Multi-view Representation Learning from Malware to Defend Against Adversarial Variants

James Lee Hu*
Department of Management Information Systems
University of Arizona
Tucson, USA
jameshu@arizona.edu

Mohammadreza Ebrahimi*
School of Information Systems and Management
University of South Florida
Tampa, USA
ebrahimim@usf.edu

Weifeng Li
Department of Management Information Systems
University of Georgia
Athens, USA
weifeng.li@uga.edu

Xin Li*
Department of Computer Science
University of Arizona
Tucson, USA
xinli2@arizona.edu

Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, USA
hsinchun@arizona.edu

Abstract—**Deep learning-based adversarial malware detectors have yielded promising results in detecting never-before-seen malware executables without relying on expensive dynamic behavior analysis and sandbox. Despite their abilities, these detectors have been shown to be vulnerable to adversarial malware variants - meticulously modified, functionality-preserving versions of original malware executables generated by machine learning. Due to the nature of these adversarial modifications, these adversarial methods often use a *single view* of malware executables (i.e., the binary/hexadecimal view) to generate adversarial malware variants. This provides an opportunity for the defenders (i.e., malware detectors) to detect the adversarial variants by utilizing more than one view of a malware file (e.g., source code view in addition to the binary view). The rationale behind this idea is that while the adversary focuses on the binary view, certain characteristics of the malware file in the source code view remain untouched which leads to the detection of the adversarial malware variants. To capitalize on this opportunity, we propose Adversarially Robust Multiview Malware Defense (ARMD), a novel multi-view learning framework to improve the robustness of DL-based malware detectors against adversarial variants. Our experiments on three renowned open-source deep learning-based malware detectors across six common malware categories show that ARMD is able to improve the adversarial robustness by up to seven times on these malware detectors.**

Index Terms—**Multi-View Learning, Adversarial Machine Learning, Adversarial Malware Variants, Deep Learning-based Malware Detectors, Adversarial Robustness**

I. INTRODUCTION

Recent studies have shown deep Learning (DL)-based malware detectors are susceptible to attacks from Adversarial Malware Generation (AMG) techniques [1] [3] [4]. These AMG methods automatically generate adversarial malware variants to evade a targeted malware detector. The generated malware samples can be used to improve a malware detector's robustness against such attacks [2], [4]. Most AMG methods employ additive modifications via injecting bytes to the malware binary to generate functionality-preserving evasive variants [11]. However, these AMG methods are often limited to operating on the binary view of a malware sample and do not account for other views representing malware samples (e.g., a malware's source code). Thus, the multi-view (MV) nature of malware could be leveraged to improve detector robustness against these adversarial variants. As such, we expect MV learning to boost adversarial robustness.

MV learning refers to a branch of machine learning models that processes multiple distinct representations from the same instance of input data [13]. When applied to malware detection, MV learning has been shown to significantly improve malware detection accuracy [14] [15]. These models often leverage a mechanism to combine different views into a single representation, known as a fusion mechanism [16]. However, the impact of MV learning, and its different fusion mechanisms, on detector models' adversarial robustness is unclear. We hypothesize that MV learning can detect adversarial variants evasive to single-view detectors by extracting features from malware views untouched by AMG methods. Thus,

in this study, we propose Adversarially Robust Multiview Malware Defense (ARMD), an MV learning framework to improve the robustness of DL-based malware detectors against adversarial variants.

In the remainder of this manuscript, first, we review AMG, DL-based malware detectors, MV learning, fusion mechanisms, and highway layers. Subsequently, we detail the components of our proposed framework and its contribution. We then conduct several experiments to evaluate the performance of ARMD. Lastly, we highlight promising future directions.

II. LITERATURE REVIEW

Five areas of research are examined. First, we review extant AMG studies as the overarching area for our study. Second, we examine DL-based Malware Detectors as an effective type of AI model to detect malicious samples. Third, we review MV Learning as a potential way to boost a DL-based detector's adversarial robustness. Fourth, we investigate Fusion Mechanisms to determine their impact on an MV Learning model's adversarial robustness. Lastly, we review Highway Layers as a potential remedy for the shortcomings of existing fusion mechanisms regarding adversarial robustness.

