Adaptive Cyclic Reconstruction Regularized Bidirectional Feed Forward Network for Robust Malware Classification

Nagaraj Shet Manjappa Narahari
Department of Computer Science & Engineering
Impact College of Engineering & Applied Sciences,
Affiliated to Visvesvaraya Technological University,
Belagavi, Karnataka, India

Jagadeesha Ramegowda
Department of Computer Science & Engineering
Impact College of Engineering & Applied Sciences,
Affiliated to Visvesvaraya Technological University,
Belagavi, Karnataka, India

ABSTRACT

The high structural similarity of the variants and the existence of the subtle visual pattern in binary representations have made malware classification more difficult than ever, as the malware families evolve very quickly. Despite the promising performance of deep learning-based image malware classification systems, the majority of the currently used methods are based on fixed loss functions, which do not respond to changing learning dynamics in the course of training, resulting in unsteady feature representations and poor generalization. In order to overcome these issues, this paper presents an Adaptive Cyclic Reconstruction-Regularized Bidirectional Feed Forward neural network, A-CRCL-BFFNN, to classify malware families strongly. The framework proposed is built upon the concept of a bidirectional contrastive autoencoder, in combination with a dynamic loss regulation scheme that balances dynamically between the reconstruction loss, cyclic consistency loss and contractive regularization through training behavior. Such an adaptation provides the model with the ability to learn structural preservation during initial training stages and increasingly implement robustness and latent space smoothness with each succeeding training stage. Binaries of malware are then converted to color-mapped image representation and normalized to maximize the discriminative features learning. In-depth experiments over the Malimg and BIG2015 benchmark datasets revealed that the presented A-CRCL-BFFNN has the highest classification accuracy, recall, and F1-score, with the incidental computation overhead.

General Terms

Pattern Recognition, Security, Algorithms et. al.

Keywords

Malware Classification; Adaptive Loss Regulation; Bidirectional Contrastive Autoencoder; Cyclic Reconstruction; Image-Based Malware Analysis.

1. INTRODUCTION

The uncontrolled growth in the number of malware and its ongoing advancement with regards to the malware structure, malware behavior, and malware obfuscation is a major threat to the current computing infrastructures [1]. The modern malware families demonstrate a high intra-family similarity and a slight inter-family differentiation, thus, making it that more challenging to accurately classify the malware families. Older signature-based methods cannot be relied upon anymore as they are unable to extend their coverage to unknown or polymorphous malware counterparts [2]. As a result, machine learning and deep learning-based data-driven methods of malware classification have attracted considerable attention because of their capability to acquire discriminative patterns automatically with regard to the raw or transformed malware representation. The classification of malware through images has become an attractive direction in which binary files are translated into images that reflect structural properties of malicious code. They allow the use of spatial correlations and texture-like patterns by deep neural networks that malware binaries contain [3]. Nevertheless, relying on significant advances, the current deep learning models are very likely to lose their strength in case of the presence of extremely similar malware variants. Such a limitation is mostly explained by the unstable representations of features, overfitting due to the overpowering loss components, and lack of control over smoothness by the latent space in the training process.

Recent research has also shown that autoencoder-based architecture combined with a feed forward neural network may promote feature extraction by retaining the information in the form of structure through the reconstruction process [4]. Specifically, bidirectional contrastive autoencoders have demonstrated its effectiveness in both forward and backward dependency of malware data, thus enhancing separability of classes. Also, there has been the use of cyclic reconstruction and contractive regularization to enforce consistency of representation and resistance to small perturbations [5]. Although these methods provide substantial classification accuracy, most of the commonly existing methods use fixed loss weighting schemes, i.e. the weight of reconstruction loss, cyclic consistency loss, and regularization loss is fixed during training. The malware classification tasks, however, cannot be modelled with the help of the static loss formulations. It is essential to maintain coarse structural information at the early training stages where in the later stages, more emphasis should be put on robustness, smoothness, and refining of discriminative features. The fixed loss weight cannot support these changing learning needs, and can either result in either an under-regularized or overly constrained latent space. Such imbalance may restrict generalization performance especially in the cases of visual similarity malware families or unbalanced set of data [6].

