMal2022 and Contagio dataset. The figure shows that the MFODBN-MDC model has shown effective performance with an optimal convergence rate.

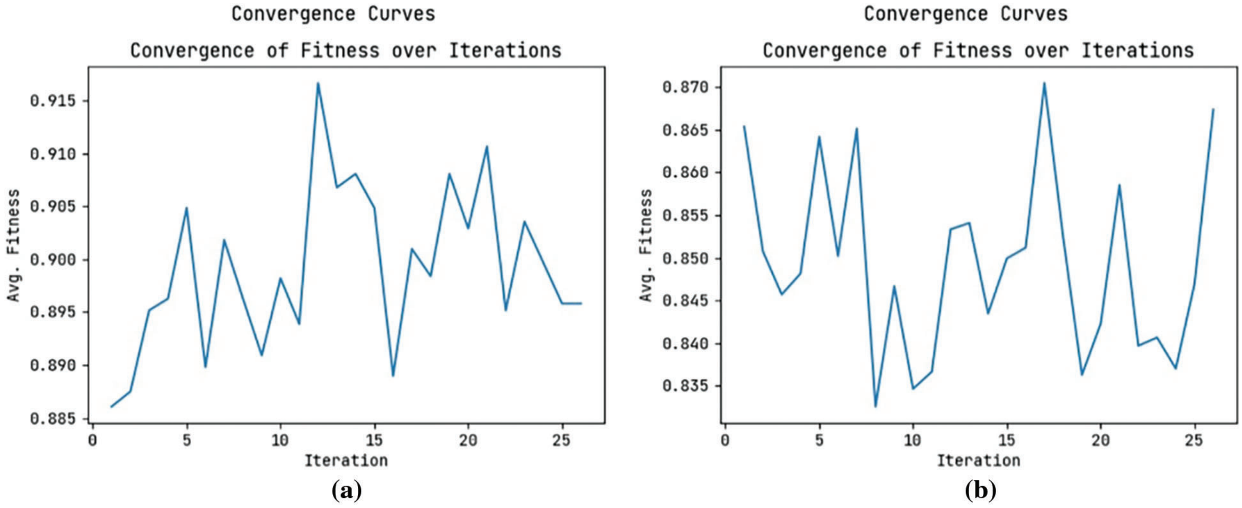


Figure 3: Convergence analysis (a) CIC Evasive-PDFMal2022, (b) Contagio dataset

The selected features on the CIC Evasive-PDFMal2022 dataset are metadata size, pages, xref Length, title characters, isEncrypted, embedded files, images, text, header, obj, endobj, stream, endstream, xref, and trailer. Besides, the chosen features from the Contagio Dataset are author_dot, author_lc, author_len, author_mismatch, author_num, author_oth, author_uc, box_nonother_types, box_other_only, company_mismatch, count_acroform, count_acroform_obs, count_action, count_action_obs, count_box_a4, count_box_legal, count_box_letter, count_box_other, count_box_overlap, count_endobj, count_endstream, count_eof, count_font, count_font_obs, count_image_large, count_image_med, count_image_small, count_image_total, count_image_xlarge, count_image_xsmall, count_javascript, count_javascript_obs, count_js, count_js_obs, count_obj, count_objstm, count_objstm_obs, count_page, count_page_obs, count_startxref.

Fig. 4 demonstrates a brief result analysis of the MFODBN-MDC model on the test CIC-Evasive-PDFMal2022 dataset. Fig. 4a depicts the confusion matrix offered by the DBN model. The figure denoted that the DBN model has identified 4358 instances under benign and 4601 instances under Malicious. Similarly, Fig. 4b indicated that MFODBN-MDC model has recognized 4237 samples under benign class and 5343 samples under malicious class. Next, Figs. 4c and 4d demonstrate the precision recall analysis of the DBN and MFODBN-MDC model. The figures exposed that the MFODBN-MDC technique has obtained maximum performance over the DBN model. Finally, Figs. 4e and 4f illustrate the receiver operating characteristic (ROC) investigation of the DBN and MFODBN-MDC model. The figure indicated that the MFODBN-MDC model has obtained a higher ROC of 0.9891 and 0.9891 under benign and malicious classes respectively.

