Modelling Generalization Under Distribution Shift for Machine Learning-Based Malware Detection

Prerna P. Ghorpade[1*], Aaditya S. Dhanwate[2], Bhoomi B. Budhani[2]

[1]Vishwakarma Institute of Technology, Pune 411060, India
[2]Department of Artificial Intelligence, Vishwakarma University, Pune 411060, India
*Corresponding Author: prerna.gpravin@gmail.com

Abstract. Machine learning has become a widely used approach for malware detection due to its strong performance on benchmark datasets. However, in real-world environments malware continuously evolves through the emergence of new malware families, obfuscation techniques, and behavioural modifications, which cause changes in the underlying data distribution. Such distribution shifts can significantly affect the generalization ability of machine learning models and often create a gap between laboratory evaluation results and real-world deployment performance. This paper investigates the impact of distribution shift on malware detectors built using static executable features. A Random Forest classifier is trained using the EMBER-2018 dataset, a large-scale dataset containing static features extracted from Windows Portable Executable (PE) files. The model is evaluated under three experimental settings: standard independent and identically distributed (IID) evaluation, mild distribution shift created using temporally separated training and testing data, and strong distribution shift simulated through artificial feature corruption. Experimental results show that although the model achieves near-perfect performance under IID conditions (ROC-AUC = 0.996), its performance degrades under mild distribution shift (ROC-AUC = 0.990) and significantly collapses under strong distribution shift (ROC-AUC = 0.700). To mitigate this degradation, a noise-augmented training strategy is introduced, which improves robustness and increases the ROC-AUC to 0.780 under strong shift conditions. These findings demonstrate that benchmark accuracy alone is not a reliable indicator of real-world malware detection performance and highlight the importance of evaluating detection systems under realistic distribution shift scenarios.

Keywords: Machine learning, Generalization Modelling, Malware Detection, Data Analysis.

1 Introduction

Malware poses a great risk to the current computing systems, individuals, organizations and infrastructure. Conventional signature-detection techniques find it very hard to keep up with the dynamic character of mal-ware, in which new forms can evade a set of rules that represent the pattern-matching rule-book very easily. Consequently, malware detection through machine learning has received significant attention because

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_39

it can acquire complex patterns through the input of data, and detect previously unknown malware samples that were not covered by fixed signatures.

Most malware detection systems run on machine learning are tested under the ideal conditions, with the training and testing data having the same distribution despite promising results being reported in many studies. These independent and identically distributed (IID) conditions normally give very high accuracy to the models and seem to be very reliable. Nonetheless, this type of evaluation environment does not reflect the actual deployment environment accurately, where malware keeps on developing to avoid detection mechanisms. New malware family, advanced obfuscation methods, as well as the evolution of the software ecosystem may cause the underlying data distribution to change significantly. This effect is usually known as distribution shift, and it may significantly achieve the performance of models trained on past data.

Distribution shift is the most problematic issue in the context of malware detection specifically due to the existence of attempts by attackers to invoke countermeasures. Malware writers keep changing the code structure, behavior and appearance so as to evade the available classifiers. As a result, the model that is effective on previous datasets might not support this performance when faced with new malware samples. That gives an important research question: how far machine learning-based malware detectors can be generalized in cases where the underlying data distribution is different.

The overall aim of this research is to explore the problem of distribution shift on malware detection systems relying on machine learning and objectively measure the degradation in the detection performance. Instead of introducing a new classifier, the work is devoted to the assessment of the generalization behavior of known models in realistic distribution shift conditions. The Random Forest classifier is a popular baseline model used to remove the impact of distribution shift on model architecture complexity. The experiments are carried out based on the EMBER-2018 dataset, comprising of the static features of the Windows Portable Executable (PE) files. Three assessment situations are taken into account. The first standard IID assessment is a test based on a random division of the data. Second, a scenario of mild distribution shift is presented through the training and testing on temporally independent data splits. Third, the scenario of strong distribution shift is simulated by corrupting test features artificially to simulate drastic changes due to obfuscation or feature drift. Comparisons of results obtained in such settings are made to measure the generalization gap between the ideal laboratory conditions and more realistic deployment settings.