To overcome these drawbacks, the A-CRCL-BFFNN is suggested in this paper to ensure effective malware classification using Adaptive Cyclic Reconstruction-Regularized Bidirectional Feed Forward Neural Network. The suggested structure adds a dynamic control of the contribution made by reconstruction loss, cyclic consistency loss, and contractive regularization, depending on the training dynamics.

The model is informed by the use of entropy-sensitive and gradient-sensitive loss weighting that encourage the model to focus on structural preservation in the initial epochs and gradually make enhanced robustness and latent stability in subsequent epochs. This is an adaptive strategy, which eliminates dominance of a single loss component and allows balanced representation learning with training phases. The suggested A-CRCL-BFFNN is tested on well-known malware databases, i.e. BIG2015 and Malimg, which comprise different malware families having different levels of similarity in terms of structure [7]. Large-scale experiments prove that adaptive loss regulation results in a higher level of classification, speedier convergence, and increase in robustness than the counterparts of the successive loss-based and traditional deep learning models do [8]. The findings confirm that small performance improvements by adaptive learning mechanisms can be converted into significant performance improvements in the large-scale malware classification problems [9].

The rest of this paper is structured in the following way. Section 2 conducts a literature review of similar studies on image-based malware classification and loss-regularized deep learning models. The architecture and adaptive loss formulation of the proposed A-CRCL-BFFNN is also elaborated in Section 3. Section 4 gives an experimental setup and performance evaluation; results and comparative analysis and Section 5 give a conclusion of the paper by giving future research directions.

2. RELATED WORKS

ML and DL have been broadly applied to the classification of malware, especially due to the increased use of images as malware features. Early methods used features, which were hand drawn and derived out of binary files or disassembled code which was then classified with traditional classifiers. Nonetheless, these approaches demonstrated only partial resistance to code obfuscation, packing, and polymorphism, which became the impetus to transition to the approach based on deep learning and feature learning.

CNNs have extensively been used in the classification of malware images because they can acquire a spatial pattern in the visualized binaries. Abdulazeez et al. [10] tested several pretrained CNN models on malware classification and found them to achieve better accuracy relative to shallow architecture. However, CNN-based systems are very sensitive to visual similarity within malware families and in cases where structural differences are small, they do not generalize. In addition, there are fixed receptive fields which restrict the possibility of capturing long range dependencies of malware binaries.

In order to solve these problems, the combination of CNNs with recurrent or ensemble classifiers has been suggested as hybrid architecture. Singh et al. [11] presented a deep hybrid model in which the sequential feature extraction and cost-sensitive classification are used to manage malware families that are visually similar. Although improvements to performance were achieved, this technique had the disadvantage of being computationally complex and unstable when there is a large overlap between feature Jacobs in two or more classes. Equally, parallel DL models that are structured into ensembles and optimized through metaheuristic algorithms have been proven to be better at detection, but provide more overhead and convergence issues [12].

Recently, malware classification has received the attention of transformer-based architecture because it can model global dependencies. Wang et al. [13] presented a self-supervising framework masked with a Swin Transformer that is capable of classifying malware images effectively. Although transformer-based models can be competitive with regards to accuracy, they have limitations in fine-grained local feature discrimination as they use window attention mechanisms and they consume large computational amounts, making them impractical to apply in mobile malware detection systems.

The strategies such as feature fusion and knowledge distillation have also been investigated in order to improve classification results. According to Guan and Zhang [14], a malware classification technique is suggested by the authors that relies on the fusion of multi-network features and knowledge distillation. Though fusion enhanced representational richness, high-dimensional feature space created in the result also exhibited redundancy and wasted more memory. On the same note, cross-modal CNNs that focus on attention by integrating structural entropy features with malware images have performed much better, but did not explicitly learn how to preserve structural consistency of the semantic latent space [15].

