RL-MAGE: Strengthening Malware Detectors against Smart Adversaries

Adarsh Nandanwar, Hemant Rathore,
Sanjay K. Sahay, and Mohit Sewak

BITS Pilani, Department of CS & IS, Goa Campus, India
{f20180396, hemantr, ssahay, p20150023}@goa.bits-pilani.ac.in

Abstract. Today, android dominates the smartphone operating systems market. As per Google, there are over 3 billion active android users. With such a large population depending on the platform for their daily activities, a strong need exists to protect android from adversaries. Historically, techniques like signature and behavior were used in malware detectors. However, machine learning and deep learning models have now started becoming a core part of next-generation android malware detectors. In this paper, we step into malware developers/adversary shoes and ask: Are machine learning based android detectors resilient to reinforcement learning based adversarial attacks? Therefore, we propose the *RL-MAGE* framework to investigate the adversarial robustness of android malware detectors. The RL-MAGE framework assumes the grey-box scenario and aims to improve the adversarial robustness of malware detectors. We designed three reinforcement learning based evasion attacks *A2C-MEA*, *TRPO-MEA*, and *PPO-MEA*, against malware detectors. We investigated the robustness of 30 malware detection models based on 2 features (android permission and intent) and 15 distinct classifiers from 4 different families (machine learning classifiers, bagging based classifiers, boosting based classifiers, and deep learning classifiers). The designed evasion attacks generate adversarial applications by adding perturbations into the malware so that they force misclassifications and can evade malware detectors. The attack agent ensures that the adversarial applications' structural, syntactical, and behavioral integrity is preserved, and the attack's cost is minimized by adding minimum perturbations. The proposed TRPO-MEA evasion attack achieved a mean evasion rate of 93.27% while reducing the mean accuracy of 30 malware detectors from 85.81% to 50.29%. We also propose the *ARShield* defense strategy to improve the malware detectors' classification performance and robustness. The TRPO-MEA ARShield models achieved 4.10% higher mean accuracy and reduced the mean evasion rate of re-attack from 93.27% to 1.05%. Finally, we conclude that the RL-MAGE framework improved the detection performance and adversarial robustness of malware detectors.

Keywords: Android Malware · Deep Learning · Evasion Attack · Machine Learning · Greybox Environment · Reinforcement Learning.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

1 Introduction

Since the launch of the android operating system in 2008, it has rapidly gained popularity. During Google I/O 2021, Google announced that the number of active android users had surpassed 3 billion [4]. However, due to newer versions of android releasing frequently, OEMs have a hard time catching up, resulting in version fragmentation. They usually stop updating older smartphones due to the high cost of maintenance. According to a study, over 40% of android users no longer receive security updates, making them vulnerable to malware attacks [1]. This leads users to lean towards anti-malware software/malware detectors to detect malware or prevent malware attacks.

Android occupies over 70% market share of smartphone operating systems [2]. Due to this over-dependence on one platform, its security is critical. In the past, signature, heuristic, specification, sandbox, etc. techniques have been used to detect malware [20, 13]. With the rapid developments in machine learning, it has also found its use in malware detection and has shown a good success rate in detecting old and new malware. However, researchers have demonstrated that machine learning and deep learning classification models are susceptible to adversarial attacks [9, 21]. This generated doubts about the reliability of the machine learning based malware detectors against adversarial attacks and if they are safe for real-world deployment. It is essential to be aware of the weaknesses of the malware detectors before it gets exploited by malware developers/adversaries.

The adversarial attack scenarios can be grouped into three broad categories based on the attacker's knowledge of the target system. First, is *white-box scenario*, where the attacker has full knowledge, like, the dataset used to train/test classifiers, features used, classifiers used, and other parameters, etc. It is far from reality, where the attacker generally has partial information about the malware detection ecosystem called the *grey-box scenario*. Finally, the third category is *black-box scenario*, where the attacker has no information about the malware detector. In recent years, many researchers have proposed adversarial attacks on malware detectors in a white-box scenario with good evasion rates [23, 14, 10]. However, still limited work has been done in the grey-box scenario. Therefore, in this work, we focus on the grey-box scenario where the attacker knows about the dataset and features used but has zero information about the classifiers and other parameters used in the malware detector.

Using an adversarial evasion attack, the attacker can bypass the malware detector and install malware on the target system, putting the user's data at risk. Research on adversarial attacks and adversarial defenses complement each other, both aiming to improve the adversarial robustness of machine learning based malware detectors. In this work, we propose a framework called the *Reinforcement Learning based Malware Attack in Greybox Environment (RL-MAGE)* to investigate and improve the adversarial robustness of the machine learning based malware detectors. Reinforcement learning has recently become more popular because of rapid advancements in computational and storage capabilities. In this work, we conduct an in-depth comparative study on the performance of three different *on-policy* reinforcement learning algorithms in malware eva-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

sion and adversarial defense tasks. We designed three *Malware Evasion Attacks (MEA)* namely *A2C-MEA*, *TRPO-MEA*, and *PPO-MEA*, in the RL-MAGE framework and compared their performance. The *MEA* perform untargeted evasion attacks on malware detectors that aim to introduce *type-II false-negative errors*. The attacks are designed to achieve high evasion rates while minimizing the execution cost by limiting the perturbations allowed by the agent. The attack agent also ensures the adversarial applications' structural, syntactical, and behavioral integrity. We tested the performance of 30 distinct malware detectors based on 4 families of classifiers against the MEA attacks. Later, we propose an adversarial retraining based defense strategy called the *ARShield* to improve the adversarial robustness of malware detectors and defend against adversarial attacks. With this work, we make the following contributions:

1. We proposed *Reinforcement Learning based Malware Attack in Greybox Environment (RL-MAGE)* to investigate the detection performance and robustness of malware detectors.
2. We developed 30 malware detectors based on 2 features (*android permissions* and *android intents*) and 15 distinct classifiers from 4 families, namely: *machine learning classifiers*, *bagging based classifiers*, *boosting based classifiers*, and *deep learning classifiers*. The 30 malware detector models achieved a mean accuracy and AUC of 85.81% and 0.86, respectively.
3. We designed 3 *Malware Evasion Attacks (MEA)* to expose vulnerabilities in the above 30 malware detectors models. The *A2C-MEA* achieved a mean evasion rate of 91.40% with just 1.74 mean perturbations that reduced the mean accuracy from 85.81% to 51.22% in 30 malware detectors. The *TRPO-MEA* achieved a mean evasion rate of 93.27% with just 1.95 mean perturbations and reduced the mean accuracy from 85.81% to 50.29% in the same 30 malware detectors. Finally, *PPO-MEA* accomplished a 92.17% mean evasion rate with 2.05 mean perturbations, reducing the mean accuracy from 85.81% to 50.79% in 30 malware detectors. We also list the most pertubated android permissions and android intents during the above attacks.
4. We developed *ARShield defense* to improve the robustness and detection performance of malware detectors. The mean evasion rate of *A2C-MEA*, *TRPO-MEA*, and *PPO-MEA* reattacks against *ARShield models* was drastically dropped from 91.40% to 1.32%, 93.27% to 1.05% and 92.17% to 1.40%, respectively. The mean accuracy of *A2C-ARShield*, *TRPO-ARShield*, and *PPO-ARShield* models also improved from 85.81% to 89.89%, 89.91%, and 89.83%, respectively.

The rest of the paper is organized as follows. The proposed RL-MAGE framework, Malware Evasion Attack (A2C-MEA, TPRO-MEA, and PPO-MEA), and ARShield defense are explained in Section-2. Experimental setup is described in Section-3. Experimental results of baseline malware detectors, MEA attack on baseline detectors, ARShield detectors, and MEA reattack on ARShield detectors are discussed in Section-4. Related work is explained in Section-5, and the conclusion is presented in Section-6.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

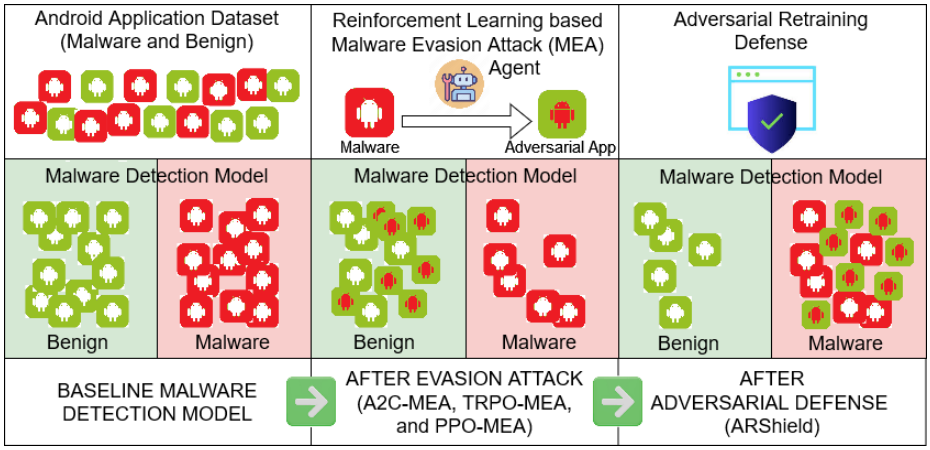


Fig. 1: Proposed RL-MAGE framework to improve detection performance and robustness of malware detectors.

2 Adversarial Attack and Defense

2.1 Proposed RL-MAGE Framework

Figure 1 summarises the *Reinforcement Learning based Malware Evasion Grey box Environment (RL-MAGE)* framework visually. The RL-MAGE framework aims to improve the malware detectors to defend against *Malware Evasion Attacks (MEA)*. The framework consists of 3 steps. The ***Step-1*** is dataset gathering, followed by constructing baseline malware detection models. The ***Step-2*** aims to develop malware evasion attacks using reinforcement learning against baseline malware detection models. These attacks generate adversarial applications to evade the malware detectors. The ***Step-3*** is to develop an adversarial defense *(ARShield)* to counter the evasion attacks and to improve the robustness of the malware detectors.