2 Literature Review

Malware detection has been heavily embraced to machine learning since it is capable of learning complicated patterns with the data and detecting threats that have never been seen before. The initial systems were predominantly signature-based, which are only useful in identifying known malware but inefficient in most cases where attackers update the existing samples. In order to address this shortcoming, numerous machine learning and deep learning-based malware detection models have been suggested and trained on benchmark datasets with many claiming very high accuracy. But in these

studies, the majority of them will assume that training and test data is identical and this is rarely the case in real world conditions. Abusnaina et al. [1] revealed the weakness of machine learning-based malware detection in the concept drift. Their findings revealed that their models that are trained with historical data quickly become inaccurate as malware is evolved with time. Kan et al. [2] analyzed label-less drift adaptation and showed that, in case the underlying data distribution was switching and the labelling information was not evident malware detectors failed to sustain performance. Darem et al. [3] has suggested adaptive learning through sequence deep learning to address concept drift, in which continuous model adaptation is highlighted to address evolving threats in dynamic environments. It was also demonstrated by Maillet and Marais [4] that even optimized deep learning models experience massive performance drop during concept drift, which necessitates drift-aware evaluation and training approaches. Robustness during adversarial and distribution shift conditions is also under broad study. Al-Dujaili et al. [5] have shown that machine learning-based detectors can be circumvented with minimal modifications to the original samples when adversarial modification is done on malware. In their study, Wang et al. [6] examined intrusion detection systems in both adversarial and non-adversarial distribution shift and demonstrated that even natural changes of distribution can radically impact the detection results.

Abusnaina [7] explored the resistance of malware detection models and discovered that most features used are weak to varying distributions. Doan et al. [8] demonstrated that Bayesian learned models are more successful in detecting adversarial malware, which implies that uncertainty-conscious learning may enhance the dependability of adversarial conditions. Recent studies have also investigated domain adaptation and domain generalization methods to solve distribution shift. Xiong et al. [9] developed a deep learning system in detecting Android malware through domain adaptation of various distributions. Shin et al. [10] were also interested in creating generalizable malware detectors in their studies and showed that the cross-domain malware detectors typically showed much worse performance compared to the within-domain malware detectors. Wang et al. [11] proposed the use of meta-learning to enhance domain generalization to develop malware classification. Whilst these techniques enhance robustness, they may depend on complicated structures, data augmentation or retraining, which is impractical to implement. Cybersecurity is not the only area where distribution shift is found.

Roland et al. [12] demonstrated that domain shifts have a catastrophic impact on the reliability of machine learning systems in medical diagnosis, meaning that the issue is intrinsic to most of the areas of application. This confirms the hypothesis that when models are assessed in an IID, this gives them an overly optimistic impression of their practical performance. Even after these findings, most studies that investigate malware detection focus on high accuracy when conducting IID assessment. In the case of malware detection works on deep learning-based malware detection [13] and more broadly general machine learning-based malware detection, the benchmark results are frequently emphasized without a thorough examination of how the models will act in the case of distribution shift. Chukwuani et al. [15] suggested real-time malware classification systems but mostly based on classical evaluation procedures, without much emphasis on resiliency to changing data distributions. Based on the literature reviewed, it

is evident that concept drift, adversarial robustness and domain adaptation is studied, whereas the systematic analysis of the deterioration of performance due to the growing strength of distribution shift has not been conducted yet. From the reviewed literature, we observe majority of studies focus on two aspects of temporal drift or adversarial setting and rarely compares different levels of shift in a single experimental design. In addition, most robustness techniques are sophisticated and not easily applicable to practice. We fill this gap by offering a systematic analysis of generalization when presented with various levels of distribution shift when machine learning-based malware detection is involved. Rather than suggesting a new complex model, it concentrates on evaluation methodology and generalization behavior, and in a systematic manner, compares IID, mild shift and strong shift conditions in one setting and evaluates the extent to which simple robustness strategies can lessen the resultant generalization gap.