Autoencoder models have been used to enhance the robustness of features by learning small latent features. The hybrid CNN-autoencoders have proven to be effective in the detection of non-linear malicious features [16]. Recent methods that focusing on control over small perturbations and other goals of preserving structural information include contractive and reconstruction-based regularization techniques. Nevertheless, these techniques are normally based on the concept of statically computed loss weighting which does not evolve with the changing dynamics of learning in training. Malware classification methods that are based on subspaces and on the kernel have also been explored in order to improve interpretability and dimensional reduction [17]. Although these approaches can shed some light on the representative trends, the range of scalability and adjustment to the complicated malware types is limited due to the set of constant geometrical assumptions.

To address these issues, the proposed work will utilize an Adaptive Cyclic Reconstruction-Regularized Bidirectional Feed Forward Neural Network that incorporates the involvement of dynamic loss regulation to regulate the reconstruction fidelity, cyclic regularity, and contractive regularization during the training. In contrast to the previous approaches, which have fixed loss formulations, the suggested approach changes the loss dominance depending on the dynamic training process, allowing stable and discriminative learning of features without the need to expand the architecture.

3. MATERIALS AND METHOD

3.1 Normal or Body Text

In this section, the proposed Adaptive Cyclic Reconstruction-Regularized Bidirectional Feed Forward Neural Network (A-CRCL-BFFNN) will be proposed to achieve effective malware classification. The essence of the proposed framework is to acquire stable, discriminative, and noise-tolerant malware representations through learning controlled by dynamically varying the numerous loss terms in training. In contrast to traditional formulations of statical losses, the model presented is able to change its learning priorities as the training progresses, thus improving the generalization and resistance to structurally similar malware variants.

The proposed methodology (Fig. 1) will start with the creation of malware representations of binary or grayscale files, then convert them to color maps to increase structural visibility to improve min-max normalization that enables uniform scaling. It is followed by the bidirectional contrastive autoencoder that supports robust feature learning that is passed into a Feed-Forward Neural Network (FFNN) to perform adaptive classifier-cyclic reconstruction-regularized classification. The critical component of this scheme is a new adaptive loss-regulation model which optimizes the stability between the reconstruction fidelity, cyclic consistency and the smoothness in the latent space on a dynamic basis during training.

$$L_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|_2^2 \tag{2}$$

$$L_{cyc} = \|f_d(f_e(\hat{x})) - x\|_2^2 \tag{3}$$

$$L_{con} = \lambda \left\|\frac{\partial z}{\partial x}\right\|_F^2 \tag{4}$$

In contrast to the models based on the theory of static losses, the proposed framework modifies the contribution of each loss component dynamically depending on the dynamics of training. The total loss shall be defined as in the following Eq. 5. and $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ are epoch-varying dynamically changing adaptive weights. Entropy variation and gradient stability are used in updating these weights (Eq. 6). Such dynamic optimization pathway is organized by this adaptive mechanism, which at the beginning of the training stage focuses on structural reconstruction. With the evolution of the model, the emphasis on the consistency is made during the mid-stage, and the optimization of the robustness and smoothness is made at the end of the phase.

