# CDAD: Concept Drift Adaptation Denial Attack in Android Malware Detection

② a phoenomenon

① this sentence is too high-level.

④ unfinished sentence.

## Abstract

Machine learning-based Android malware detectors are used to alleviate mobile security threats in the real world. Unfortunately, with the evolution of malware, deployed detection models will soon become outdated, known as concept drift. To address this issue, Concept drift adaptation strategies based on active learning are proposed to improve detection models' performance. The core of concept drift adaptation lies in selecting high value samples during the testing phase and optimizing the model based on these samples. However, we find that ~~we~~ adversaries can construct poisoned samples based on high-value benign samples, controlling sample selection and model updating. Based on ~~the above~~ insights, we propose the concept drift adaptation denial (CDAD) attack to prolong the survival time of new malware. We conduct the first CDAD attack on Android malware detection models constructed by four mainstream concept drift adaptation strategies. We evaluate CDAD attack on an Android malware concept drift dataset containing over 580,000 samples spanning more than 10 years. Experimental results indicate that the latest concept drift adaptation method is vulnerable to our CDAD attack, the attack success rate can reach 92.77%. We achieve an attack success rate of 88% across four mainstream concept drift strategies under the white-box threat model and an attack success rate of 82.95% under the black-box threat model. Furthermore, the impact on the original model's performance (measured by the F1 score) during the attack process is minimal, with an average reduction of less than 0.02, demonstrating high stealthiness in CDAD attack.

③ To address this issue,

⑤ existing

⑥ not based on.

⑦

⑧ consolidate into one sentence.

## 1 Introduction

Android operating system has become indispensable to people's lives over the last decade. As of January 2024, the Android operating system ranked first in the global operating system market share, reaching 41.63% [45], with nearly 4 billion active users worldwide [?]. Unfortunately, mobile devices and applications powered by the Android operating system have been selected as valuable targets by cyber criminals [5]. According to a security analysis by Kaspersky, even the official Google Play Store had over 600 million malware downloads in 2023 [51]. Facing the massive amount of malware generated daily, researchers have proposed automatic detection detectors for Android malware based on machine learning [63].

However, deploying Android malware detectors in the real world faces many challenges. One of the most critical challenges is that real-world data distribution can change over time, yielding the phenomenon of concept drift [41, 54]. Researchers have demonstrated that Android malware detectors, which performs well on training datasets, experiences a decline in its F1 score from 0.99 to 0.76 within approximately 6 months when faces with concept drift [12]. A direct solution is to add new data to the training dataset to ensure that training dataset distribution ~~is consistent with the real world~~ data distribution. But the number of new Android applications in the real world is overwhelmingly large. Google Play launched 1,069 mobile apps every day in 2024 [8]. Therefore, obtaining sample labels for all new data is imp~~ossible, which leads to an~~ infeasible insufficient quantity of training datasets, resulting in a decline in the performance of Android malware detectors.

citation

Existing ~~several related landmark papers~~ solutions [6, 12, 16, 60] mainly focus on concept drift adaptation through active learning for Android malware detection. In order to tackle the practical limitations previously mentioned, such as the cost of labeling, the aforementioned research [6,60] has devised diverse evaluation methods aimed at assessing the value of test data. Researchers introduce high-value samples to mitigate model performance degradation caused by concept drift [12, 16]. We conduct relevant experimental evaluations for mainstream concept drift adaptation strategies [6, 12, 17, 60]. ~~Please refer to Appendix F for detailed experimental evaluation data~~. But we find that most previous research has primarily focused on improving the performance of concept drift adaptation. The vulnerabilities of concept drift adaptation through active learning for Android malware detectors have received little attention.

⑨ what?!

② study one specific paper/solution is less interesting. I thought you studied many.