A. Adversarial Malware Generation (AMG)

AMG aims to perturb malware samples and generate variants that evade malware detectors. Among the prevailing AMG methods, append attacks (considered as additive modifications) are the most practical due to their high chance of preserving the functionality of the original malware executable [11]. We summarize selected significant append-based prior work based on their data source, attack method used, and the view(s) of the malware sample they operate under in Table I.

Three major observation are made from Table I. First, the majority of studies use VirusTotal, an open-source online malware database, as a source of their malware samples [1] [2] [3] [4] [5] [6] [8] [10] [11] [12]. Second, regarding selected attack methods, a few notable attack methods include simple append attack [11], attacking using randomly generated perturbation [5], and attacking using specific perturbations that lowers a malware detector's score [6]. More advanced methods incorporate machine learning techniques (Genetic Programming [2] [7], Gradient Descent [5], and Dynamic Programming [9]) and implement advanced DL-based techniques (Generative Adversarial Networks [10], Deep Reinforcement Learning [1]

[8] [12], and Generative Recurrent Neural Networks [4] [3]). Third, and most importantly, most AMG methods only operate within a single view of the malware. Many of these AMG methods operate in the binary view [2] [3] [4] [5] [6] [7] [8] [11]. A few studies delved into AMG attacks on other views (e.g., the source code view [9] and API call view [9]). The main exceptions are two Deep RL-based AMG studies [1] [12]. These two studies include multiple different perturbations in their RL action space, a few of which results in a simultaneous binary and source code edit. Overall, we observe that most AMG studies only operate within a single view of the malware. As such, when attacking an MV malware detector, these AMG methods are expected to be rendered ineffective due to their perturbations only affecting certain parts of the malware detector's input.

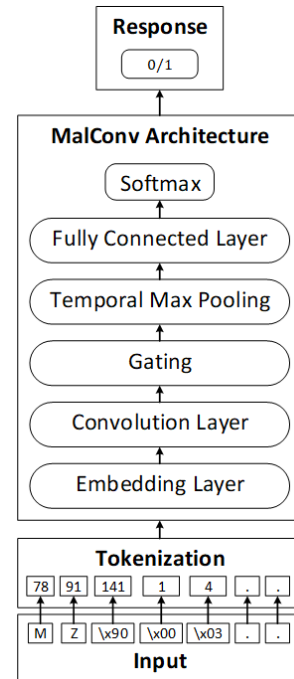### B. Deep Learning-based (DL-based) Malware Detectors

Fig. 1. MalConv Architecture

DL-based malware detectors have shown high performance in malware categorization [17] [18] [19]. One such well-known detector is MalConv, a widely-used open-source DL-based malware detector operating only in the binary view of the

TABLE I
SELECTED SIGNIFICANT PRIOR RESEARCH ON AMG APPEND ATTACKS AGAINST MALWARE DETECTORS

Year	Author(s)	Data Source	Attack Method	View
2021	Ebrahimi et al. [1]	VirusTotal	Deep RL	Binary & Source Code
2021	Demetrio et al. [2]	VirusTotal	Genetic programming	Binary
2021	Hu et al. [3]	VirusTotal	GPT2	Binary
2020	Ebrahimi et al. [4]	VirusTotal	Generative RNN	Binary
2019	Castro et al. [5]	VirusTotal	Random perturbations	Binary
2019	Chen et al. [6]	VirusShare, Malwarebenchmark	Enhanced random perturbations	Binary
2019	Dey et al. [7]	Contagio PDF malware dump	Genetic programming	Binary
2019	Fang et al. [8]	VirusTotal	Deep RL	Binary
2019	Park et al. [9]	Malmig & MMBig	Dynamic programming	Source Code
2019	Rosenberg et al. [10]	VirusTotal	GAN	API Call
2019	Suciu et al. [11]	VirusTotal, Reversing Labs, FireEye	Append attack	Binary
2018	Anderson et al. [12]	VirusTotal	Deep RL	Binary & Source Code