2.2 Malware Evasion Attack (MEA)

A reinforcement learning setup consists of an *environment* and an *agent*. The environment defines the observation space, action-space and returns a reward for an action performed by the agent. The agent interacts with the environment and adjusts its internal weight based on the *reward* received for the *action* performed. In our case, the state of the environment is a multi-binary representation of the android application in terms of its features (android permission/android intent). The environment calculates the reward by taking the difference between the probability of the state being *malware* for the current state and the following state (after taking action) as given by the underlying malware detector.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

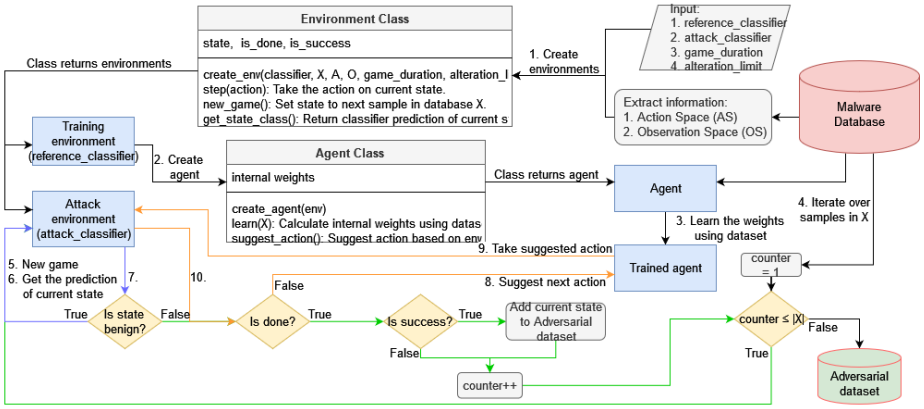


Fig. 2: Overview of the steps involved in the RL-MAGE based adversarial attack to create an adversarial dataset.

Steps involved in the MEA attack on malware detectors are described in figure 2. The attack process can be divided into 2 phases: *training phase* and *attack phase*.

The *training phase* is performed once for the reinforcement learning agent to learn the environment and devise a strategy to navigate it. *Exploration* is done in a training environment which is constructed using reference malware detectors. The environment is designed to sequentially go through all the available malware applications and expose the agent to all types of applications for it to learn. The agent is restricted by the maximum number of perturbations/alterations that it is allowed to make. The limit is 5 for android permissions and 4 for android intents. Once the agent learns from the training environment using the rewards received for its actions, we move to the next phase.

In the *attack phase*, we create attack environments for each malware detection model being attacked and release the agent in them. If the environment's state is classified as *benign*, we reset the environment and move to the next malware. Whenever the state of the environment is classified as *malware*, the agent performs actions from the environment's action space until we encounter the stop condition. The stop condition occurs either when the malware application is misclassified as benign or if the number of permissible actions has crossed the limit. In case of misclassification, the final state of the environment is used to rebuild the android application and added to the adversarial dataset.

Reinforcement learning agents may use different algorithms to learn the strategy to navigate the environment. These are used to adjust the internal weights and parameters using the states, actions, and rewards while interacting with the training environment. In this work, we explore the *on-policy algorithms* where the policy used to take action is the same policy being evaluated and updated. We train the following Malware Evasion Agents (MEAs) for our attacks:

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

1. **A2C-MEA**: The Advantage Actor Critic (A2C) MEA uses the A2C algorithm [11], which is a synchronous and deterministic variant of Asynchronous Advantage Actor Critic (A3C). It avoids the use of a replay buffer by using multiple workers.
2. **TRPO-MEA**: This MEA uses the Trust Region Policy Optimization (TRPO) algorithm [17], which introduces a KL-Divergence constraint while taking steps. This is similar to a distance measure between probability distributions.
3. **PPO-MEA**: The Proximal Policy Optimization (PPO) algorithm [18] borrows the multiple worker idea from A2C and the Trust region idea from TRPO. PPO uses clipping to prevent huge updates to the policy.

2.3 Adversarial Retraining Defense (ARShield)

We designed the *ARShield defense* to develop ARShield malware detectors that are more immune to malware evasion attacks. The ARShield is based on adversarial retraining based defense strategy [9, 13]. The MEA alters the malware applications to force the malware detectors to misclassify. When there is a misclassification, the final state of the environment is saved after converting it back into a feature vector, in the same format as the original dataset used to train the malware classifiers. A new dataset is constructed by including the original dataset with their original labels used to train baseline models and the new adversarial samples with their labels set to malware. We then retrain all the malware detectors using this new adversarial dataset resulting in more robust ARShield models that are more immune to malware evasion attacks.

3 Experimental Setup

The dataset used for the experiments contains 5721 benign applications downloaded from the *Google Play Store* [5] and screened using *VirusTotal* [6]. The dataset also contains 5553 malicious applications from the Drebin dataset [7]. We then use the reverse engineering tool *Apktool*[3] to develop a parser that extracts *android permission* and *android intent* usage in each android application. This parser recorded 195 permissions and 273 intents to create 2 feature vectors (android permission and android intent). We then train 30 different malware detection models using 15 distinct classification algorithms belonging to 4 families (basic machine learning classifiers, bagging based classifiers, boosting based classifiers, and deep learning classifiers). Table 1 shows classifies and their corresponding categories. The detailed discussion on the design and parameters of these malware detection models is explained in other papers [15, 19]. We used a train test split of 70:30. All the experiments were conducted on the *Google Colab Pro* platform using the *Python* programming language using *scikit-learn*, *TensorFlow*, and *OpenAI Gym*.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

Table 1: Classifiers and their families used in RL-MAGE framework.