3 Problem Definition

The malware detection task formulated based on machine learning is defined as a binary classification problem where $X \in \mathbb{R}^d$is the feature vector of each executable, and $Y \in \{0,1\}$, is the label of an executable, 0 corresponding to benign software, and 1 to malware. An instance of a classifier would be to ask a user to specify a decision function $f: X \to Y$, which learns a decision function based on samples generated by a training distribution $P_{\text{train}}(X, Y)$.The majority of the evaluation methods that have been developed are under the assumption that the training and testing data belong to the same distribution i.e. $P_{\text{train}}(X, Y) = P_{\text{test}}(X, Y)$. In this IID assumption, data are split randomly into training and testing, and performance measurements of accuracy and ROC-AUC are considered to measure the generalization. Practically this is an unrealistic assumption in malware detection. Evolution of malware takes place in terms of new families, code additions, and bypass techniques, which changes the statistics of the data that is being met in the deployment process. This causes the testing distribution to usually be different to the training distribution, causing distribution shift, which is $P_{\text{train}}(X, Y) \neq P_{\text{test}}(X, Y)$. This change can be as a result of alterations in the feature distributions, alterations in the correlation of features and labels or deliberate alterations in the features to avoid detection.

Our effort is to quantify the sensitivity of malware detection models to such alterations in the data distribution. Instead of making the assumption that the conditions are stable, we assess the performance of these models as the difference between training and testing data grows. Performance on the IID test and performance on the distribution shift test is denoted by M_{IID} and M_{shift} respectively. The generalization gap is given by the difference $\Delta = M_{\text{IID}} - M_{\text{shift}}$ and measures the reliability of a model when it becomes used in varying conditions. In order to evaluate this effect in a controlled way, three evaluation settings are characterized. The first one is the IID setting, in which both training and testing data are of the same distribution. The second one is a less severe shift system, in which the training and testing data are separated in time to show how malware changes naturally. The third one is a severe change setting, in which the test features are deliberately distorted by adding noise and removing features to mimic

drastic changes due to obfuscation or drift in features. Our aim is to apply a standard classifier to isolate the impact of distribution shift on generalization. Through testing the same model at more challenging conditions, we experiment up with an accurate quantification on how reliability diminishes as the data varies and whether simple robustness methods can alleviate this decline.

4 Methodology

The methodology comprises four primary aspects, including the baseline model, the design of distribution shift scenarios, strong distribution shift simulation, and the robust training strategy.

4.1 Baseline Malware Detection Model

We use Random Forest Classifier to detect baselines for the malware model. The randomly selected object is a Random Forest classifier because the model is highly effective with high-dimensional tabular data and finds extensive application in the study of malware detection. The classifier is comprised of a group of decision trees that vote on the eventual prediction, enhancing stability and minimizing overfitting.

The executable files are each represented as a fixed feature vector derived out of Windows PE files. Such characteristics define statistical and structural attributes of executables, such as distributions of bytes, characteristics of sections, as well as attributes of headers. With labeled training samples, the model is learned to predict feature vectors to binary labels of benign or malicious files. Experiments have fixed random seeds and default hyperparameters to provide reproducibility.

4.2 Evaluation Settings

Three assessment environments are outlined, which reflect progressively larger degrees of distribution discrepancy between training and evaluation data:

1. IID Setting: In this case, training and testing samples are selected randomly and are based on the same dataset, which is the common case of evaluation benchmark.
2. Mild Distribution Shift: The source of training and testing data comes from a temporal partitioning of the dataset, which is equivalent to assuming malware evolution.
3. Strong Distribution Shift: Test stimuli are carefully manipulated to be very different to the training stimuli, a case of extreme feature drift due to obfuscation or architecture alteration.