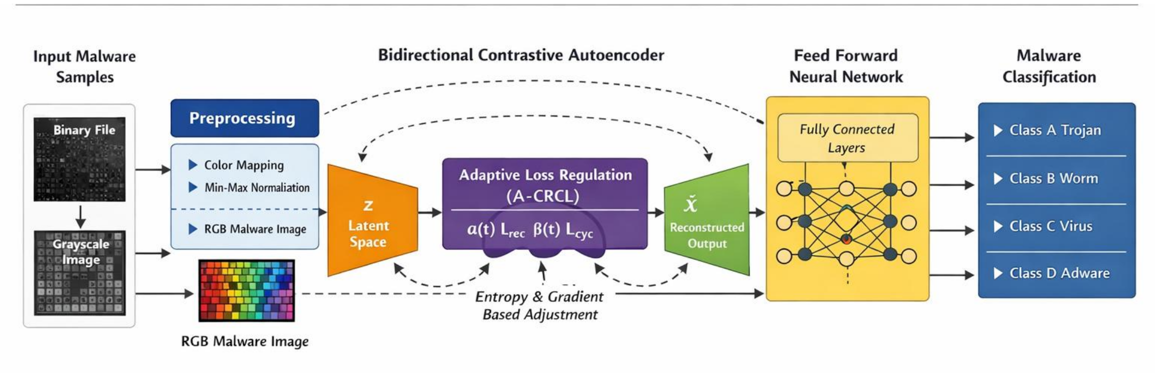

Fig. 1. A-CRCL-BFFNN Malware Classification Model

$$L_{total} = \alpha(t)L_{rec} + \beta(t)L_{cyc} + \gamma(t)L_{con} \tag{5}$$

$$\alpha(t) \propto \frac{1}{H(z)}, \beta(t) \propto \|\nabla L_{cyc}\|, \gamma(t) \propto Var(z) \tag{6}$$

The normalized latent features acquired on the adaptive BCAE is sent to a FFNN to classify malware. The FFNN has fully connected layers that have ReLU activation and dropout regularization to avoid overfitting. The last layer uses a SoftMax function to provide probabilities of classes (Eq. 7).

$$P(y|z) = SoftMax(Wz + b) \tag{7}$$

The samples of malware are initially transformed into image-based representation in order to take advantage of spatial correlations presented by binary structures. Binary malware images are converted to byte arrays and to grayscale images which are then translated back to arrays and mapped to grayscale images and vice versa. In order to make discriminative patterns more visible, the grayscale pixel intensities are allocated across RGB channels, by transforming a color map. Min-max normalization is then used to bring pixel values to the range [0,1] so that the distribution of input is even and convergence in training remains constant.

A bidirectional contrastive auto encoder (BCAE) is used to represent both forward and backward structural dependence in malware representations. The BCAE ensures consistency between encoding and decoding activities unlike the traditional autoencoders which only learn a single directional mapping and thus do not preserve important structural information. Let x[?]R the malware representation of the input. Encoder and decoder will be encoder and decoder as shown in the Eq. 1. In which, z is the latent embedding, and θ_e, θ_d are the parameters of the encoders and the decoders respectively. The contrastive learning is incorporated to maximize the distance between the classes and minimize the distance within the latent space.

$$z = f_e(x;\theta_e), \hat{x} = f_d(z;\theta_d) \tag{1}$$

The proposed A-CRCL framework combines three complementary loss functions: Reconstruction loss makes sure that the encoded latent representation preserves critical structural properties of the malware sample (Eq. 2), Cyclic consistency loss makes sure that the latent representations remain stable over encoding-decoding cycles, avoiding the loss of information in the course of bi-directional transformations (Eq. 3), To enhance robustness to small perturbations of the input and malware obfuscation, contractive regularization makes sure that the latent representations are sensitive (Eq. 4).

The algorithm provides the A-CRCL-BFFNN malware classifier process, starting with converting the input dataset in binary or grayscale into RGB images, and then performing min-max normalization to provide data consistency. A bidirectional contrastive autoencoder is then used to encode features, combined with a dynamic optimization that is used to compute dynamically varying loss weights, which are used to minimize total loss. This is then completed by the process of classifying the extracted features using a FFNN, with the final product being the predicted malware family.

Algorithm: A-CRCL-BFFNN Malware Classification

1. Input malware dataset
2. Convert binary/grayscale files to RGB images

3. Apply min–max normalization
4. Encode features using bidirectional contrastive autoencoder
5. Compute adaptive loss weights
6. Optimize total loss
7. Classify malware using FFNN
8. Output predicted malware family

4. RESULTS AND DISCUSSION

The experimental analysis was carried out in a deep learning platform written in Python and running on a Windows operating system on a high-performance computational infrastructure. Particularly, the system has used an Intel Core i7 processor with a 32 GB of RAM and NVIDIA graphics card with CUDA, which was to have a tremendous impact on the speed of the voluminous training operations. To ensure the reproducibility of the experiment as well as the possibility to scale the architecture, the proposed A-CRCL-BFFNN was implemented with the help of the TensorFlow framework. To have a detailed evaluation of the performance, the standard measures of classification, such as Accuracy, Precision, Recall, and F1-score, were used. The paper further examined computational complexity and convergence behavior to strictly test the stability of the model during training and the overall efficiency of the algorithm used.