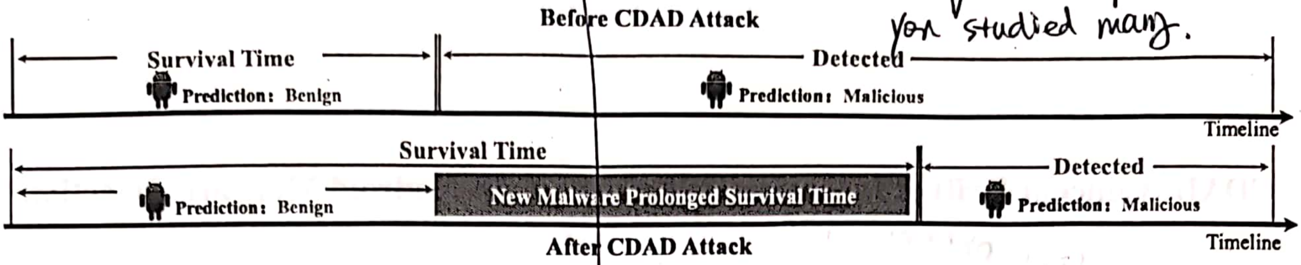

Figure 1: Prolonged survival time of new malware under CDAD attack

Previous research either focuses on the security of training phase (poisoning attacks [15, 25, 33, 42, 59]) or the security of inference phase (evasion attacks [4, 23, 36, 40]). And we note that the concept drift adaptation focuses on the concept drift samples selection, which is different from the training and inference phase. Therefore, researching the security of concept drift adaptation methods is of utmost importance. The purpose of concept drift adaptation is to shorten the survival time of new malware, so we investigate whether there exists an attack method that can prolong the survival time of new malware (as shown in Figure 1).

In order to study the changes in the survival time of new Android malware under the concept drift adaptation methods, we ~~propose~~ three research questions:

① intend to answer

Q1-What is the survival time of new malware samples under latest concept drift adaptation based on active learning [12]?
Q2-Can we design an attack method to prolong the survival time of new malware in the face of the optimal concept drift adaptation method?
Q3-How effective is the attack method, and what factors influence the attack's effectiveness?

③ we probably don't need these Qs. as they are repeating what we already said.

We explore the answers to the above questions by studying the Android malware detectors with concept drift adaptation capabilities. We choose Android malware because of the availability of public, large-scale, and timestamped datasets(e.g., AndroZoo [?, 1, 2, 12, 67]).

no bold.

**To answer Q1**, we apply the optimal concept drift adaptation methods [12] against the new malware samples. We find that some new malware samples have longer survival time than other old malware samples(§3). For example, the survival time of the tascudap family reaches over 2 years. Among the new malware families each month, over 95% of them contain samples with a survival time greater than 0 month. This implies that there are vulnerabilities in the current concept drift adaptation strategies. For detailed data on the survival time of new malware, refer to Figure 8 in Appendix 9.

**To answer Q2**, we propose the concept drift adaptation denial (CDAD) attack (§5) to prolong the survival time of new malware. CDAD attack can efficiently generate poisoned samples with clean labels. Our attack framework comprises three modules: surrogate model training, malware attack value assessment and malware survival time prolongation.

\emph{3

④ what?!

**To answer Q3**, we use the Attack Success Rate (ASR) to measure the attack effectiveness, which refers to the proportion of samples whose survival time has been effectively prolonged among all attacked samples. We conduct our evaluation under both white-box and black-box threat models (§4) and analyze the attack influencing factors (§6).

To better demonstrate the vulnerability of mainstream concept drift adaptation methods to CDAD attacks, we conduct experiments on four mainstream Android malware concept drift detection strategies, including Hierarchical Contrastive Classifier (HCC) [12], Transcending (TRANS) [6], high-dimensional outlier distance (CADE) [60], and uncertainty (UNCER) [17]. Our attack evaluation dataset spans 10 years and the total number of samples of our dataset reaches more than 580000. The experimental evaluation results show that our CDAD attack method achieves an attack success rate of 92.77% on the latest Strategy (HCC).