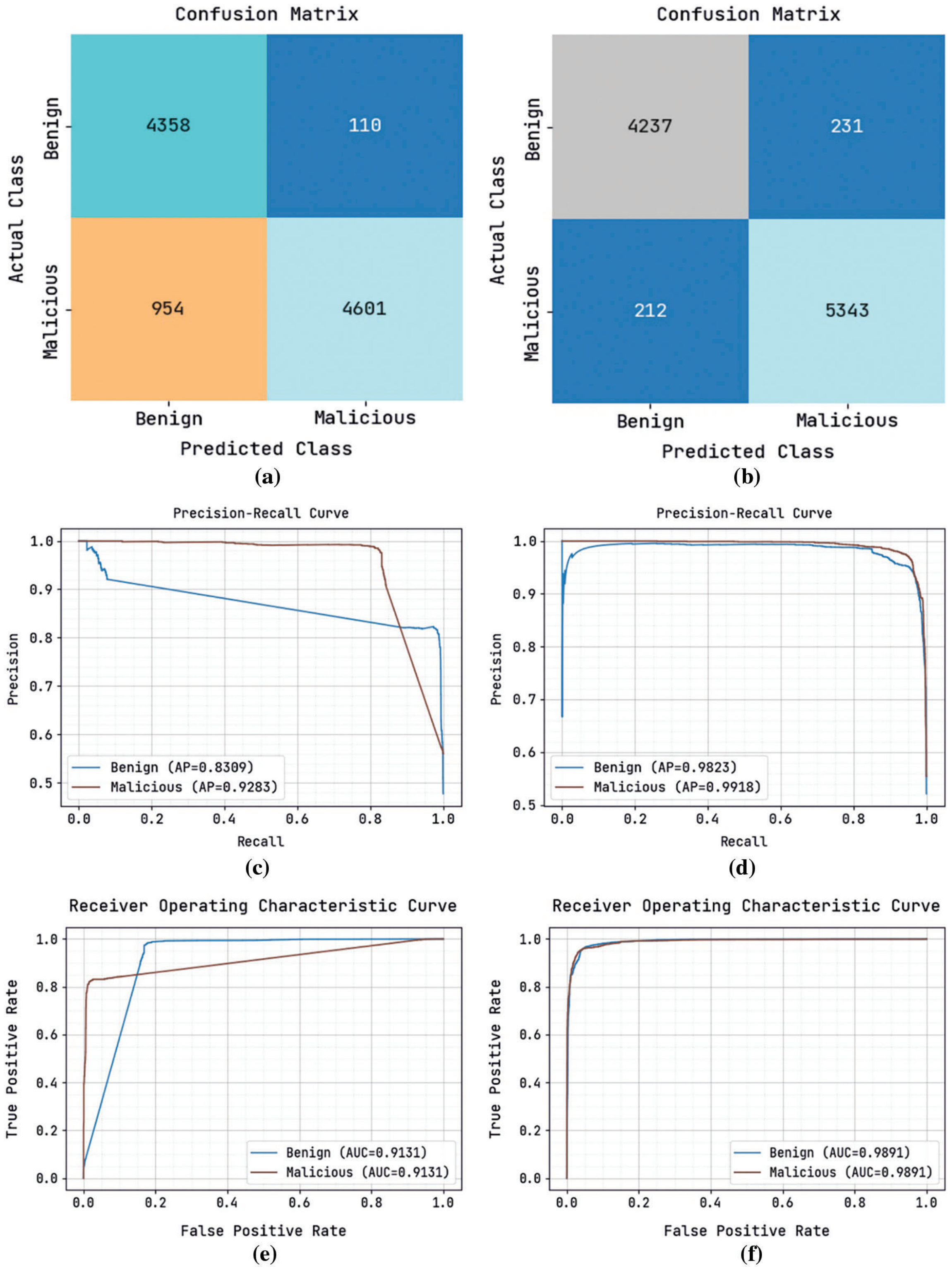
Fig. 5 offers the accuracy and loss graph analysis of the DBN and MFODBN-MDC methods on CIC-Evasive-PDFMal2022 dataset. The outcomes show that the accuracy value tends to increase and loss value tends to decrease with an increase in epoch count. Also, It is observed that the training loss is low, and validation accuracy is high on the CIC-Evasive-PDFMal2022 dataset.

Fig. 6 depicts a brief result analysis of the MFODBN-MDC system on the test Contagio dataset. Fig. 6a demonstrates the confusion matrix presented by the DBN method. The figure denotes that the DBN method has recognized 4880 instances under benign and 4512 instances under Malicious. Likewise, Fig. 6b shown that the MFODBN-MDC method has recognized 4975 samples under benign class and 4757 samples under malicious class. Then, Figs. 6c and 6d prove the precision recall analysis of DBN and MFODBN-MDC method. The figure reports that the MFODBN-MDC approach has gained maximal performance over the DBN system. Lastly, Figs. 6e and 6f show the ROC investigation of the DBN and MFODBN-MDC method. The figure shows that the MFODBN-MDC technique has correspondingly attained high ROC of 0.993 and 0.993 under benign and malicious classes.

Fig. 7 offers the accuracy and loss graph analysis of the DBN and MFODBN-MDC methods on Contagio dataset. The results exposed that the accuracy value tends to increase and loss value tends to decrease with an increase in epoch count. Also, it is observed that the training loss is low, and validation accuracy is high on Contagio dataset.