Machine Learning (ML) Classifier(s)	Logistic Regression (LR) Support Vector Machine (SVM) Kernel Support Vector Machine (KSVM) Decision Tree (DT)
Bagging based Classifier(s)	Random Forest (RF) Bootstrap Aggregation Logistic Regression (BALR) Bootstrap Aggregation Kernel Support Vector Machine (BAKSVM)
Boosting based Classifier(s)	Adaptive Boosting (AB) Gradient Boosting (GB) eXtreme Gradient Boosting (XGB) Light Gradient Boosting Machine (LGBM)
Deep Learning (DL) Classifier(s)	Deep Neural Network 0 Hidden Layer (DNN0L) Deep Neural Network 1 Hidden Layer (DNN1L) Deep Neural Network 2 Hidden Layer (DNN2L) Deep Neural Network 4 Hidden Layer (DNN4L)

3.1 Performance Metrics

In this work, we use the following performance metrics to evaluate different phases of the RL-MAGE framework.

- **Accuracy**: is used to measure the performance of a malware detector. It is the ratio of the number of samples correctly classified by the total number of samples. Higher accuracy is better.
- **Mean Accuracy (MA)**: is the average of the accuracies of many malware detectors.
- **Accuracy Reduction:** is the difference between the accuracy of a malware detector before and after the attack. It is used to evaluate the robustness of a malware detector. Lower accuracy reduction indicates that a malware detector is more immune to attacks.
- **Evasion Rate (ER)**: is the percentage of adversarial samples (generated by an attack agent) to the number of malware samples in a dataset. A high evasion rate indicates a strong adversarial attack and a weak malware detector.
- **Mean Evasion Rate (MER)**: is the average evasion rate of a group of adversarial attacks against a group of malware detectors.
- **Perturbation Application Percentage (PAP)** is the number of adversarial applications in which a particular feature was modified (perturbated) by an attack agent to the total number of adversarial applications.
- **Perturbation Frequency Percentage (PFP)**: of a feature is the percentage ratio of the number of times that feature was modified (perturbated) to the total number of perturbations during an attack.

4 Experimental Results

This section will discuss the results of all the conducted experiments. It includes the baseline models, MEA attack, ARShield defense, and MEA re-attack on ARShield models.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

4.1 Baseline Android Malware Detection Models

We trained 30 unique baseline malware detection models based on 15 classifies (from 4 families: Machine Learning Classifiers, Bagging based Classifiers, Boosting based Classifiers, Deep Learning Classifiers) and two static features (permission and intent). We measure the performance of these models using Mean Accuracy (MA) and Area under the ROC Curve (AUC). The Mean Accuracy (MA) of 93.28% was achieved by the 15 permission based malware detection models. The maximum MA of 94.27% was obtained by 4 DNN models and the minimum MA of 92.26% by 4 Boosting Models. The average AUC for 15 permission based models was 0.93. On the other hand, the MA of 78.34% was obtained by the 15 intent based malware detection models. The highest and lowest MA of 78.73% and 78.07% were obtained by 4 DNN and 4 Boosting models, respectively. We obtained a mean AUC of 0.78 for the 15 intent based malware detection models. We observe that the permission based malware detection models perform much better than the intent based detection models. The mean accuracy and AUC of all 30 malware detection models were 85.81% and 0.86, respectively.

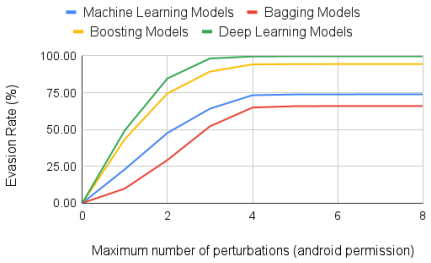
4.2 Adversarial Attack on Malware Detection Models

The MEA agents (A2C-MEA, TRPO-MEA and PPO-MEA) modify the malware samples (by adding perturbations) such that the generated adversarial samples can evade the malware detection models. The alterations made in the attack add perturbations (permissions and intents) while preserving the syntactic and semantic integrity of the modified application. The attack is designed to minimize the alterations, while their count is limited to 5 for permission models and 4 for intent models. We use metrics such as the Evasion Rate (ER), Mean Evasion Rate (MER), and change in Mean Accuracy (MA) to evaluate the attack strategies.

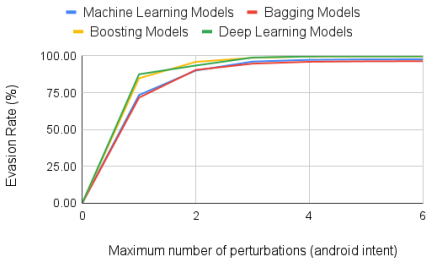
4.2.1 Evasion Rate @ MEA Attack: Evasion Rate (ER) is defined as the percentage of malware applications (which are classified as malware by the baseline models) that are successfully converted to adversarial applications by introducing alterations/perturbations such that they are forcefully misclassified as benign by the same baseline models. MER is the mean evasion rate across classifier families or features. Figure 3 plots the MER over the number of alterations by the 3 MEAs (A2C-MEA, TRPO-MEA, and PPO-MEA) against different families of malware detection models.