**Our Contributions.** To sum up, we mainly make the following contributions:

- We have discovered significant security vulnerabilities in the concept drift adaptation strategies proposed ~~in recent top security conferences~~. The survival time of new malware can be prolonged by our CDAD attack.
- We propose an automatic poisoned sample generation framework for CDAD attack. This framework generates poisoned samples with clean labels, and it does not require any modifications to new malware samples. Therefore, our framework reduces the attack cost and enhances the stealthiness of our CDAD attack.
- We conduct CDAD attack effectiveness tests on two Android malware concept drift dataset over a period of 10 years, and conduct detailed discussions on 3 primary attack influencing factors, attack stealthiness and the costs of attackers and defenders.
- We provide an open source implementation of CDAD[1] attack to benefit future research in the community and encourage further improvement on our approach.

[1] CDAD is available at https://anonymous.4open.science/r/Denial-of-Concept-Drift-Adaptation-Code-C0EF

no bold

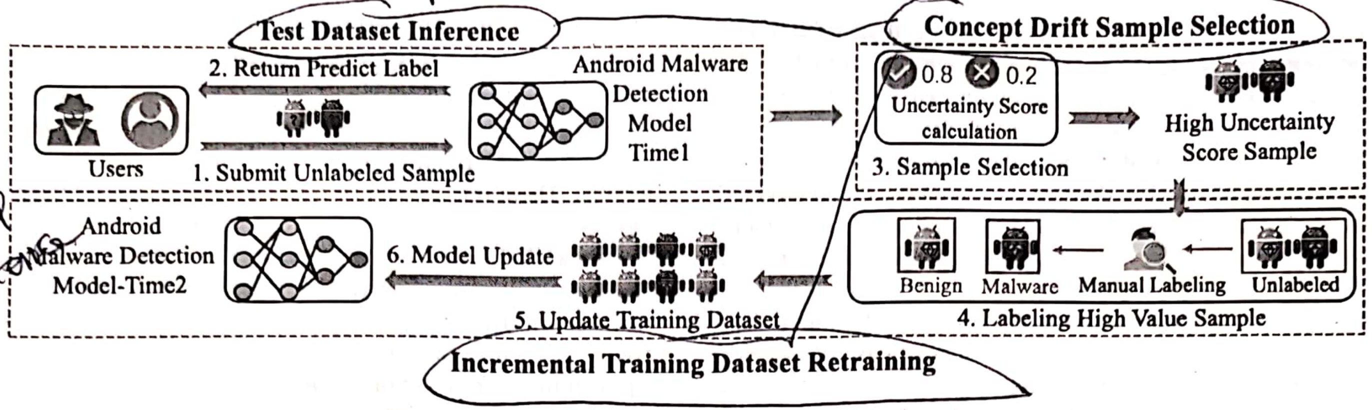
Test Dataset Inference
Concept Drift Sample Selection
2. Return Predict Label
Android Malware Detection Model Time1
Users
1. Submit Unlabeled Sample
0.8
0.2
Uncertainty Score calculation
High Uncertainty Score Sample
3. Sample Selection
Android Malware Detection Model-Time2
6. Model Update
Benign
Malware
Manual Labeling
Unlabeled
5. Update Training Dataset
4. Labeling High Value Sample
Incremental Training Dataset Retraining

Figure 2: Concept drift adaptation based active learning

② add a sentence

## 2 Background

① almost the same title. what is the difference?

### 2.1 Concept Drift Adaptation

**Behavior Pattern.** Different concept drift adaptation methods also share common behavior patterns (as shown in Figure 2). The classical behavior pattern consists of three sequential execution phases: test dataset inference, concept drift sample selection, and incremental training dataset retraining.

**Stage1-Test dataset inference:** New Android applications are submitted to the detection model for testing. The detection model gives a predicted label. In the case of binary classification, the return label value is 0 or 1.

**Stage2-Concept Drift Sample selection:** The primary goal of sample selection is to find concept drift samples. It can address the issue of how to allocate labeling budgets for new data when the overall labeling budget is limited. Researchers design different sample selection strategies to find the most helpful samples for improving the model's performance under limited label budget.

**Stage3-Incremental training dataset retraining:** The model trainer adds selected samples as an incremental update part of the training dataset. After retraining, model performance degradation caused by concept drift can be alleviated.