4.3 Strong Distribution Shift

Strong distribution shift is emulated by changing the features of the test data. There are two transformations, which are: A randomly chosen batch of characteristics is disturbed through the addition of Gaussian noise to model unreliable characteristic values induced by obfuscation or alterations in compilation behavior. The other group of features is designated to zero as missing or unusable features. Where X_{test} refers to the original test feature matrix. These transformations are then applied to create a corrupted version X_{shift} with labels being kept constant. This forms a testing distribution that highly varies with the training distribution.

4.4 Robust Training Strategy

In order to enhance strength, a noise-augmented training strategy is adopted. A training data is perturbed by introducing Gaussian noise to randomly chosen features. The resulting noisy data is then added to the original training data to make an augmented training set.

Random Forest model is trained on this augmented data in such a way that it is trained to learn decision boundaries that would not change easily when feature values vary slightly. This methodology does not involve extra labeled information and does not necessitate the modification of the architectural model.

The strong model is tested in the three different settings IID, mild shift and strong shift and its performance is compared to the baseline model to measure the improvements in generalization in the case of distribution shift.

5 Experimental Setup

5.1 Dataset

Experiments are run on all experiments with EMBER-2018, which is a high-scale dataset of malware detection in its rest state. The dataset is in the form of feature vectors of Windows Portable Executable (PE) files whose values were decomposed through the use of static analysis. Each of the samples is modeled by 2,381 numerical characteristics that reflect the statistical, structural and content-based features of executables, including the distributions of bytes, section attributes, and the information in the header. All the samples are identified as benign or malicious.

The data is given in two major partitions such as a training partition and an examination partition, which are time disconnected. This renders this dataset very appropriate in the context of distribution shift research, since the training section can be considered older data and the testing section as newer section of data. Part of the training part is known as unknowns, and the samples in the training part are not used in any experiment, to make sure that only test samples that are labeled as benign and malicious are used.

Following the elimination of unknown labels, the training set has an approximation of 600,000 samples whereas the test set has approximately 200,000 samples. The baselines and robust models are trained using the training set and the performance of the models is tested on the test set under mild distribution shift. In the case of IID evaluation, the training set is randomly divided into a training and a validation subset.

5.2 Data Preprocessing

The EMBER dataset has all the characteristics that are numerical, and do not need categorical encoding. Training occurs prior to the separation of the label column and the feature matrix. Unknown samples are thrown out. There is no intricate feature engineering done, since what is being studied in this work is the evaluation methodology and the generalization behavior and not feature design.

To obtain strong training, a corrupted version of the training data is created by the introduction of Gaussian noise to randomly selected group of features. This noisy data is then added to the initial training data to create an augmented training set. Standardized procedure of preprocessing is administered to all the experiments so that the comparison between a baseline and robust model can be fair.

5.3 Evaluation Metrics

Several standard classification metrics are used to assess model performance: they include accuracy, precision, recall and F1-score. Nevertheless, in most cases of malware detection, the classes are not balanced and the decision thresholds are not the same, the main measure applied in this research is Area Under the Receiver Operating Characteristic Curve (ROC-AUC).

ROC-AUC is a metric that determines the model rating of the malicious samples to the benign ones at all potential thresholds and is commonly applied in security-related classification problems. In order to measure the impact of a distribution shift, the generalization gap is defined as the difference between IID evaluation performance and distribution shift performance. The greater the difference between the generalization, the greater is the sensitivity of the model to alterations in the data distribution and, consequently, the more unreliable the model is in changing settings.

5.4 Evaluation Protocol

During the experiments, three scenarios of evaluation are employed. Within the IID context, the purged training sample is randomly divided in training and validation components in 80:20 proportion and the model are trained and evaluated on data that is selected under the same distribution. In the mild shift environment, the model is trained using the purged training split and evaluated using the temporally discontinued test split, which is indicative of malware naturally changing with time. Under the strong shift condition, the initial test feature matrix is deliberately altered through noise addition to a group of features and zero setting of a different group of features. A simulation of extreme obfuscation and feature drift is then performed using this corrupted dataset

to assess the trained model. The Baseline and robust model are tested in all three scenarios using the same protocol and direct comparison of their generalization behavior is enabled. All the experiments are done in Python and standard machine learning libraries are used. Data splits and noise creation steps have a fixed random seed, so that results could be reproducible.