Note: RNN: Recurrent Neural Network; NN: Neural Network; GAN: Generative Adversarial Network; RL: Reinforcement Learning

malware [19]. MalConv's architecture utilizes a Convolutional Neural Network (CNN) architecture to directly process a malware sample's byte data (Figure 1). Specifically, MalConv takes the first 2 million bytes of a malware executable as input, passing the data through its tokenization and embedding for data pre-processing. Subsequently, the information is passed through a convolution and a max pooling layer to extract features from the malware binary regardless of location. A fully connected layer then interprets extracted features and performs classification. Lastly, a softmax layer returns a score mapped to a final result from 0,1 indicating benign or malicious. Due to its widespread popularity in malware academia as both a reference and a benchmark method, MalConv is considered representative of most open-source DL-based malware detectors [1] [4] [3] [20] [21] [11].

C. Multi-View (MV) Learning

MV learning refers to learning from multiple distinct representations of the same instance of data [13]. It has emerged as a viable way to enhance deep learning performance [14]. MV malware detectors can extract salient features available in certain views, allowing them to learn a more accurate representation of the malware data [15]. As such, MV learning has yielded promising improvements in advanced malware classification tasks [15] [14]. MV malware detectors require adversaries to attack multiple views to achieve performance comparable to attacks against single-view models [22]. Since recent AMG techniques focus on single-view operations, We expect that MV learning can potentially enhance a malware detector's adversarial robustness.

D. Fusion Mechanisms

MV learning models combine different views into a single representation for processing. Most models achieve this through a fusion mechanism [16]. Advanced fusion mechanisms can also control the flow of information during fusion, improving performance.

Two dominant fusion mechanisms exist in literature: concatenation-based and attention-based fusion. Concatenation is the most basic fusion mechanism and simply combines views additively without further processing [23]. Attention-based fusion utilizes an attention mechanism to determine the importance of different areas of the input as it combines views (Figure 2) [24].
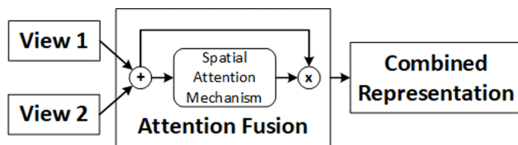

Fig. 2. Attention Fusion Basic Architecture

Unfortunately, attention mechanisms can suffer from overfitting during training, attending to specific areas of the input (i.e., generating irrelevant contexts) [25]. This may result in poor performance in adversarial robustness when attended areas are perturbed in an adversarial attack.

E. Highway Layer

The shortcomings of attention-based fusion mechanisms can be fixed using gates to filter irrelevant contexts [25]. One such renowned gating method is the Highway Layer (Figure 3) [26].
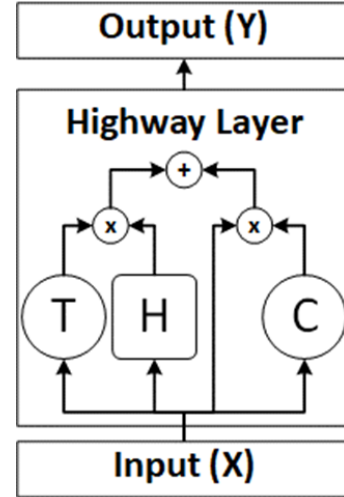

Fig. 3. Highway Layer Architecture

The Highway Layer operates using two gates for information control: the Transformation (T) and the Carry (C) Gates. It is defined using the following equation:

$$Y = T(X, W_T) * H(X, W_H) + X * C(X)$$

The Transformation Gate T determines how much of input X is subjected to nonlinear transformation H while the Carry Gate C determines how much of input X is allowed through the layer as is.

Highway Layer-based Networks have yielded significant performance in classification tasks, showing their ability to effectively control information flow [27] [28]. However, their use as fusion mechanisms for MV learning is understudied.

III. RESEARCH GAPS AND QUESTIONS

Based on our literature review, two research gaps are identified. First, within the MV-learning domain, while MV learning has been shown to improve malware classification accuracy, its impact on a model's adversarial robustness is unclear. Second, regarding the methodology, while highway layers has shown promising performance in natural language processing tasks, it is unclear how it could be applied in malware analysis to enhance adversarial robustness of malware detectors against AMG. To address the identified gaps, the following research question is posed:

- How can we leverage MV learning to create enhanced malware detectors that are robust against adversarial variants?