4.1 Dataset and Data Splitting

The empirical analysis on the proposed framework was done based on two of the most popular and commonly used benchmark datasets, which were selected to test the model strictly in different conditions. The first and the most well-known is the BIG2015 dataset comprising 10,868 malware samples grouped into 9 different families and is unique in that it offers some difficulties concerning the high inter-family similarity, which also puts the discriminative power of the model to the test. The second dataset is a Malimg dataset which has 9339 malware images categorized in 25 malware families and is characterized by a high imbalance in classes and is used to test the capability of the model to work with skewed distributions. To have a thorough evaluation, a random split of the samples of the two datasets into 80 and 20 percent training and testing sets was done respectively. In addition, a 5-fold cross-validation process was used to ensure statistical validity and elimination of the overfitting issue, and the final reported results indicate the mean performance of the 5 folds.

4.2 Hyperparameter Configuration

The A-CRCL-BFFNN model described has been set in terms of hyperparameters (Table 1) that it has been determined based on a series of empirical tuning in order to develop an optimal compromise between a high level of classification and low level of computation. The process of network optimization was performed with the Adam optimizer, with a learning rate of 0.001 so that convergence was stable. The ReLU activation function was used in all the network layers and each of these layers had 256 hidden units to allow the efficient extraction of features and the ability to perform a non-linear mapping. To help increase model generalization and reduce the overfitting possibility, dropout rate of 0.5 was introduced in the training process. The training process was done in 100 epochs and the batch size was 64 as this was enough to be exposed to the data and at the same time, the memory was not overly taxed. In addition, the control of the loss weights was set up to a uniform distribution, which created a position of neutrality to the training process.

Table 1. Training Hyperparameters

Parameter	Value
Learning rate	0.001
Optimizer	Adam
Activation function	ReLU
Dropout rate	0.5
Batch size	64
Number of epochs	100
Hidden units per layer	256
Loss weight initialization	Uniform
Adaptive update frequency	Per epoch

An important aspect of the training stability was the dynamic control of the adaptive loss weights $\alpha(t)$, $\beta(t)$, and $\gamma(t)$. These weights were automatically updated at a per-epoch rate which was controlled by the monitoring of entropy change and gradient stability. The mechanism enabled the model to adapt the effect of various components of losses in real time without adding any more trainable parameter hence simplifying the optimization trajectory.

4.3 Overall Classification Performance

A detailed comparative analysis of the proposed A-CRCL-BFFNN model and existing basic deep learning models, such as CNN Vision Transformers (ViT), and Swin Transformers, is performed in Table 2 on both datasets, the BIG2015 dataset and Malimg dataset. The proposed framework showed a higher level of efficacy on the BIG2015 dataset, reaching the accuracy of 99.86% which is much higher when compared to standard CNN (96.42%) and Swin Transformer (97.80%). Likewise, with the Malimg dataset, the model achieved an impressive accuracy of 99.89%, which is higher than the baselines and has a great tolerance to different malware families.

Table 2. Training Hyperparameters

Method	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN	BIG2015	96.42	95.1	94.8	94.95
ViT	BIG2015	94.5	93.7	93.2	93.45
Swin Transformer	BIG2015	97.8	96.9	96.7	96.8
CRCL-BFFNN (Static)	BIG2015	99.71	98.6	98.68	98.63
A-CRCL-BFFNN (Proposed)	BIG2015	99.86	98.92	98.95	98.93

CNN	Malimg	96.85	96.1	95.75	95.92
Swin Transformer	Malimg	98.02	97.8	97.5	97.65
CRCL-BFFNN (Static)	Malimg	99.78	99.75	99.79	99.77
A-CRCL-BFFNN (Proposed)	Malimg	99.89	99.86	99.88	99.87

One of the most important observations made in the course of the experiment concerns the unique performance benefit presented by the adaptive loss regulation mechanism. The suggested adaptive model offered a steady increase in all important indicators in comparison with the non-adaptive model (CRCL-BFFNN). In particular, the adaptive methodology achieved greater accuracy, recall, and F1-scores (98.93% on BIG2015 and 99.875% on Malimg) which suggests that adapting weight helps the network to optimally trade-off between reconstruction fidelity and classification accuracy. These recall and F1-score improvements are specifically important in the context of malware classification, in which the false negative reduction is critical to system security.