### 2.2 Concept Drift Adaptation Strategy

The ~~mainstream~~ existing concept drift adaptation methods in the field of Android malware detectors include the following four strategies.

**1) Model Uncertainty.** The core idea of uncertainty measurement [17] is to detect concept drift based on the output layer of the target model. The model gives priority to selecting the samples with high uncertainty of the current model for labeling. A common uncertainty measurement for a neural network is to use one minus the max softmax output of the network.

**2) Encoder Space Distance.** CADE [60] trains an encoder through existing labeled data for learning a compressed representation (dimension reduction) of a given input distribution. Then, the newly obtained test samples can be provided to the encoder to obtain the encoder's spatial features. Finally, the distance function can effectively identify concept drift samples far from the existing training dataset.

**3) Credibility and Confidence.** Transcending [6] introduced the thery of conformal prediction [52] (credibility and confidence) into the field of concept drift adaptation. Given a new test sample, Transcending first computes the non-conformity score of the sample. Then, it computes credibility as the percentage of samples in the calibration set that have higher non-conformity scores than the test sample. Finally, it computes confidence as one minus the credibility of the opposite label. A lower credibility score or a lower confidence score means the test sample is more likely to have drifted.

**4) Hierarchical Contrastive Loss.** The method proposed by Chen et al. [12] is currently the best-performing strategy in Android malware concept drift adaptation. The model consists of two modules. The first module is an encoder and the second module acts as the classifier. In terms of loss function settings, to ensure that the model is robust to concept drift, the training loss of the model is set to the sum of hierarchical contrast loss and classification loss.

### 2.3 Adversarial Attacks on What?

③

~~Adversarial machine learning attacks are widely studied in multiple fields~~ [7]. Currently, adversarial attacks against Android malware detectors can be roughly divided into two categories: evasion attacks and poisoning attacks.

**Evasion Attacks** have received extensive attention in the field of Android malware detection [11, 26, 39, 68]. Specifically, the attacker's goal in an evasion attack is to add a small perturbation to a target malware sample to get it misclassified. Such perturbed example is called an adversarial example.

**Poisoning Attacks** are one of the most dangerous threats to machine learning models [25, 42]. These attacks assume attackers can inject poisoned samples into the training dataset.

\paragraph instead of \textbf

In poisoning attacks, the adversary's goal is to degrade model performance through some malware modifications to the training dataset. After being trained on the poisoned dataset, the model's performance degrades at test time. According to the different degradation degrees of the victim model, poisoning attacks can be roughly divided into untargeted and targeted poisoning attacks. The goal of untargeted poisoning attacks is to decline the overall performance of the victim model. The goal of targeted poisoning attacks is to force the victim model to perform abnormally on a specific input class. Backdoor attacks [15, 25, 42] are a special case of targeted poisoning attacks where the victim model only misclassify samples containing specific triggers.

In summary, existing research has either focused on the model inference security under static conditions (evasion attacks) or the security of the training process of the model (poisoning attacks). However, we found that in addition to the training phase, poisoning attacks are also very likely to occur during the sample selection phase of the concept drift adaptation.

## 3 Motivation

① What?? I am looking for scientific motivation

Our motivation comes from a key real-world observation: the longer the survival time of new Android malware, the more benefits the attacker obtains, such as monetary gain, user privacy, etc. For example, Zimperiumz Lab discovered a new type of Trojan named GriftHorse in 2021 [62], and the infected device would pay the attacker 30 euros a month. This new malware infected over 10 million user devices in over 70 countries and regions in a few months.

How is this related?

**Validating the Intuition.** In order to understand the survival time of new Android malware, we use the existing optimal concept drift adaptation method (HCC [12]) to quantitatively analyze the survival time of new malware samples.