IV. RESEARCH DESIGN

Motivated by our research question, we propose an Adversarially Robust Multi-view Malware Defense (ARMD), an MV learning framework to improve the adversarial robustness of malware detectors.
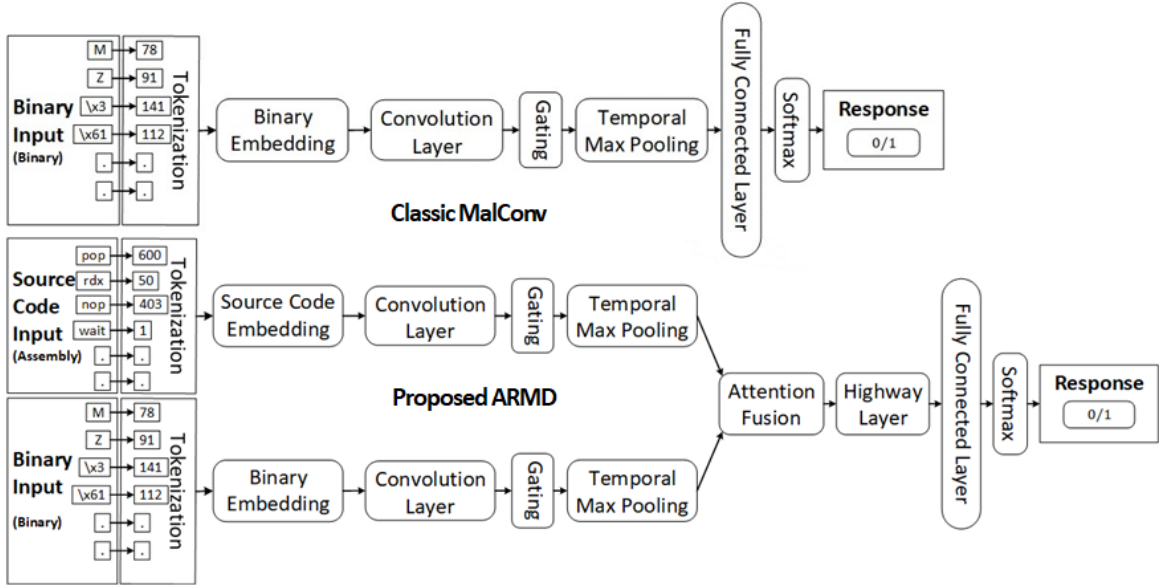
Fig. 4. ARMD's Architecture vs. MalConv Architecture

Following previous AMG studies, we first introduce the threat model that our proposed ARMD will defend against. Next, we introduce the overall architecture of ARMD. Then, we highlight ARMD's novelty, its composite fusion mechanism, in detail. Finally, we present the dataset we use to train and evaluate ARMD.

A. Threat Model

A threat model is a systematic representation of cyber attacks [29] [30]. To assess the performance of ARMD against a realistic adversary, our threat model focuses on a black-box setting. Accordingly, three major components of our threat model are:

- **Adversary's Goal:** Evade ARMD through the generation of adversarial malware samples.
- **Adversary's Knowledge:**
  - Structure and parameters of ARMD are unknown to the attacker.
  - Attacker does not have access to the confidence score produced by ARMD (fully black-box attack).
- **Adversary's Capability:** The adversary applies functionality-preserving modifications to malware samples.

B. ARMD Architecture

To defend against the aforementioned threat model, ARMD extends the MalConv architecture through a separate detector architecture, a composite fusion mechanism, and a softmax decision layer (Figure 4).

ARMD's separate detector architecture allows ARMD to learn representations of both views simultaneously. Its composite fusion mechanism uses an attention layer to learn contexts of both representations and a highway layer to filter out irrelevant contexts. Lastly, ARMD's softmax decision Layer makes the final decision based on fused representations, analogous to MalConv's softmax layer. Our intuition here is that through the gating mechanisms of the highway layer, ARMD can filter out irrelevant contexts generated by the attention fusion, thus increasing adversarial robustness.