Tab. 1 provides detailed classification outcomes of the DBN and MFODBN-MDC models on two datasets. Fig. 8 reports the result analysis of the MFODBN-MDC model and DBN model on the CIC Evasive-PDFMal2022 dataset. The results indicated that the DBN model has obtained $accc_y$, $prec_n$, $reca_l$, $F1_{score}$, and Area Under the Curve (AUC) of 89.38%, 89.85%, 90.18%, 89.38%, and 91.31% respectively. However, the MFODBN-MDC model has offered enhanced performance with $accc_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC of 95.58%, 95.55%, 95.51%, 95.53%, and 98.91% respectively.

Fig. 9 shows the result analysis of the MFODBN-MDC and DBN models on Contagio dataset. The results showed that the DBN system has gained $accc_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC of 93.93%, 94.17%, 93.93%, 93.92%, and 92.40% correspondingly. But the MFODBN-MDC technique has presented enhanced performance with $accc_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC of 97.33%, 97.42%, 97.33%, 97.33%, and 99.30% correspondingly.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 4: CIC Evasive-PDFMal2022 dataset (a) CM DBN, (b) CM MFODBN-MDC, (c) PCR DBN, (d) PCR MFODBN-MDC, (e) ROC DBN, (f) ROC MFODBN-MDC