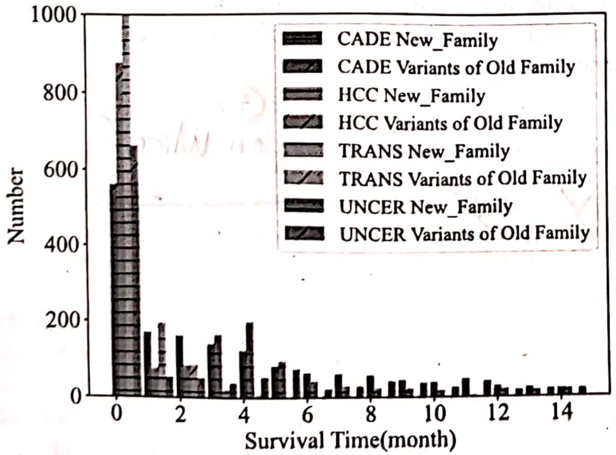

Figure 3: Survival time of new malware

As shown in Figure 3, some new malware samples still have a long survival time. Among them, families such as 'gomanag' can even survive for more than 5 years (62 months). Under the latest HCC method, the average survival time of new malware families is 4.71 months, while that of new variants is 3.76 months. Therefore, we can conclude that the survival time of new malware is crucial to malware developers.

**Research Motivation.** Unfortunately, there is a lack of research on how attackers (malware developers) can achieve malware survival time prolongation under concept drift adaptation. Therefore, we investigate the risks of new malware survival time prolongation in the existing concept drift adaptation strategies. Our goal is to reveal the dangers of CDAD attacks and draw the attention of researchers to the risks in concept drift adaptation process of Android malware detection model.

## 4 Threat Model ②

~~To ensure the reasonableness of the research hypothesis, we present the threat model based on previous works and real world situations.~~

In our attack scenario, a capable attacker can carry out attacks based on a white-box threat model. This means that the attacker can access information such as the victim model's active learning incremental training dataset, concept drift adaptation strategies, sample feature vectors, and model parameters. This setting follows Kerckhos' principle [36], ensuring that the security of the model does not rely on secrecy. Additionally, concept drift adaptation strategies may ultimately favor some state-of-the-art methods [13], it will be challenging to maintain concept drift adaptation strategies strict confidentiality.

We further validate the effectiveness of our proposed CDAD attack under the conditions of a black-box threat model. In a black-box threat model [73], the attacker cannot obtain the training dataset or model parameters from the victim model. Therefore, the attacker can only rely on a surrogate model for approximate analysis, as demonstrated by previous work [48, 71].

③ This section is about threat model, not "validation"

## 5 Attack Methodology

### 5.1 Adversary's Challenges

Although attackers have some capabilities and knowledge ~~(mentioned in §4)~~, attacks against concept drift adaptation still face severe challenges. These challenges make our CDAD attack different from previous attacks (as shown in Table 1). The detailed challenges faced by attackers are as follows:

**Attack Continuity.** Previous poisoning attacks [21, 29, 50, 61] typically validate their effectiveness under the condition of a fixed training dataset without considering the scenario where the training dataset is continuously updated. One of the

④ one sentence.

Some of these should be discussed in Intro.

① This table looks too big. ② no bold !!

Table 1: Adversarial setting (●:have this characteristic, ○:lack of this characteristic)

| Attack Method | Attack Continuity | Training out of Control | Malware Integrity | Label Correctness |
|---|---|---|---|---|
| Android HIV [11] | ○ | ○ | ○ | ● |
| HRAT [68] | ○ | ○ | ○ | ● |
| Severi et al. [42] | ○ | ○ | ○ | ● |
| Li et al. [25] | ○ | ○ | ○ | ● |
| AdvDroidZero [18] | ○ | ○ | ○ | ● |
| Jigsaw Puzzle [59] | ○ | ○ | ○ | ● |
| **Our Work (CDAD)** | ● | ● | ● | ● |

③ Waste of Space!

most notable features of active learning is that it continually introduces new data into the training dataset. Therefore, ensuring the continuity of the attack's effectiveness during the model update process is challenging. This paper is the first to study the continuity of poisoning attacks on the concept drift adaptation process for Android detection models.

**Training Process is out of control.** Even under the white-box threat model, the attacker cannot directly poison the training dataset. They only have sample submission and query access to the latest state of the victim model. Compared with the attack challenge of existing attack scheme [42, 46], our attacker ~~challenge~~ has a higher degree of difficulty.