C. ARMD Novelty: Composite Fusion Mechanism

ARMD's novelty lies in the proposed composite fusion mechanism, highlighted in Figure 5. The composite fusion mechanism is detailed in the accompanying equations and is shown as follows:

$$X = (view^{SourceCode}, view^{Binary})$$

$$Y = \sigma(f^{1x1}([AvgPool(X); MaxPool(X)]))$$

$$Z = (H(Y, W_H) * T(Y, W_T) + Y * C(Y)) * X$$

Within our equations, X denotes the combined representation from both views, connected through a simple concatenation. Y denotes the contexts generated from combined representation throughuy7 the average and max pooling of our spatial attention mechanism. Finally, Z denotes the attention contexts filtered by the highway layer applied back to the original combined representation X. Our intuition here is that as the highway layer parses the generated attention contexts
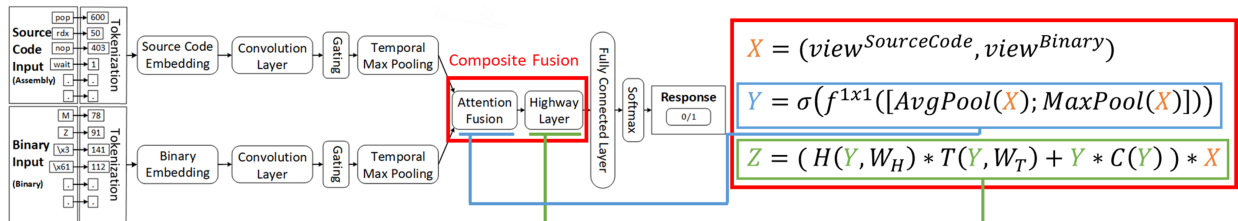

Fig. 5. Illustration of ARMD's Novelty (Composite Fusion Mechanism)

Y, the transformation gate $(T(Y, W_T))$ learns to minimize the effects of irrelevant contexts with the nonlinear transformation $(H(Y, W_H))$ while the carry gate $(C(Y))$ allows relevant contexts through as is. When the filtered contexts are applied back to the combined representation X, the resulting Z contains the fused representation with only the parts untouched by AMG methods attended to, ready to be passed into ARMD's softmax decision layer for a final classification.

D. Traning & Validation Testbed

To train and evaluate ARMD, we collected 33,853 benign and malicious files. For benign files, we collected 13,554 windows system files. For malicious files, we obtained 20,299 malware samples in eight renowned malware categories from VirusTotal, following the example of previous AMG studies [1] [2] [3] [4] [5] [6] [8] [10] [11] [12] (Table II).

TABLE II
DATASET USED FOR MODEL TRAINING

Malware Category	Description	Examples	# of Files
Adware	Shows unwanted ads and force internet traffic to sites	eldorado, razy, gator	3,627
Backdoor	Negates normal authentications for host access	lunam, rahack, symmi	2,160
Botnet	A network of bots connected through the internet	virut, salicode, sality	1,765
Dropper	Secretly installs other malwares on the host	dinwod, gepys, doboc	668
Ransom-ware	Encrypts data and files, restricting access until decrypted by malware authors	vtflooder, msil, bitman	1,422
Rootkit	Grants admin privilege to malware author	onjar, dqqd, shipup	481
Spyware	Allows malware authors to covertly steal personal data	mikey, qqpass, scar	1,172
Virus	Corrupts files on the host system	nimda, shodi, hematite	1,756
MISC	Samples that couldn't be organized into previous categories	N/A	7,248
Total	-	-	20,299

After combining and shuffling together both the benign and malicious files, we split our 33,853 file dataset into a 27,083 file training set and a 6,770 file validation set. We then used our training set to train ARMD while reserving our validation set for later evaluation.

V. EVALUATION

A. Experiment Overview

We conducted three experiments to evaluate ARMD's performance (Table III). Experiment #1 evaluates ARMD and each benchmark method in malware detection. Experiment #2 evaluates ARMD's and each benchmark method's adversarial robustness when under attack from a state-of-the-art AMG method. Finally, Experiment #3 evaluates the impact of ARMD's composite fusion mechanism on performance and adversarial robustness through an ablation analysis of ARMD with alternate fusion mechanisms.