During the *A2C-MEA attack*, we observed an MER of 84.57% against the 15 permission based models while making just 2.20 perturbations on average. While against the 15 intent based models, the MER obtained was 98.22% with 1.27 mean perturbations. We record the maximum MER in deep learning models, 99.68% on permission based models, and 99.62% on intent based models. The *TRPO-MEA attack* obtained an MER of 89.44% using 2.52 mean perturbations against 15 permission based models. The highest MER of 99.65% was seen in 4 DL models. The MER obtained against 15 intent models was 97.10% using average 1.38 perturbations. Boosting models showed the maximum MER of 98.65%.

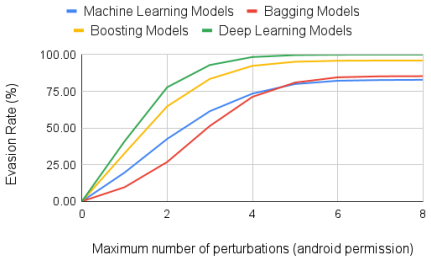
ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6



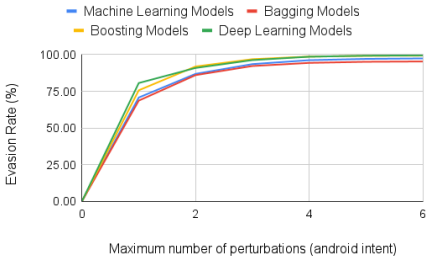
(a) **Mean Evasion Rate** of **Permission** based Malware Detection Models with **A2C-MEA**



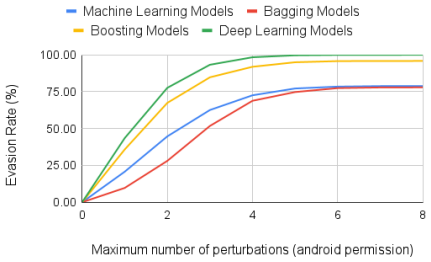
(b) **Mean Evasion Rate** of **Intent** based Malware Detection Models with **A2C-MEA**



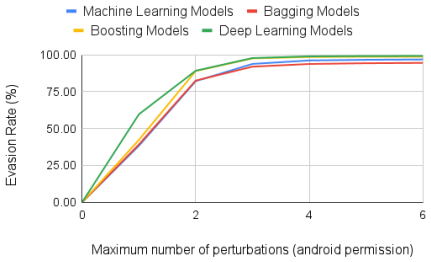
(c) **Mean Evasion Rate** of **Permission** based Malware Detection Models with **TRPO-MEA**



(d) **Mean Evasion Rate** of **Intent** based Malware Detection Models with **TRPO-MEA**



(e) **Mean Evasion Rate** of **Permission** based Malware Detection Models with **PPO-MEA**



(f) **Mean Evasion Rate** of **Intent** based Malware Detection Models with **PPO-MEA**

Fig. 3: Performance of **Malware Evasion Attack** (A2C-MEA, TRPO-MEA and PPO-MEA) against different android malware detection models.

Finally, in the case of *PPO-MEA attack*, we achieved an MER of 87.36% and 96.99% against 15 permission and 15 intent based models with just 2.42 and 1.69 perturbations, respectively. We again observe the highest MER on deep learning models. We can see that the MER in intent based models is much higher

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

compared to the permission based models, showing us that intent based models are more vulnerable to MEA attacks. Finally, the A2C-MEA, TRPO-MEA, and PPO-MEA achieved an MER of 91.40%, 93.27%, 92.17%, respectively, against 30 malware detection models.

4.2.2 Accuracy Reduction @ MEA Attack After a successful malware evasion attack, a drop in accuracy is expected due to a high number of forced misclassifications. Figure 4 shows the accuracies of the different families of baseline malware detection models (blue bar) and after the MEA attack on baseline models (red bars). The *A2C-MEA attack* on 15 permission based models dropped the MA from 93.28% to 55.89%. DL models show the highest drop of 44.93%. The MA for 15 intent based models dropped from 78.34% to 46.55%, with 4 boosting models showing the highest drop of 32.38%. In the *TRPO-MEA attack*, the MA of the 15 permission based models reduced from 93.28% to 53.66%. Maximum drop in DL models (44.91%). While in the 15 intent based models, the MA dropped to 46.92% from 78.34%. Finally, the *PPO-MEA attack* dropped the MA from 93.28% to 54.62% in 15 permission based models and from 78.34% to 46.96% in 15 intent based models. We observe the highest drop in DL models in both permissions (44.89%) and intent (32.07%) models. We observe that the reduction in accuracy is greater in permission based models as compared to intent based models. Finally, the A2C-MEA, TRPO-MEA, and PPO-MEA reduced the MA from 85.81% to 51.22%, 85.81% to 50.29% and 85.81% to 50.79%, respectively, in 30 malware detection models.

4.2.3 Vulnerability List Table 2 lists the top 5 most pertubated android permissions and android intents by the three MEA agents (A2C-MEA, TRPO-MEA, and PPO-MEA). They are evaluated using the metrics Perturbation Application Percentage (PAP) and Perturbation Frequency Percentage (PFP).

The *android.permission.USE_CREDENTIALS* and *android.permission.READ_CALL_LOG* are the two most pertubated android permissions, being added in more than 60% of applications. On the other hand, the intent *android.intent.action.MY_PACKAGE_REPLACED* is modified in more than 70% of applications making it the most pertubated android intent.