4.4 Ablation Study

A thorough ablation study was done using progressive integration of modules into the model framework to isolate and quantify the contribution of each individual architectural component systematically. Fig. 2 provides the performance history, with the initial FFNN setting having the lowest accuracy of 95.60% on the BIG2015 data. The inclusion of the Bidirectional Contrastive Autoencoder (BCAE) with the FFNN led to significant performance increase to 98.45 which indicates a significant importance of the autoencoder in deriving strong and discriminative features in the raw input data.

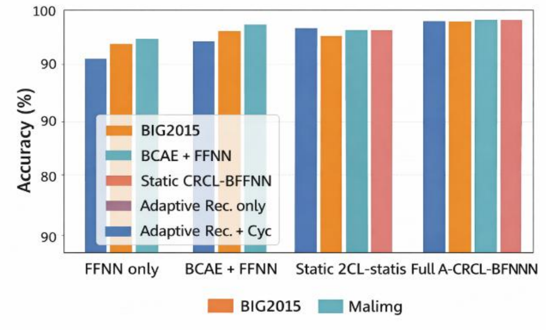

Fig. 2. Adaptive Loss Component Ablation Study

The paper also shows that the adaptive mechanism is better than the regularization that is not adaptive. Whereas the Static CRCL-BFFNN recorded a very good 99.71% accuracy, dynamic weighting showed steady incremental improvements. The Adaptive Reconstruction alone model was better than the static model and the latter was then complemented by an addition of Adaptive Cyclic consistency which made the results of this model even better. Finally, Full A-CRCL-BFFNN configuration provided the highest accuracy of 99.86% and 99.89% on BIG2015 and Malimg respectively. These results empirically confirm that although partial adaptivity is beneficial, holistic and concurrent adaptive weighting of reconstruction fidelity, cyclic consistency and latent space smoothness cannot be ignored in the achievement of stable convergence and maximum classification strength.

4.5 Computational Complexity Analysis

Fig. 3 shows a quantitative evaluation of the computational efficiency of the proposed model and the baseline architectures in more detail. A critical metric is analyzed in regard to three important metrics, Floating Point Operations (FLOPs), epoch training time, and inference latency. Although the lowest with only 0.05Million FLOPs, the base FFNN does not have the characteristics to extract features necessary to perform high-fidelity classification with a training time of 10.0 seconds. Adding the feed forward feature of bidirectional contrastive autoencoder to the CRCL-BFFNN models marginally adds the computational load to the 0.06Million FLOPs, which is a sacrifice to achieve better representation learning.

More importantly, the static CRCL-BFFNN prevention is compared to the suggested A-CRCL-BFFNN, it is evident that the addition of the adaptive loss regulation mechanism does not impose justifiable computational overhead. There is a slight increment in training time (12.0 to 12.5 seconds) and inference time (1200ms to 1210ms). The architectural efficiency of the adaptive mechanism is explained by the fact that the dynamic weight updates are not calculated through extra trainable layers or heavyweight matrix operations; the values of entropy and gradient stability are calculated through lightweight scalar computations. The A-CRCL-BFFNN, therefore, provides a better trade-off, and can provide good adaptive performance, without any loss in the deployment feasibility demanded of real-time malware detection systems.