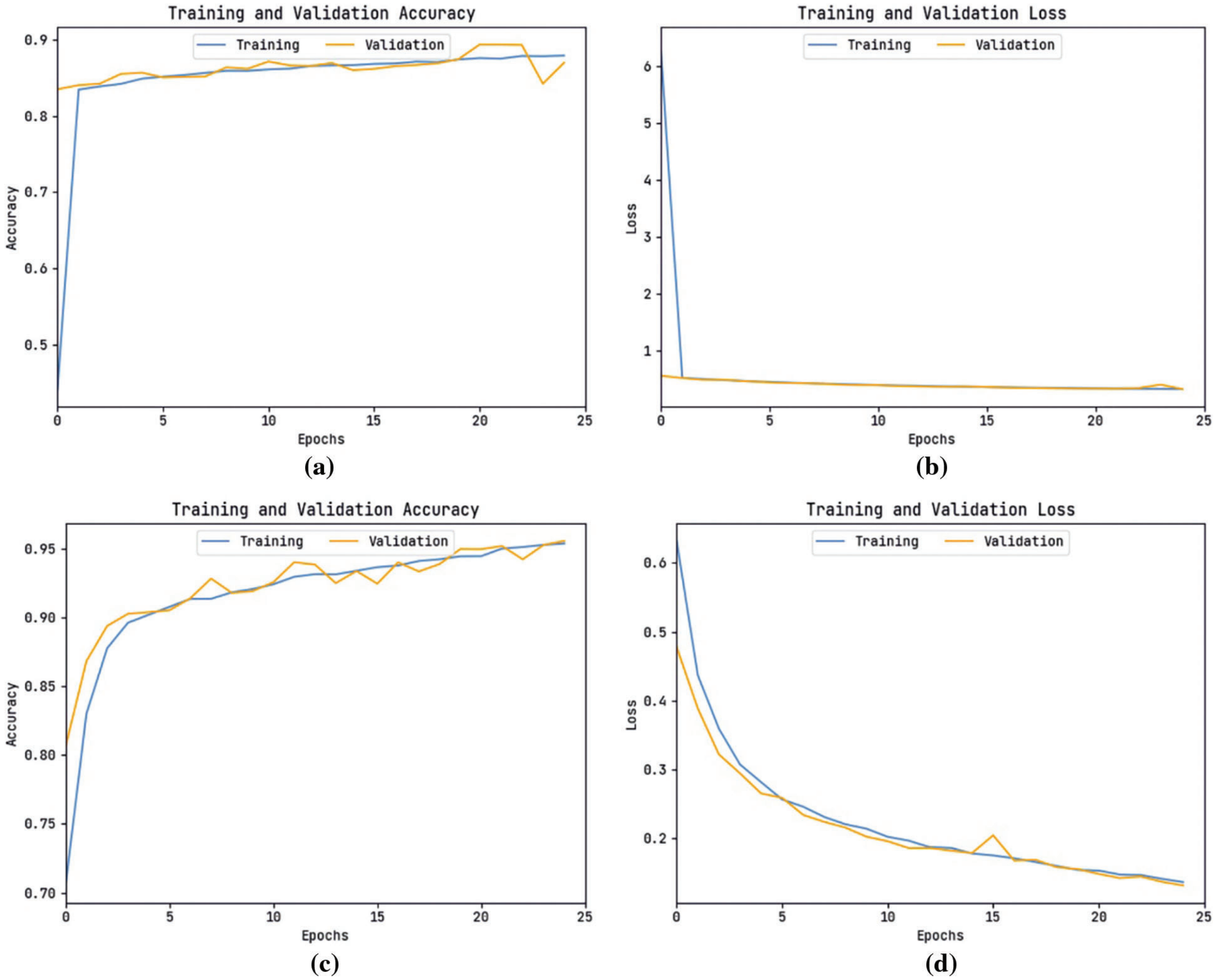
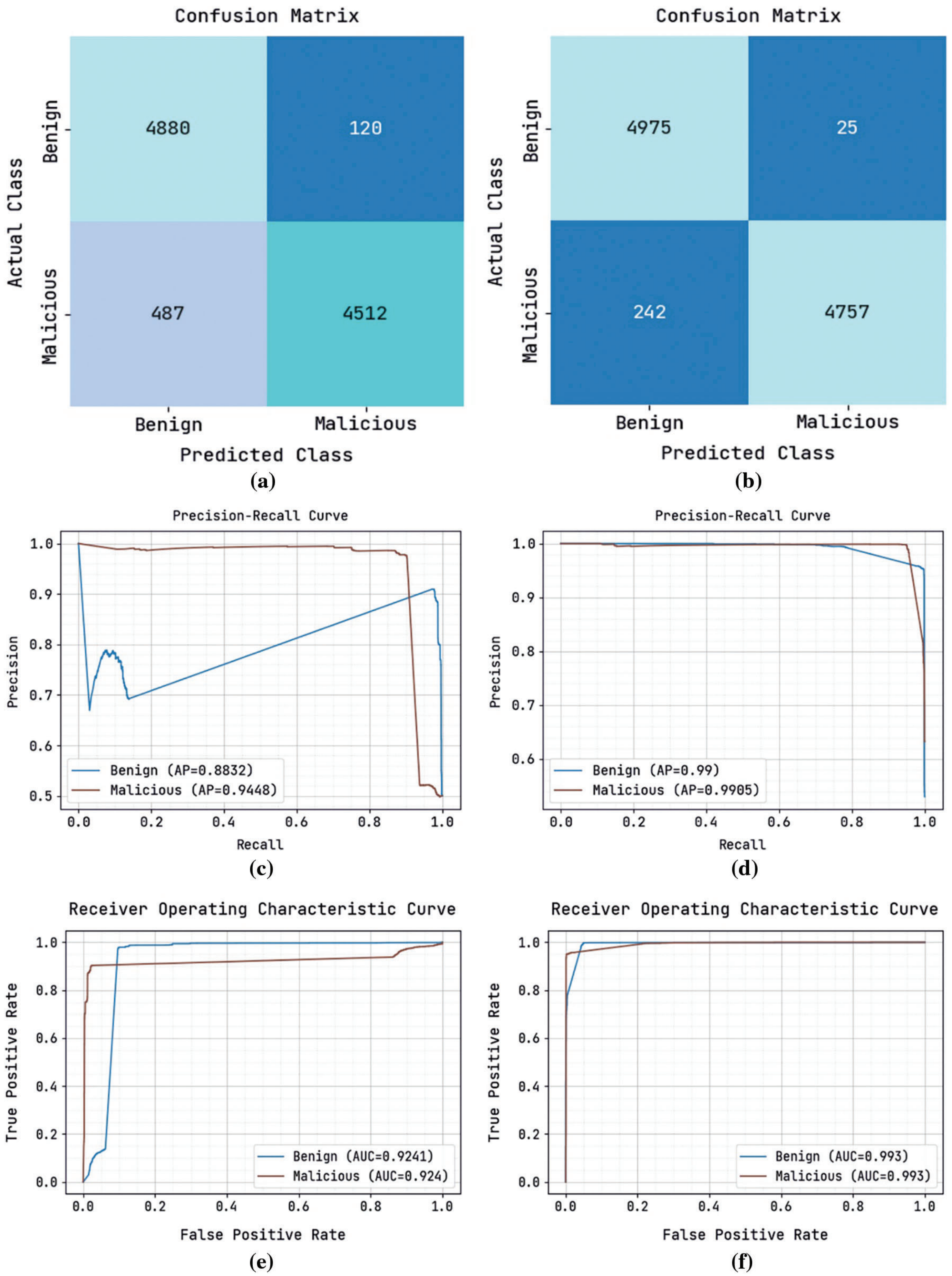


Figure 5: CIC Evasive-PDFMal2022 dataset (a) Accuracy of DBN, (b) Loss of DBN, (c) Accuracy of MFODBN-MDC, (d) Loss of MFODBN-MDC

To ensure the MFODBN-MDC model's better performance, detailed comparative analyses with existing models are carried out in Tab. 2. For comparison study, decision tree (DT), random forest (RF), AdaBoost, Logistic regression (LR), ridge regression (RR), and SGDC models. The experimental outcomes indicated that the MFODBN-MDC technique has obtained maximal performance over the other methods under several measures.