Out of all the perturbations made during the attacks on permission based classifiers, *android.permission.USE_CREDENTIALS* and *android.permission.READ_CALL_LOG* were around 50% of them. For A2C-MEA, these 2 had a combined PFP of 63.41%. Whereas, out of all intents, *android.intent.action.MY_PACKAGE_REPLACED* had the highest PFP. It contributed to more than 70% in A2C-MEA and TRPO-MEA.

4.3 ARShield Defense Strategy

The final stage of the proposed framework is the ARShield defense strategy. We measure the improvements in the detection and robustness performance of the ARShield models over the baseline models in terms of accuracy, evasion rate, and change in accuracy of the models after MEA attacks.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

Table 2: List of top android permissions and android intents that are most perturbed during A2C-MEA, TRPO-MEA and PPO-MEA attacks on malware detection models.

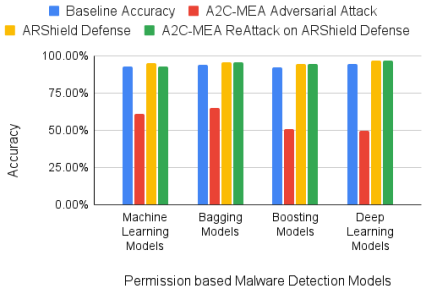
Perturbation	Name	Frequency Percentage	Sample Percentage
A2C-MEA			
Android Permission (Maximum 5 perturbations)	android.permission.USE_CREDENTIALS	32.41	73.69
	android.permission.READ_CALL_LOG	31.00	70.48
	android.permission.READ_PROFILE	19.28	43.84
	android.permission.CAMERA	15.65	35.58
	android.permission.GET_ACCOUNTS	0.52	1.18
Android Intent (Maximum 4 perturbations)	android.intent.action.MY_PACKAGE_REPLACED	55.80	70.67
	android.intent.action.MEDIA_BUTTON	33.00	41.79
	android.intent.action.SEND	5.98	7.57
	android.intent.action.TIMEZONE_CHANGED	2.46	3.12
	android.intent.category.DEFAULT	1.38	1.74
TRPO-MEA			
Android Permission (Maximum 5 perturbations)	android.permission.USE_CREDENTIALS	26.22	68.48
	android.permission.READ_CALL_LOG	23.19	60.58
	android.permission.READ_PROFILE	19.04	49.73
	android.permission.CAMERA	11.56	30.19
	android.permission.GET_ACCOUNTS	8.28	21.63
Android Intent (Maximum 4 perturbations)	android.intent.action.MY_PACKAGE_REPLACED	56.78	78.58
	android.intent.action.MEDIA_BUTTON	15.59	21.57
	android.intent.action.SEND	8.65	11.98
	android.intent.action.TIMEZONE_CHANGED	4.23	5.85
	android.intent.category.MONKEY	2.84	3.94
PPO-MEA			
Android Permission (Maximum 5 perturbations)	android.permission.USE_CREDENTIALS	27.37	67.98
	android.permission.READ_CALL_LOG	25.40	63.07
	android.permission.READ_PROFILE	20.96	52.06
	android.permission.GET_ACCOUNTS	10.51	26.10
	android.permission.CAMERA	9.53	23.66
Android Intent (Maximum 4 perturbations)	android.intent.action.MY_PACKAGE_REPLACED	40.76	68.24
	android.intent.category.BROWSABLE	12.48	20.89
	android.intent.action.PACKAGE_REMOVED	10.77	18.02
	android.intent.action.TIMEZONE_CHANGED	9.24	15.47
	android.intent.action.MEDIA_BUTTON	8.58	14.36

4.3.1 Detection Performance: Figure 4 shows the accuracies of the 4 families of malware detection models at various stages of the RL-MAGE framework. On applying ARShield defense using the A2C-MEA adversarial samples, the Mean Accuracy (MA) of permission based models increased from 93.28% to 95.39%, an improvement of 2.11%, and the MA of intent based models jumped from 78.34% to 84.38%, which is an improvement of 6.04%. We see maximum improvement in MA for boosting models, of 2.41% and 6.22% in permission and intent based models respectively.

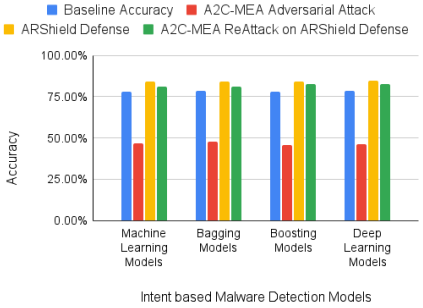
On applying the TRPO-ARShield, we observe a boost in the mean accuracy of the permission models by 2.00%, from 93.28% to 95.28%. Maximum improvement of 2.23% is shown by bagging Classifiers. In the case of intent based models, we observe a more significant increase of 6.19%, from 78.34% to 84.53%. Deep learning classifiers showed the most improvement of 6.37%.