The data and experiment parameters are different for each experiment.

TABLE III
OVERVIEW OF EXPERIMENT

Experi-ment	Goal	Data Used	Evaluation Metric
Experiment #1	Evaluate effect of MV learning on malware classification performance	Validation set from section IV-D, unseen in the training process	Recall, F1
Experiment #2	Evaluate effect of MV learning on adversarial robustness	4 prominent categories of malware samples unseen in Experiment #1 and the training process	Evasion Rate
Experiment #3	Ablation study on the effect of different fusion mechanisms on malware classification performance and adversarial robustness	Data from previous 2 experiments is used to evaluate malware classification and adversarial robustness, respectively.	F1 & Evasion Rate

B. Experiment #1 – Malware Detection Performance

To evaluate ARMD's MV learning's effect on malware classification, several renowned single-view models were selected as benchmarks (Table IV).

TABLE IV
OVERVIEW OF EXPERIMENT BENCHMARK METHODS

Method	Description	Owner	Reference(s)
MalConv	The original MalConv described from literature	Laboratory for Physical Sciences	Raff et al., 2018
NonNeg	Version of MalConv trained with non-negative weight constraints for boosted performance	Laboratory for Physical Sciences	Fleshman et al., 2019
ConvNet	DL-based CNN detector with multiple convolution layers for extensive feature extraction and learning	Avast	Krcal et al., 2018

After training ARMD, the remaining 6,770 file dataset is used as the validation set. Each benchmark is also evaluated using the validation set. For each model, accuracy, precision, recall, and F1 are recorded (Table V).

TABLE V
EXPERIMENT #1 RESULTS

Metric	MalConv	Nonneg	ConvNet	ARMD (Our Method)
Accuracy (%)	87.93%	55.23%	73.71%	**97.10%**
Precision (%)	**98.89**%	98.25%	91.72%	96.81%
Recall (%)	82.75%	33.19%	66.42%	**98.89%**
F1 (%)	90.10%	49.62%	77.05%	**97.84%**

Note: P-Values are significant at 0.05.

As seen in Table V, our results show a significant increase in ARMD's performance compared to single-view models, which is consistent with previous research. The implementation of MV architecture in a malware detector improves said detector's malware detection performance. As such, our implementation of MV architecture for a more adversarial robust malware detector will not sacrifice the detector's malware detection performance.

C. Experiment #2 – Adversarial Robustness

To evaluate ARMD's MV learning's effect on adversarial robustness, we selected a recent, high-performing Reinforcement Learning (RL) method, adapted from Ebrahimi et al. 2021 [1], as our AMG method, which includes perturbation affecting both malware binary and source code contents. We ran the RL AMG method against ARMD and our detector benchmarks. For our data, we selected a new dataset, unseen in Experiment #1, of four prominent malware categories to run the RL AMG method on (Table VI).

TABLE VI
MALWARE SAMPLES USED IN EXPERIMENT #2

Malware Category	Description	Examples	# of Files
Botnet	A network of bots connected through the internet	virut, salicode, sality	526
Ransom-ware	Encrypts data and files, restricting access and usage until decrypted by malware authors	vtflooder, msil, bitman	900
Rootkit	Grants admin privilege to malware author	onjar, dqqd, shipup	53
Spyware	Allows malware authors to steal personal information covertly	mikey, qqpass, scar	640
Virus	Corrupts files on the host system	nimda, shodi, hematite	659
Total	-	-	2,778

Table VII shows the evasion rate obtained from running the RL AMG method against each model.

TABLE VII
EXPERIMENT #2 RESULTS

Category	MalConv	Nonneg	ConvNet	ARMD (Our Method)
Botnet	68.29%	70.72%	61.60%	**5.13%**
Ransom-ware	64.28%	41.78%	**14.89%**	15.44%
Rootkit	69.23%	88.68%	56.60%	**11.32%**
Spyware	80.00%	36.93%	27.23%	**7.51%**
Virus	85.71%	96.21%	67.83%	**10.02%**
Total	73.84%	59.96%	39.94%	**10.30%**

Note: P-Values are significant at 0.05.