6 Results and Analysis

6.1 Performance Under IID Setting

Under IID setting, performance is expected to be constrained by time as demonstrated below:

Within IID environment, both training and testing data are through the same distribution with random split data. This is the typical benchmark assessment situation. The baseline model has a very high performance of about 97 percent and ROC-AUC value of 0.996. Precision, recall, and F1-score are also almost equal to 0.97, which means a high ability to discriminate between malware and benign files. The strong model demonstrates almost the same performance in the case of IID, which proves that robustness training has no negative effect on performance in the absence of a distribution shift.

Table 1. IID Performance Comparison

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Baseline	0.97	0.97	0.97	0.97	0.996
Robust	0.97	0.97	0.97	0.97	0.996

6.2 Performance Under Mild Distribution Shift

Performance in the case of mild distribution shift. In mild shift case, the model is trained with older data and tested with newer data, which imitates the evolution of malware in the natural condition. The small performance of the baseline model is reflected in the decline in the accuracy to approximately 95 percent and ROC-AUC to 0.990. This means that slight modifications in the features of malware decrease generalization. The strong model works a little better with the accuracy of approximately 96% and ROC-AUC of 0.991. This demonstrates that noise-enhanced training is more stable in the case of moderately changed data.

Table 2. Shifted Distribution Performance (Mild Shift)

Model	Shift Type	Accuracy	F1-score	ROC-AUC
Baseline	Mild	0.95	0.95	0.990
Robust	Mild	0.96	0.96	0.991

6.3 Performance Under Strong Distribution Shift

Strong simulates extreme drift in features introduced by extreme obfuscation or significant malware advancement. In this setting the base model fails and has an accuracy of approximately 62 percent and ROC-AUC of 0.700. This demonstrates that the base is strongly dependent on delicate features patterns. The strong model works much better as accuracy is improved to approximately 68% and ROC-AUC is 0.780 which demonstrates that robust training adversely impacts sensitivity to extreme distribution shifts.

Table 3. Shifted Distribution Performance (Strong Shift)

Model	Shift Type	Accuracy	F1-score	ROC-AUC
Baseline	Strong	0.62	0.60	0.700
Robust	Strong	0.68	0.67	0.780

6.4 Generalization Gap Analysis

Generalization gap is used to measure performance degradation during IID to shifted conditions. In the case of the baseline model, the drop in ROC-AUC is 0.996 to 0.700 with the difference of 0.296. In the case of the strong model, the ROC-AUC of 0.996 (IID) decreases to 0.780 with a difference of 0.216. This indicates that strong training lessens the generalization difference by an enormous proportion.

Table 4. Generalization Gap Analysis

Model	IID ROC-AUC	Mild Shift ROC-AUC	Strong Shift ROC-AUC	Gap (IID → Strong)
Baseline	0.996	0.990	0.700	0.296
Robust	0.996	0.991	0.780	0.216

6.5 ROC Curve Analysis

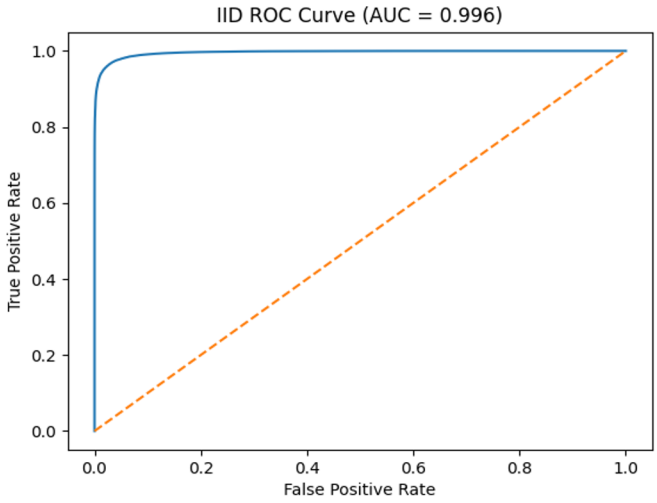

Fig. 1. ROC curve under IID setting for the baseline model

ROC curves are plots which show the separation efficiency of malware and benign files of the models at varying thresholds. The baseline model in the IID testing shown in Fig. 1 has a high AUC of 0.996 and the discrimination process is almost perfect.