Similarly, the PPO-ARShield improved the MA of permission and intent models from 93.28% to 95.44% and 78.34% to 84.22%, which is an improvement

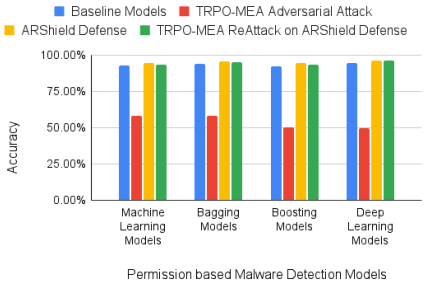
ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6



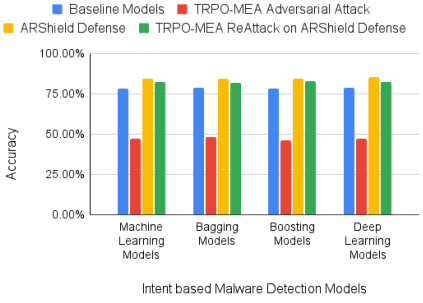
(a) **Performance** of **Permission** based Malware Detection Models against **A2C-MEA**



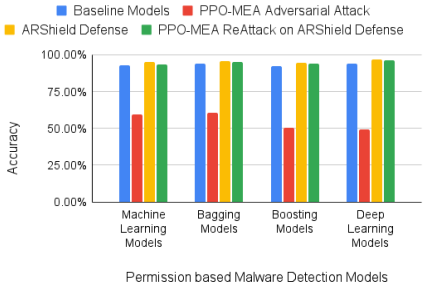
(b) **Performance** of **Intent** based Malware Detection Models against **A2C-MEA**



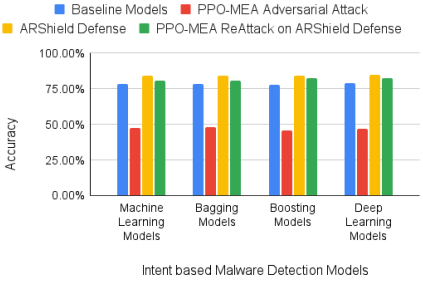
(c) **Performance** of **Permission** based Malware Detection Models against **TRPO-MEA**



(d) **Performance** of **Intent** based Malware Detection Models against **TRPO-MEA**



(e) **Performance** of **Permission** based Malware Detection Models against **PPO-MEA**



(f) **Performance** of **Intent** based Malware Detection Models against **PPO-MEA**

Fig. 4: Performance of different android malware detection models against A2C-MEA, TRPO-MEA and PPO-MEA

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

of 2.16% and 5.88%, respectively. Here too, we notice boosting models improving the most. MA of permission and intent based models improved by 2.33% and 6.03% respectively. From the data, we observe that the ARShield defense positively impacts the malware detection capabilities of the models, leading to an overall accuracy improvement of 4.06% across all models. Out of the four families, boosting classifiers benefit the most with ARShield defence.

4.3.2 Robustness Performance: Apart from improving the malware detection performance of the malware detection models in normal circumstances, the ARShield defense also improves the adversarial robustness of the models to help them defend against adversarial attacks.

In the case of A2C-MEA reattack on 15 permission ARShield models, we observe an MER of 1.27% with a reduction in MA from 95.39% to 94.69% (0.7% drop). Whereas in 15 intent ARShield models, we record a drop in MA of 2.28%, from 84.38% to 82.10%, and an MER of 1.38%. Performing the TRPO-MEA attack on the 15 permission ARShield models leads to a drop in MA from 95.28% to 94.48%, which is a dip of just 0.80%. The MER obtained is 1.25%. On the other hand, in 15 intent based ARShield models, we get an MER of 0.84% with a 2.40% drop in MA from 84.53% to 82.14%. In the case of the PPO-MEA attack on permission ARShield models, we get an MER of 1.14%, reducing the MA from 95.44% to 94.72% (0.72% drop). The same attack on intent ARShield models resulted in an MA drop from 84.22% to 81.72% (2.50% drop) and an MER of 1.66%.

Permisison based ARShield models of basic machine learning category show the highest MER during reattack. We get 2.34%, 1.20%, and 1.74% MER with A2C-MEA, TRPO-MEA and PPO-MEA respectively. On the other hand, intent based ARShield models of bagging category show the highest MER during reattack. We observe 3.14%, 3.00% and 3.67% with the above 3 MEAs respectively. The above results show that the newly enforced ARShield models are much more effective and robust against adversarial attacks.

5 Related Work

Since Papernot et al. in 2016 highlighted that deep neural networks are vulnerable to adversarial attacks, there has been a lot of focus on adversarial attacks on neural networks [12]. However, attacks on applications of traditional machine learning have not received similar attention as image classification problems. With billions of users using android daily, the platform's security against malware is of high priority. Researchers like Taheri et al., Rathore et al., and Grosse et al. came up with novel adversarial attacks like generative adversarial network (GAN), single policy attack (SPA), and adversarial sample crafting on android malware detectors and achieved good results [23, 20, 16, 10]. But these attacks were proposed for white box scenarios which are not representative of the real world. Greybox and black box scenarios more closely represent the real world.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

Researchers have extensively studied adversarial attacks in the image classification application on grey and black box environments. In 2021, Sinha et al. fooled DNNs by modifying just 1 pixel in sample images with a mean evasion rate of 48.33% using the differential evolution attack in a grey box scenario [22]. However, there is still a lack of research on adversarial attacks on android malware detection in a partial or no-knowledge environment. Rathore et al. proposed a multi-policy attack that attacked android permission based malware detectors in a grey box scenario but achieved an evasion rate of just 53.20% across 8 machine learning models [14]. In 2021, Rathore et al. used GAAN to achieve a evasion rate of 94.69% but with 10 bit alterations [16]. When it comes to black box scenarios, Bostani et al. developed EvadeDroid achieved 81.07% evasion rate but used 8 android features and 52.48 ± 29.45 alterations [8]. Using ShadowDroid, Zhang et al. claimed a 100% evasion rate using 5.86 mean alterations [24]. But they used 8 different types of android features and studied just one classifier (SVM) without focusing on the generality of the attack which is very important in a partial knowledge scenario where the nature of the classifier is unknown.