From our Experiment #2 results, we observe ARMD outperforming every other single-view benchmark detector in almost every category when detecting adversarial samples generated by the RL AMG method. This supports our expectation that a MV malware detector is more adversarial robust than a single-view detector. This is due to the MV detector forcing adver-

saries to conduct AMG attacks on both views simultaneously. While the chosen RL AMG method can conduct MV attacks, its simultaneous effect on both views is weaker than its effect on a single view. As such, ARMD was able to defend against the selected RL AMG method more so than our single-view benchmarks.

D. Experiment #3 – Ablation Analysis

To evaluate how different fusion mechanisms affect adversarial robustness, ARMD's fusion mechanism was modified with several alternative mechanisms (Table VIII).

TABLE VIII
OVERVIEW OF ALTERNATIVE FUSION MECHANISMS

Fusion Mechanism	Reference	Description
Concatenation	Gao et al., 2020	Source code and binary representations are concatenated
Attention	Jetley et al., 2018	An attention mechanism is used to determine the importance of different areas of the input.
Highway	Srivstava et al., 2015	Several gates are employed to control the flow of information from the input layer to the convolution layers
Highway + Attention	-	Composite fusion with a Highway Layer fusion followed by an attention layer to pass along information.
Attention + Highway (Our method)	-	Composite fusion with an attention layer fusion followed by a highway layer to pass along information.

Experiment #3 consists of two parts, malware classification performance and adversarial robustness evaluation. Data from Experiment #1 is used for malware detection performance, with model performance evaluated using F1 score. Data from Experiment #2 is used for adversarial robustness evaluation, with model performance evaluated using evasion rate. After running the RL AMG method against each fusion mechanism, their respective F1 scores and evasion rates are recorded (Table IX).

Our results show that our proposed fusion mechanism performs comparably with other alternatives in malware detection while outperforming other alternatives in adversarial robustness.

E. Experiment Result Summary

We derived three important observations from our experiment results. (1) Experiment #1 demonstrates that implementing MV learning into malware detectors increases a detector's

TABLE IX
ABLATION STUDY RESULTS

Sub-Experiment	Metric / Category	Concatenation	Attention	Highway	Highway + Attention	Attention + Highway (Our method)
Malware Detection Performance	Accuracy (%)	97.80%	**97.82%**	97.81%	97.27%	97.14%
	Precision (%)	97.94%	98.05%	97.34%	**98.27%**	96.49%
	Recall (%)	98.76%	98.69%	**99.43%**	97.61%	99.30%
	F1 (%)	98.35%	98.37%	98.37%	97.94%	97.87%
Adversarial Robustness Evaluation	Botnet	5.32%	70.34%	10.46%	10.27%	**5.13%**
	Ransomware	15.67%	18.67%	18.00%	17.78%	**15.44%**
	Rootkit	13.21%	20.75%	11.32%	18.87%	**11.32%**
	Spyware	7.82%	13.30%	8.76%	10.33%	**7.51%**
	Virus	10.17%	36.42%	**8.80%**	13.05%	10.02%
	Total	10.55%	31.47%	12.13%	13.54%	**10.30%**

Note: P-Values are significant at 0.05.

malware detection performance. (2) Experiment #2 indicates MV learning's ability to improve a model's adversarial robustness. (3) Experiment #3 shows ARMD's specific fusion mechanism (Attention + Highway) as the most adversarially robust fusion mechanism with minimal effect on the detection performance. These observations suggest that as ARMD operates, its highway layer filters out the irrelevant contexts generated by its attention layer, thus improving ARMD's adversarial robustness and supporting our initial intuitions.