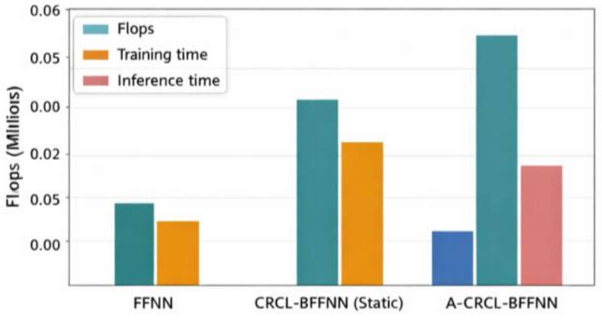

Fig. 3. Computation Complexity Analysis

4.6 Summary of Results

The findings of the experiment demonstrate the validity of the hypothesis that the adaptive cyclic reconstruction-regularized learning framework is a powerful novel contribution to malware classification and that it is architecturally sparse. The model manages to control its learning path, unaware of its actions, by the introduction of dynamic modulation in the interactions between reconstruction fidelity and cyclic consistency and the contractive losses. This capability of changing the learning priorities among various stages of training, between structural recovery and semantic consistency, is what guarantees the production of latent representations that are more stable and at the same time highly discriminative. The system therefore has high feature separation over stationary weighting schemes and high feature separations are obtained without complex increases in network depth or number of parameters.

Moreover, such a result of about 0.2% in the accuracy improvement might seem insignificant in its own right, but its practical consequences in the field of cybersecurity are significant. Considering a large-scale malware detection system where the system is processing millions of files per day, each percentage point change will result in hundreds of misclassified examples, and thus greatly decrease the chances of missing a threat or a false positive. The regular outperformance of the A-CRCL-BFFNN in comparison to the variants with the static loss and general deep learning models confirms its strength, since it makes it one of the most powerful solutions to the specific categorization of various and dynamic malware families.

5. CONCLUSION

In this paper, the A-CRCL-BFFNN was introduced to support the sound classification of malware families. The proposed model overcomes some of the major constraints of current image-based malware classifiers by providing an adaptable loss control system enabling the loss controller to dynamically strike a balance between reconstruction accuracy, cycle, and contractive regularization in training. The adaptive strategy allows the model to set the learning priorities in the various stages of the training, unlike the case in static-loss formulations, which leads to more stable and discriminative latent representations. The proposed model can effectively learn both the forward and backward structural dependencies present in malware binaries by injecting a bidirectional contrastive autoencoder with adaptive cyclic regularized reconstruction learning. Image color mapping of malware and min-max normalization also increase the visibility of the features and stabilize network convergence. Large-scale testing of BIG2015 and Malimg datasets indicate that A-CRCL-BFFNN, with minimal computational cost, has significantly better accuracy, recall, and F1-score in comparison with traditional deep learning models and fixed CRCL-based frontends.

Both the ablation and convergence studies affirm that adaptive weighting of loss elements is significant and effective in enhancing generalization as well as overfitting especially in classifying visually related malware families. Even the slightest improvements manifested by adaptive learning are propagated into the meaningful improvement in the large-scale malware classification setting, which confirms the practical importance of the suggested solution. Although the proposed approach is effective, it is tested on the basis of static malware image data, which, perhaps, cannot reflect all the real-world problems: advanced obfuscation, packing, polymorphism, and adversarial manipulation. The given framework will be expanded in the context of future work to include strategies of obfuscation-aware and adversarial training to improve the level of robustness even more.

6. REFERENCES

[1] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, “A survey on ML techniques for Multi-Platform Malware Detection: securing PC, mobile devices, IoT, and cloud environments,” *Sensors*, vol. 25, no. 4, p. 1153, Feb. 2025, doi: 10.3390/s25041153.

[2] A. Hussain, A. Saadia, M. Alhussein, A. Gul, and K. Aurangzeb, “Enhancing ransomware defense: deep learning-based detection and family-wise classification of evolving threats,” *PeerJ Computer Science*, vol. 10, p. e2546, Nov. 2024, doi: 10.7717/peerj-cs.2546.

[3] M. Ashawa, N. Owoh, S. Hosseinzadeh, and J. Osamor, “Enhanced Image-Based malware classification using Transformer-Based Convolutional Neural Networks (CNNs),” *Electronics*, vol. 13, no. 20, p. 4081, Oct. 2024, doi: 10.3390/electronics13204081.