Fig. 10 demonstrates a comparative $accu_y$ and AUC examination of the MFODBN-MDC model with existing models. The figure reported that the DT, RF, and RR models have showcased poor performance with least values of $accu_y$ and AUC. Followed by, the AdaBoost and SGDC models have reported slightly enhanced values of $accu_y$ and AUC. In line with, the LR model has accomplished considerately $accu_y$ and AUC values of 96.33% and 96.20%. However, the MFODBN-MDC model has resulted in maximum $accu_y$ and AUC of 97.33% and 99.30%.

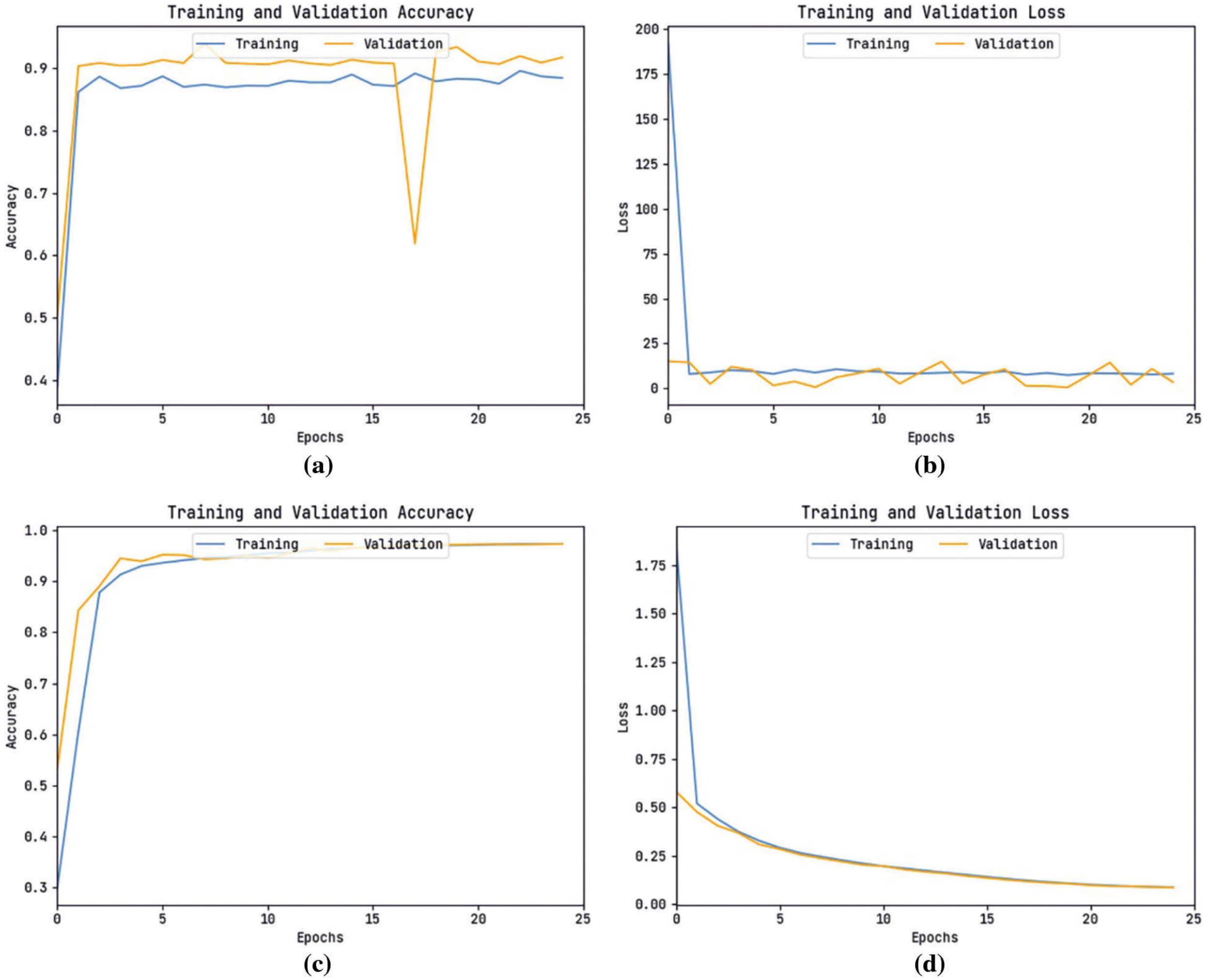


(a) (b)

(c) (d)

(e) (f)

Figure 6: Contagio dataset (a) CM DBN, (b) CM MFODBN-MDC, (c) PCR DBN, (d) PCR MFODBN-MDC, (e) ROC DBN, (f) ROC MFODBN-MDC



(a)

(b)

(c)

(d)

Figure 7: Contagio dataset (a) Accuracy of DBN, (b) Loss of DBN, (c) Accuracy of MFODBN-MDC, (d) Loss of MFODBN-MDC

Table 1: Result analysis of proposed method under two datasets with different measures