② What ?!

**Malware Integrity.** Although some attackers have used code updates to counter the detection methods [10, 11, 27, 44], thereby prolonging the survival time of new malware. However, this method requires attackers to pay expensive costs for code development. Furthermore, frequent code updates by attackers may alert malware detectors. Therefore, we consider maintaining malware integrity to be a unique challenge.

**Label Correctness.** The labels of poison samples uploaded by the attacker will not be mislabeled, which is different from the assumption of many advanced model attacks in the image field [?, 70]. The reason is that sample labeling in the active learning process is done manually. This is also a special challenge in our CDAD attack.

## 5.2 Attack Method

Taking the above attack challenges as the prerequisite, we propose concept drift adaptation denial (CDAD) attack. The overall workflow is shown in Figure 4. The first module is to conduct a surrogate model, which is responsible for simulating the target model and providing a basis for subsequent attack steps (§5.2.1). The target model refers to the victim model mentioned previously. The second module is malware attack value assessment, which is responsible for assessing the attack value of new malware samples and selecting attack strategy (§5.2.2). The third module involves the generation of poisoned samples (§5.2.3). To clearly describe our CDAD attack, we have introduced relevant symbols. Please refer to Appendix A for a complete list of symbols and their meanings.

### 5.2.1 Surrogate Model Training

We build a surrogate model for obtaining the information needed for subsequent attack operations without providing new malware to the target model. The training process of the surrogate model is independent of the training process of the target model.

Considering that our attack scheme has obvious temporal characteristics, we use $i$ to uniformly represent different model update time nodes. In the initial state, both the surrogate model and the target model are trained based on their respective initial training datasets. Due to the openness of Android data collection, the initial training datasets for the surrogate model and the target model are consistent. Therefore, we also set the initial parameters $\theta_a^0$ of the surrogate model and the initial parameters $\theta_d^0$ of the target model to be consistent. Differences between the surrogate model and the target model come from the model retraining stage. At time node $i$, regarding training data acquisition, the owner of the target model is always a large security vendor, so it can effectively collect new Android samples $D_t^i$ and detect concept drift samples $D_c^i$. More importantly, the owner of the target model has the ability to conduct reliable sample label analysis on concept drift samples $D_c^i$.

However, as attacker, we lack the ability to provide reliable labels for concept drift samples. We collect new data samples $D_t^i$ from the real world, identifies concept drift samples $\bar{D}_c^i$, and obtains query results as pseudo labels for concept drift samples $\bar{D}_c^i$ based on the target model $\theta_d^i$. It is important to emphasize that due to the openness of the Android platform, our ability to collect data aligns with the owner of target model, so we all get Android samples $D_t^i$. Refer to Appendix B for further datasets collection details. Additionally, our purpose is to approximate the detection ability of the target model $\theta_d^i$, so we do not need to care about the true label of the concept drift samples $\bar{D}_c^i$ but only needs to get the sample prediction result (pseudo label) of the target model $\theta_d^i$, which greatly reduces our label cost. Then, the surrogate model $\theta_a^i$ is retrained based on concept drift samples $\bar{D}_c^i$ to ensure that its detection performance is always close to the target model.

In addition, our surrogate model construction method difanalysis paid by the target model, the more favorable it is for attackers to launch attacks.

**2) Label Budget Occupied by Attackers:** The proportion of poisoned samples within the label budget represents the intensity of the CDAD attack. The higher the proportion of poisoned samples within the label budget, the greater the attacker's attack cost. To effectively illustrate the impact of label budget proportion on attack effectiveness, we set the label budget proportions to 100%, 70% and 50% respectively. We then evaluate how different label budget proportions affect the attack result. As shown in Table 4, different label budget proportions have different impacts on ASR. The average ASR of multiple sets of attack tests can still reach 87.73%. Furthermore, we can observe a clear trend in ASR: As the proportion of label budget decreases, ASR gradually declines. Specifically, the settings of 70% and 50% proportions result in a decrease of 3.88% and 11.25% in ASR, respectively.