[4] F. Harrou, B. Bouyeddou, A. Dairi, and Y. Sun, “Exploiting Autoencoder-Based anomaly detection to enhance cybersecurity in power grids,” *Future Internet*, vol. 16, no. 6, p. 184, May 2024, doi: 10.3390/fi16060184.

[5] F. S. Alsubaei, A. A. Almazroi, W. S. Atwa, A. A. Almazroi, N. Ayub, and N. Z. Jhanjhi, “Adaptive malware identification via integrated SimCLR and GRU networks,” *Scientific Reports*, vol. 15, no. 1, p. 25309, Jul. 2025, doi: 10.1038/s41598-025-08556-4.

[6] B. Thiyam and S. Dey, “An improved deep autoencoder-based network intrusion detection system with enhanced performance,” *International Journal of Internet Technology and Secured Transactions*, vol. 13, no. 3, pp. 270–290, Jan. 2024, doi: 10.1504/ijitst.2024.136658.

[7] S. Li, J. Wang, S. Wang, and Y. Song, “PAFE: A lightweight visualization-based fast malware classification method,” *Heliyon*, vol. 10, no. 16, p. e35965, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35965.

[8] Q. Shi *et al.*, “TransGraphNet: robust detection of malicious encrypted network traffic via transformer and graph neural models,” *PeerJ Computer Science*, vol. 11, p. e3353, Nov. 2025, doi: 10.7717/peerj-cs.3353.

[9] H. KauserSk and M. AnuV, “Hybrid deep learning model for accurate and efficient android malware detection using DBN-GRU,” *PLoS ONE*, vol. 20, no. 5, p. e0310230, May 2025, doi: 10.1371/journal.pone.0310230.

[10] F. A. Abdulazeez, I. T. Ahmed, and B. T. Hammad, “Examining the performance of various pretrained convolutional neural network models in malware detection,” *Applied Sciences*, vol. 14, no. 6, p. 2614, Mar. 2024, doi: 10.3390/app14062614.

[11] S. Singh, D. Krishnan, V. Vazirani, V. Ravi, and S. A. Alsuhibany, “Deep hybrid approach with sequential feature extraction and classification for robust malware detection,” *Egyptian Informatics Journal*, vol. 27, p. 100539, Sep. 2024, doi: 10.1016/j.eij.2024.100539.

[12] W. Yan, J. Tang, and S. Stucki, “Design and implementation of a lightweight deep CNN-Based plant biometric authentication System,” *IEEE Access*, vol. 11, pp. 79984–79993, Jan. 2023, doi: 10.1109/access.2023.3296801.

[13] F. Wang *et al.*, “MalSort: Lightweight and efficient image-based malware classification using masked self-supervised framework with Swin Transformer,” *Journal of Information Security and Applications*, vol. 83, p. 103784, May 2024, doi: 10.1016/j.jisa.2024.103784.

[14] J. Yang, J. Cui, and M. Huang, “Extreme Weather Prediction for Highways Based on LSTM-CNN,” *5th International Conference on Applied Machine Learning (ICAML)*, pp. 191–196, Jul. 2023, doi: 10.1109/icaml60083.2023.00045.

[15] R. Song, J. Yv, and Z. Du, “Test analysis and optimization of valve flow characteristic curve of 660MW unit,” *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 53, pp. 255–258, Dec. 2022, doi: 10.1109/imcec55388.2022.10019839.

[16] S. Andriani, S. Galantucci, A. Iannacone, A. Maci, and G. Pirlo, “CNN-AutoMIC: Combining convolutional neural network and autoencoder to learn non-linear features for KNN-based malware image classification,” *Computers & Security*, vol. 156, p. 104507, May 2025, doi: 10.1016/j.cose.2025.104507

[17] C. Chen, Y. Hu, L. Gao, and B. Li, “An Optimization Method for Impedance Matching Teardrop via Pads Based on TDR Simulation,” *International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–2, Aug. 2023, doi: 10.23919/aces-china60289.2023.10249876