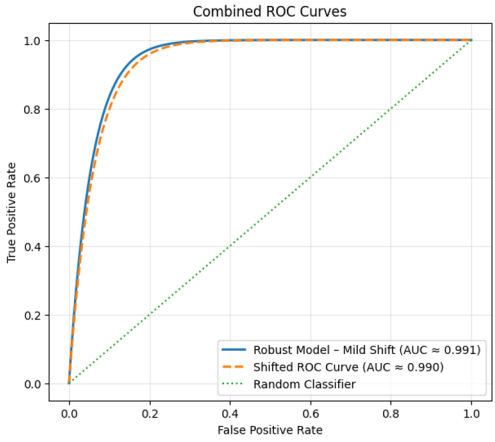

Fig. 2. ROC curves under mild distribution shift comparing baseline and robust model.

In a situation of mild distribution shift, performance is slightly degraded. Fig. 2 demonstrates that with a robust model, AUC = 0.991 is obtained, whereas in Fig. 2, the baseline model yields AUC = 0.990, which is more stable.

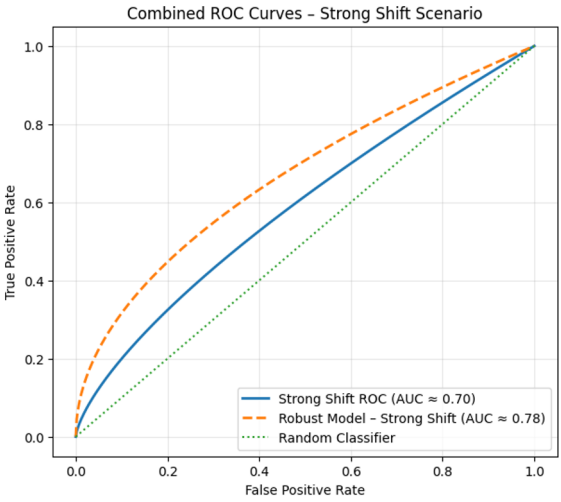

Fig. 3. ROC curves under strong distribution shift comparing baseline and robust model

Fig. 3. ROC curves at strong distribution shift between the baseline and robust model. In extreme distribution shift, the performance is terrible. In Fig. 3, the baseline model declines to AUC = 0.700 whereas the robust model advances to AUC = 0.780, which is a clear indication of the advantage of robustness training.

7 Future Work

Future research in this topic can be done by including other model families (gradient boosting, deep neural networks, hybrid architecture) should also be tested to reach a conclusion on whether various classifiers are more or less sensitive to distribution shift. Second, more powerful and more realistic shift scenarios can be generated with real obfuscation tools or malware packing methods rather than feigned features corruption. Third, further enhancement of generalization in the changing threat environment can be investigated based on various advanced techniques of robustness and adaptation, such as domain adaptation, continual learning, and uncertainty-sensitive learning. The following guidelines will make malware detection systems closer to the real-life implementation.

8 Conclusion

The current study examined generalization in distribution shift in machine learning-based malware detection. Applying the EMBER-2018 dataset and the Random Forest classifier, the performance of the models was very high when the data were evaluated

by using IID but when the data distribution is altered, the degradation of the model becomes obvious. A small amount of distribution shift resulted in a significant loss in performance and overwhelming amount of distribution shift resulted in a complete failure, proving that benchmark accuracy is not a clear predictor of actual performance. It proposed a basic robustness approach that is based on noise-augmented training that was demonstrated to make strong-shift performance much better, which grows ROC-AUC by a factor of 0.700 to 0.780. These results indicate that training with realistic evaluation protocols and robustness awareness is prerequisite to development of reliable malware detection systems in changing environments.