Measures	CIC Evasive-PDFMal2022		Contagio Dataset	
	DBN Model	MFODBN-MDC	DBN Model	MFODBN-MDC
Accuracy	89.38	95.58	93.93	97.33
Precision	89.85	95.55	94.17	97.42
Recall	90.18	95.51	93.93	97.33
F1-Score	89.38	95.53	93.92	97.33
AUC-Score	91.31	98.91	92.40	99.30

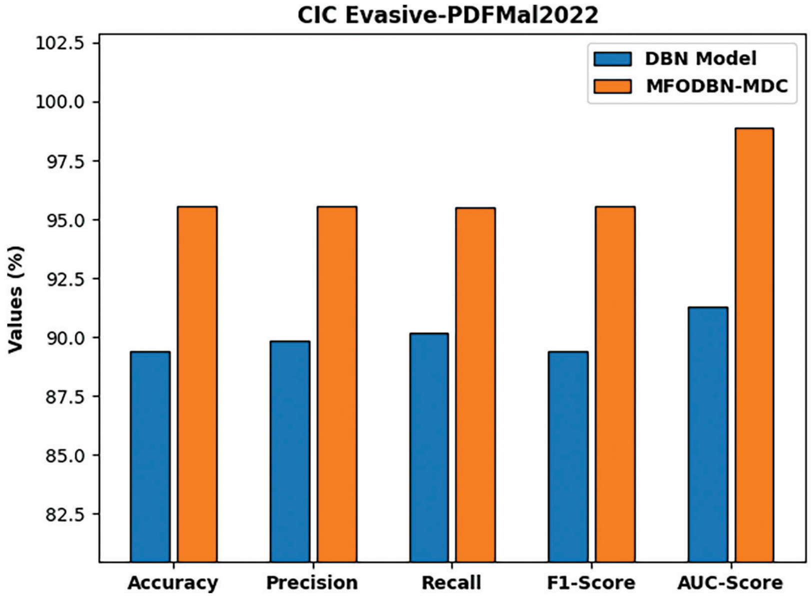


Figure 8: Result analysis of proposed method on CIC Evasive-PDFMal2022 dataset

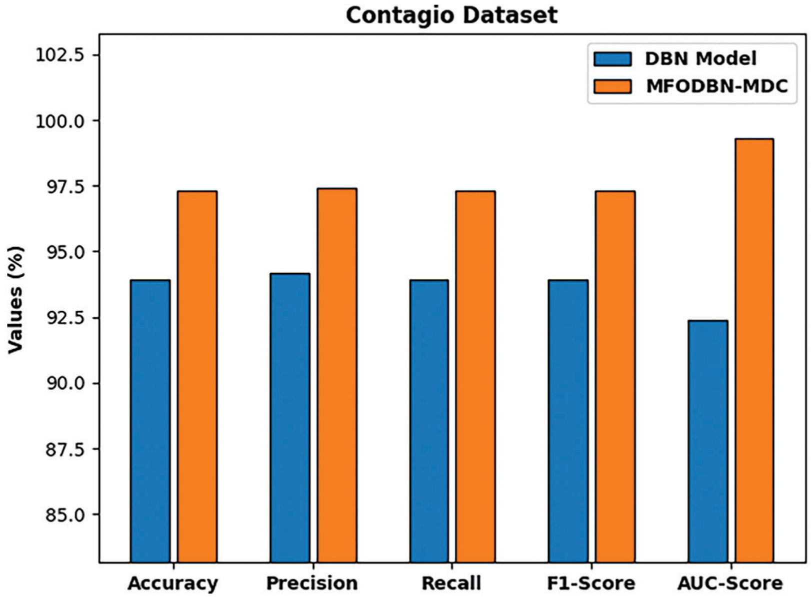


Figure 9: Result analysis of proposed method on Contagio dataset

Table 2: Comparative analysis of MFODBN-MDC technique with existing algorithms

Methods	Accuracy	Precision	Recall	F1-Score	AUC
DT Model	93.47	92.92	93.52	93.29	93.46
RF Model	93.19	93.54	93.18	93.41	92.82
AdaBoost	95.82	96.65	95.84	96.17	95.71
LR	96.33	95.73	96.38	96.39	96.20
RR	93.50	93.20	93.31	93.45	93.71
SGDC Model	95.68	95.57	95.56	95.81	95.76
MFODBN-MDC	97.33	97.42	97.33	97.33	99.30