Table 4: Proportion of label budget occupied by attacker

| **Proportion** | **F1** | **FPR (%)** | **FNR (%)** | **ASR (%)** |
|---|---|---|---|---|
| 100 | 0.90 | 0.44 | 14.12 | **92.77** |
| 70 | 0.90 | 0.43 | 14.22 | **88.89** |
| 50 | 0.90 | 0.46 | 14.21 | **81.52** |

**3) Different Feature Extraction Methods:** Considering that Android malware detectors, in practice, may adopt different feature extraction methods, we extract static features [5], such as permissions, from our self-constructed dataset to conduct attack experiments on heterogeneous features. The attack results are shown in Table 5. It can be seen that our proposed CDAD attack can achieve effective attacks (with an ASR of over 90%) against different feature extraction methods.

Table 5: Feature heterogeneity

| **Feature** | **F1** | **FNR (%)** | **ASR (%)** |
|---|---|---|---|
| API [67] | $0.90_{-0.02}$ | $14.12_{+1.12}$ | **92.77** |
| Drebin [5] | $0.67_{+0.01}$ | $44.57_{+1.39}$ | **95.03** |

### 6.2.3 Black Box Attack Effectiveness

To demonstrate the effectiveness of conducting CDAD attacks under the black-box threat model, we have set up the role of a weak attacker. Specifically, compared to the strong attacker setting, we adjust some settings for the weak attacker. In terms of mastery of target model information, weak attackers cannot access the parameters and training dataset of the target model. Since complex models represent greater computational overhead, we have weakened the model settings for the attacker. Because of the current optimal concept drift adaptation methods mainly consist of an encoder and a classifier, we provide four sets of comparative settings based on the target model (as shown in Table 6).

Table 6: Attack effectiveness under model heterogeneity

| **Model** | **F1** | **ASR (%)** | **R-ASR (%)** |
|---|---|---|---|
| Enc-Cla | 0.89 | **82.95** | **82.95** |
| Enc↓-Cla | 0.87 | **72.31** | **59.09** |
| Enc-Cla↓ | 0.83 | **82.22** | **44.31** |
| Enc↓-Cla↓ | 0.86 | **87.18** | **39.77** |

Previous research has indicated that clean-label attacks suffer from end-to-end performance degradation [53], as model updates can lead to a deterioration in attack effectiveness. However, the CDAD attack alleviates the issue of diminished attack effectiveness under end-to-end conditions. The experimental result data in Table 6 shows that the average ASR under various black-box settings reaches 81.17%. The reason is that although the attacker may not have complete knowledge of the target model in an end-to-end setting, the CDAD attack can effectively influence the data that the target model relies on for updates. Therefore, CDAD indirectly impacts the target model's updates, enhancing the attack's effectiveness in an end-to-end setting. This demonstrates that, under the assumption of a black-box model, our attack method still poses a significant security threat to the currently optimal concept drift adaptation methods.

Moreover, we notice that the reduction in ASR caused by the weakened encoder is more pronounced than that caused by the weakened classifier, with a difference of nearly 10%. This indicates that the leakage of encoder information poses a greater threat to the target model. Our experimental analysis also echo the current best concept drift strategies that rely on the encoder to learn the similarities among malware families.

$$\mathbf{R\text{-}ASR} = \frac{\textbf{NSAS (Weak\_Model\_Settings)}}{\textbf{NSAS (Equal-Enc\&Cla)}} \tag{3}$$

NO bold!

Additionally, we observe an interesting phenomenon. The ASR under the synchronized weakening of both the encoder and classifier are the highest among all settings, even surpassing the control group under model alignment by approximately 10%. To investigate the reasons behind this phenomenon, we analyze the selection of attack targets under different settings. We find that while the synchronized weakening setting exhibits an advantage in terms of the ASR metric, it demonstrates a disadvantage in the number of attack targets. The absolute number of attacks in the synchronized weakening setting only accounts for 55% of the attacks in the control setting. Therefore, we conclude that due to the difference in attack value assessment capabilities resulting from model misalignment, the evaluation of attack effectiveness under the black-box assumption should take into account