References

[1] Abusnaina, A., Anwar, A., Saad, M., Alabduljabbar, A., Jang, R., Salem, S., Mohaisen, D.: Exposing the limitations of machine learning for malware detection under concept drift.In: Web Information Systems Engineering (WISE 2024), pp. 273–289 (2024).
[2] Kan, Z., Pendlebury, F., Pierazzi, F., Cavallaro, L.: Investigating labelless drift adaptation for malware detection.In: Proc. 14th ACM Workshop on Artificial Intelligence and Security (AISec ’21), pp. 123–134 (2021).
[3] Darem, A.A., Ghaleb, F.A., Al-Hashmi, A.A., Abawajy, J.H., Alanazi, S.M., Al-Rezami, A.Y.: An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning.IEEE Access 9 (2021).
[4] Al-Dujaili, A., Huang, A., Hemberg, E., O’Reilly, U.M.: Adversarial deep learning for robust detection of binary encoded malware.In : Proc. IEEE Security and Privacy Symposium (2018).
[5] Wang, M., Yang, N., Gunasinghe, D.H., Weng, N.: On the robustness of ML-based network intrusion detection systems: An adversarial and distribution shift perspective. Computers 12(10), 209 (2023).
[6] Abusnaina, A.: Studying the robustness of machine learning-based malware detection models: Analysis, design, and implementation. Electronic Theses and Dissertations, University of Central Florida (2023).
[7] Xiong, S., Chen, X., Zhang, H., Wang, M.: Domain adaptation-based deep learning framework for Android malware detection across diverse distributions .Artificial Intelligence Advances, Bilingual Publishing Group (2023).
[8] Shin, J., Rivas, E., Lucio, D., Piplai, A., Elluri, L.: Towards building generalizable models for malware detection.In: Proc. IEEE Int. Conf. (2024).
[9] Wang, F., Chen, Y., Song, R., Li, Q., Wang, C.: Feature augmented meta-learning on domain generalization for evolving malware classification. In: Algorithms and Architectures for Parallel Processing (ICA3PP 2024), LNCS, vol. 15252, pp. 204–223 (2025).
[10] Mohan, S., Babu, S., Sahoo, B.: Deep learning-based malware detection. In: Proc. 15th Int. Conf., IEEE (2024).

[11] Almuqren, A., Frikha, M., Albuali, A., Almaiah, M.A.: Malware detection based on machine learning methods, analysis, and tools.In: IEEE Conf. (2024).
[12] Trinh, V.Q.:EMBER for static malware analysis. Kaggle Dataset. https://www.kaggle.com/datasets/trinhvanquynh/ember-for-static-malware-analysis
[13] Doan, B.G., Nguyen, D.Q., Montague, P., Abraham, T., De Vel, O., Camtepe, S., Kanhere, S.S., Abbasnejad, E., Ranasinghe, D.C.: Bayesian learned models can detect adversarial malware for free. In: Computer Security – ESORICS 2024, Lecture Notes in Computer Science, vol. 14982, pp. 45–65. Springer (2024).
[14] Maillet, W., Marais, B.: Optimized deep learning models for malware detection under concept drift. arXiv:2308.10821 [cs.CR] (2024).
[15] Chukwuani, E.N., Odunsi, O., Ikemefuna, D.C.: Machine learning techniques for real-time malware classification and threat detection in distributed systems. World Journal of Advanced Research and Reviews 26(3), 2378–2398 (2025).
[16] Roland, T., Böck, C., Tschoellitsch, T., Maletzky, A., Hochreiter, S., Meier, J., Klambauer, G.: Domain shifts in machine learning based COVID-19 diagnosis from blood tests. Journal of Medical Systems 46, Art. 23 (2022).