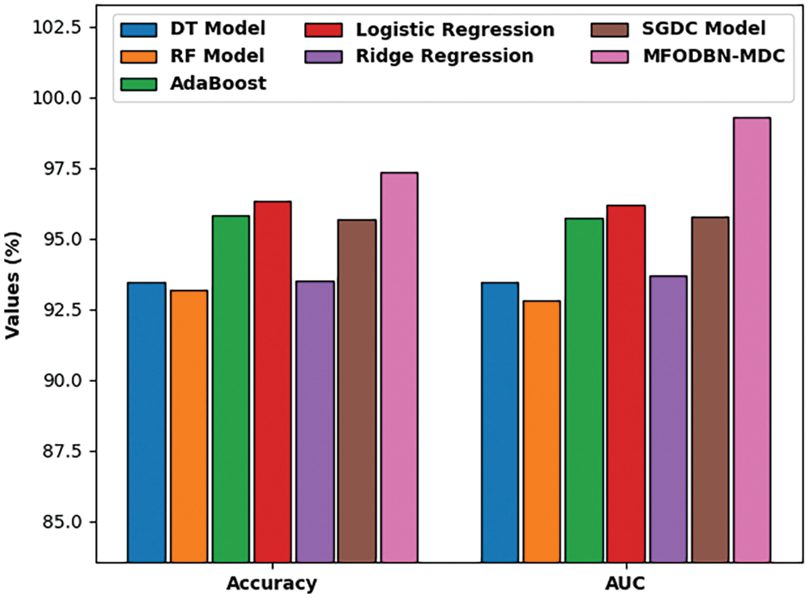


Figure 10: Acc_y and AUC analysis of MFODBN-MDC technique with existing algorithms

Fig. 11 proves a comparative $prec_n$, $reca_l$, and $F1_{score}$ examination of the MFODBN-MDC model with existing model. The figure reports that the DT, RF, and RR methods have shown poor performance with minimum values of $prec_n$, $reca_l$, and $F1_{score}$. Next, the AdaBoost and SGDC approaches have reported somewhat improved values of $prec_n$, $reca_l$, and $F1_{score}$. In line with, the LR method has gained considerately $prec_n$, $reca_l$, and $F1_{score}$ values of 95.73%, 96.38%, and 96.39%. However, the MFODBN-MDC method has resulted in maxima $prec_n$, $reca_l$, and $F1_{score}$ of 97.42%, 97.33%, and 97.33%. After examining the abovementioned tables and figures, it is clear that the MFODBN-MDC model has accomplished maximum PDF malware detection and classification outcomes.

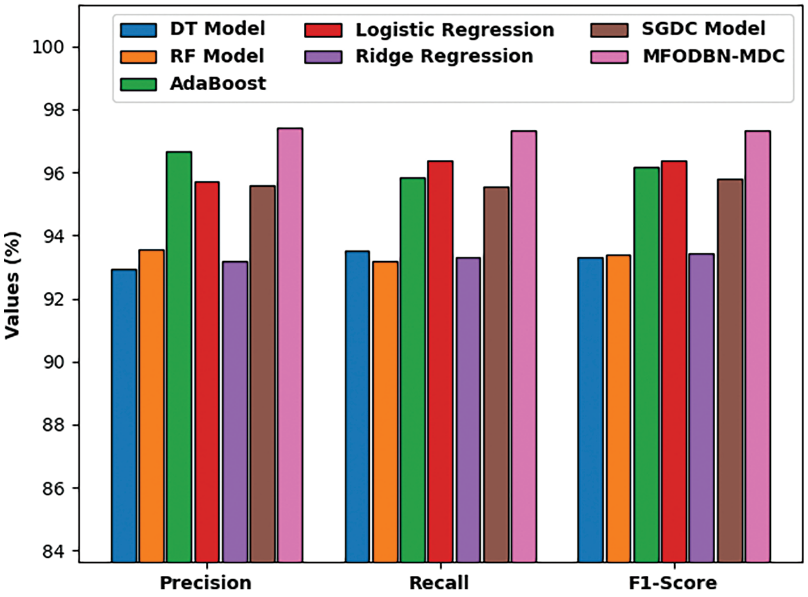


Figure 11: Comparative analysis of MFODBN-MDC technique with existing algorithms

5 Conclusion

In this study, a MFODBN-MDC technique was established for the identification and classification of PDF malware. The proposed MFODBN-MDC technique contains three stages of operations such as pre-processing, MFO based feature subset selection, DBN classification, and Adamax hyperparameter optimization. For exhibiting the better performance of the MFODBN-MDC model, a wide range of simulations are executed, and the outcomes are evaluated under various aspects. The extensive comparative analysis reported the enhanced outcomes of the MFODBN-MDC model over the recent approaches. Therefore, the MFODBN-MDC model can be utilized as a proficient tool for PDF malware detection and classification. In the future, the classification results of the MFODBN-MDC model can be improved by using outlier detection and feature reduction approaches.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Maiorca, B. Biggio and G. Giacinto, "Towards adversarial malware detection: Lessons learned from PDF-based attacks," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–36, 2020.

[2] K. Shaukat, F. Iqbal, T. M. Alam, G. K. Aujla, L. Devnath *et al.,* "The impact of artificial intelligence and robotics on the future employment opportunities," *Trends in Computer Science and Information Technology*, vol. 5, no. 1, pp. 50–54, 2020.