6 Conclusion

In this work, we propose a novel RL-MAGE framework to improve the classification performance and robustness of android malware detectors. We designed three reinforcement learning agents, A2C-MEA, TRPO-MEA, and PPO-MEA, for evasion attacks and the ARShield defense strategy to improve malware detectors. The 30 baseline malware detectors achieved mean accuracy of 85.81%. Out of the three on-policy reinforcement learning algorithms, we observe that TRPO is the best algorithm for evasion as well as defense task. TRPO-MEA achieved the highest evasion rate of 93.27% with 1.95 mean alterations in 30 malware detectors. The TRPO-ARShield models also achieved the highest mean accuracy of 89.91% while displaying the lowest evasion rate of just 1.05% during re-attack.

Acknowledgement

One of the authors Dr. Sanjay K. Sahay is thankful to Data Security Council of India for financial support to work on the Android malware detection system.

References

1. More than one billion Android devices at risk of malware threats. Available: https://www.which.co.uk/news/article/more-than-one-billion-android-devices-at-risk-of-malware-threats-aXtug2P0ET0d
2. Android Statistics. Available: https://www.businessofapps.com/data/android-statistics/ (2023)
3. Apktool. Available: https://ibotpeaches.github.io/Apktool/ (2023)
4. Google I/O. Available: https://io.google/2021/program/content/?lng=en (2023)
5. Google Play Store. Available: https://play.google.com/store/ (2023)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6

6. VirusTotal. Available: https://www.virustotal.com/gui/home/upload (2023)
7. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K.: Drebin: Effective and explainable detection of android malware in your pocket. In: Network and Distributed System Security Symposium (NDSS). vol. 14, pp. 23–26 (2014)
8. Bostani, H., Moonsamy, V.: Evadedroid: A practical evasion attack on ml for black-box android malware detection. arXiv preprint arXiv:2110.03301 (2021)
9. Demetrio, L., Coull, S.E., Biggio, B., Lagorio, G., Armando, A., Roli, F.: Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. ACM Transactions on Privacy and Security (TOPS) **24**(4), 1–31 (2021)
10. Grosse, K., Papernot, N., Manoharan, P., Backes, M., McDaniel, P.: Adversarial perturbations against deep neural networks for malware classification. arXiv preprint arXiv:1606.04435 (2016)
11. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning (ICML). pp. 1928–1937 (2016)
12. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy (IEEE EuroS&P). pp. 372–387. IEEE (2016)
13. Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S., Xiang, Y.: A survey of android malware detection with deep neural models. ACM Computing Surveys (CSUR) **53**(6), 1–36 (2020)
14. Rathore, H., Sahay, S.K., Nikam, P., Sewak, M.: Robust android malware detection system against adversarial attacks using q-learning. Information Systems Frontiers pp. 1–16 (2021)
15. Rathore, H., Sahay, S.K., Rajvanshi, R., Sewak, M.: Identification of significant permissions for efficient android malware detection. In: International Conference on Broadband Communications, Networks, and Systems. pp. 33–52. Springer (2021)
16. Rathore, H., Samavedhi, A., Sahay, S.K., Sewak, M.: Robust malware detection models: learning from adversarial attacks and defenses. Forensic Science International: Digital Investigation **37**, 301183 (2021)
17. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: ICML. pp. 1889–1897 (2015)
18. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
19. Sewak, M., Sahay, S.K., Rathore, H.: Deepintent: implicitintent based android ids with e2e deep learning architecture. In: IEEE 31st PIMRC. pp. 1–6. IEEE (2020)
20. Sewak, M., Sahay, S.K., Rathore, H.: Value-approximation based deep reinforcement learning techniques: An overview. In: 2020 IEEE 5th international conference on computing communication and automation (ICCCA). pp. 379–384. IEEE (2020)
21. Sewak, M., Sahay, S.K., Rathore, H.: DRLDO: A Novel DRL based De-obfuscation System for Defence Against Metamorphic Malware. Defence Science Journal **71**(1), 55–65 (2021)
22. Sinha, S., Saranya, S.: One pixel attack analysis using activation maps. Annals of the Romanian Society for Cell Biology pp. 8397–8404 (2021)
23. Taheri, R., Javidan, R., Shojafar, M., Vinod, P., Conti, M.: Can machine learning model with static features be fooled: an adversarial machine learning approach. Cluster Computing **23**, 3233–3253 (2020)
24. Zhang, J., Zhang, C., Liu, X., Wang, Y., Diao, W., Guo, S.: ShadowDroid: Practical Black-box Attack against ML-based Android Malware Detection. In: International Conference on Parallel and Distributed Systems. pp. 629–636. IEEE (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_6