[3] A. Nasir, K. Shaukat, K. I. Khan, I. A. Hameed, T. M. Alam *et al.,* "What is core and what future holds for blockchain technologies and cryptocurrencies: A bibliometric analysis," *IEEE Access*, vol. 9, pp. 989–1004, 2021.

[4] R. Bhargava, "Cyber crime and cyber security in Madhya Pradesh," *National Journal of Environment and Scientific Research*, vol. 2, no. 8, pp. 53, 2021.

[5] A. Tekerek and M. M. Yapici, "A novel malware classification and augmentation model based on convolutional neural network," *Computers & Security*, vol. 112, no. 3, pp. 102515, 2022.

[6] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed and M. Xu, “A survey on machine learning techniques for cyber security in the last decade,” *IEEE Access*, vol. 8, pp. 222310–222354, 2020.

[7] T. M. Alam, K. Shaukat, I. A. Hameed, W. A. Khan, M. U. Sarwar *et al.,* “A novel framework for prognostic factors identification of malignant mesothelioma through association rule mining,” *Biomedical Signal Processing and Control*, vol. 68, pp. 102726, 2021.

[8] Y. Li, X. Wang, Z. Shi, R. Zhang, J. Xue *et al.,* “Boosting training for PDF malware classifier via active learning,” *International Journal of Intelligent Systems*, vol. 37, no. 4, pp. 2803–2821, 2022.

[9] H. Bae, Y. Lee, Y. Kim, U. Hwang, S. Yoon *et al.,* “Learn2Evade: Learning-based generative model for evading PDF malware classifiers,” *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 4, pp. 299–313, 2021.

[10] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, A. S. Ariffin *et al.,* “Finger vein biometrics: Taxonomy analysis, open challenges, future directions and recommended solution for decentralised network architectures,” *IEEE Access*, vol. 8, pp. 9821–9845, 2020.

[11] A. Corum, D. Jenkins and J. Zheng, “Robust PDF malware detection with image visualization and processing techniques,” in *2019 2nd Int. Conf. on Data Intelligence and Security (ICDIS)*, South Padre Island, TX, USA, pp. 108–114, 2019.

[12] K. Sethi, R. Kumar, L. Sethi, P. Bera and P. K. Patra, “A novel machine learning based malware detection and classification framework,” in *2019 Int. Conf. on Cyber security and protection of digital services (Cyber Security)*, Oxford, United Kingdom, pp. 1–4, 2019.

[13] T. Panker and N. Nissim, “Leveraging malicious behavior traces from volatile memory using machine learning methods for trusted unknown malware detection in Linux cloud environments,” *Knowledge Based Systems*, vol. 226, no. 4, pp. 107095, 2021.

[14] Y. Li, Y. Wang, Y. Wang, L. Ke and Y. Tan, “A feature-vector generative adversarial network for evading PDF malware classifiers,” *Information Sciences*, vol. 523, no. 9, pp. 38–48, 2021.

[15] R. Damaševičius, A. Venčkauskas, J. Toldinas and Š. Grigaliūnas, “Ensemble-based classification using neural networks and machine learning models for Windows PE malware detection,” *Electronics*, vol. 10, no. 4, pp. 485, 2021.

[16] V. Moussas and A. Andreatos, “Malware detection based on code visualization and two-level classification,” *Information—An International Interdisciplinary Journal*, vol. 12, no. 3, pp. 118, 2021.

[17] I. Shhadat, B. Bataineh, A. Hayajneh and Z. A. Al-Sharif, “The use of machine learning techniques to advance the detection and classification of unknown Malware,” *Procedia Computer Science*, vol. 170, pp. 917–922, 2020.

[18] K. C. Roy and Q. Chen, “DeepRan: Attention-based BiLSTM and CRF for ransomware early detection and classification,” *Information Systems Frontiers*, vol. 23, no. 2, pp. 299–315, 2021.

[19] T. Landman and N. Nissim, “Deep-Hook: A trusted deep learning-based framework for unknown malware detection and classification in Linux Cloud environments,” *Neural Networks*, vol. 144, no. 1, pp. 648–685, 2021.

[20] Z. Liu, P. Jiang, J. Wang and L. Zhang, “Ensemble forecasting system for short-term wind speed forecasting based on optimal sub-model selection and multi-objective version of mayfly optimization algorithm,” *Expert Systems with Applications*, vol. 177, no. 3, pp. 114974, 2021.

[21] M. Javeed, M. Gochoo, A. Jalal and K. Kim, “HF-SPHR: Hybrid features for sustainable physical healthcare pattern recognition using deep belief networks,” *Sustainability*, vol. 13, no. 4, pp. 1699, 2021.

[22] T. Le, J. Kim and H. Kim, “An effective intrusion detection classifier using long short-term memory with gradient descent optimization,” in *Int. Conf. on Platform Technology and Service (PlatCon)*, Busan, South Korea, pp. 1–6, 2017.