VI. CONCLUSION AND FUTURE DIRECTIONS

Deep Learning (DL)-based malware detectors are susceptible to Adversarial Malware Generation (AMG) techniques. Based on the nature of these adversarial modifications, these AMG methods are often limited to operating on the binary view of a malware sample and do not account for alternative views representing the sample. Recognizing this gap as an opportunity for improving defense, in this study, we developed a novel detector architecture that considers two views of the malware, combining each representation through a novel composite fusion mechanism. Our results show that ARMD is significantly more adversarially robust against AMG attacks without sacrificing too much malware detection performance. This highlights the viability of leveraging the single-view nature of AMG techniques to boost a detector's adversarial robustness through multi-view learning. Considering our results, we recommend malware detector developers incorporate multi-view capabilities into their DL-based detectors to enhance adversarial robustness.

Based on our results, we identify three promising directions for future research. First, we envision expanding the number of views ARMD can consider (e.g., including a malware's API calls) so that ARMD can capture behavioral data about a malware file to render ARMD more invulnerable to single-view AMG attacks. Second, we believe evaluating ARMD on polymorphic AMG techniques, such as file encryption through the open-source tool darkarmour, can show how ARMD handles massive, file-wide perturbations. Lastly, implementing ARMD in a live-fire cybersecurity environment where real-world hackers can try to evade ARMD with their custom-made adversarial samples will yield interesting results on how well our model can perform in a real-world setting.

ACKNOWLEDGMENT

REFERENCES

[1] M. Ebrahimi, J. Pacheco, W. Li, J. L. Hu, and H. Chen, "Binary black-box attacks against static malware detectors with reinforcement learning in discrete action spaces," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 85–91.

[2] L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Functionality-preserving black-box optimization of adversarial windows malware," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3469–3478, 2021.

[3] J. L. Hu, M. Ebrahimi, and H. Chen, "Single-shot black-box adversarial attacks against malware detectors: A causal language model approach," in *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2021, pp. 1–6.

[4] M. Ebrahimi, N. Zhang, J. Hu, M. T. Raza, and H. Chen, "Binary black-box evasion attacks against deep learning-based static malware detectors with adversarial byte-level language model," *arXiv preprint arXiv:2012.07994*, 2021.

[5] R. L. Castro, C. Schmitt, and G. D. Rodosek, "Armed: How automatic malware modifications can evade static detection?" in *2019 5th International Conference on Information Management (ICIM)*. IEEE, 2019, pp. 20–27.

[6] B. Chen, Z. Ren, C. Yu, I. Hussain, and J. Liu, "Adversarial examples for cnn-based malware detectors," *IEEE Access*, vol. 7, pp. 54 360–54 371, 2019.

[7] S. Dey, A. Kumar, M. Sawarkar, P. K. Singh, and S. Nandi, "Evadepdf: Towards evading machine learning based pdf malware classifiers," in *International Conference on Security & Privacy*. Springer, 2019, pp. 140–150.

[8] Z. Fang, J. Wang, B. Li, S. Wu, Y. Zhou, and H. Huang, "Evading anti-malware engines with deep reinforcement learning," *IEEE Access*, vol. 7, pp. 48 867–48 879, 2019.

[9] D. Park, H. Khan, and B. Yener, "Generation & evaluation of adversarial examples for malware obfuscation," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 1283–1290.

[10] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Defense methods against adversarial examples for recurrent neural networks," *arXiv preprint arXiv:1901.09963*, 2019.

[11] O. Suciu, S. E. Coull, and J. Johns, "Exploring adversarial examples in malware detection," in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 8–14.

[12] H. S. Anderson, A. Kharkar, B. Filar, D. Evans, and P. Roth, "Learning to evade static pe machine learning malware models via reinforcement learning," *arXiv preprint arXiv:1801.08917*, 2018.

[13] S. Sun, "A survey of multi-view machine learning," *Neural computing and applications*, vol. 23, no. 7, pp. 2031–2038, 2013.

[14] A. Appice, G. Andresini, and D. Malerba, "Clustering-aided multi-view classification: a case study on android malware detection," *Journal of intelligent information systems*, vol. 55, no. 1, pp. 1–26, 2020.

[15] A. Narayanan, M. Chandramohan, L. Chen, and Y. Liu, "A multi-view context-aware approach to android malware detection and malicious code localization," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1222–1274, 2018.

[16] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 10, pp. 1863–1883, 2018.

[17] B. Cakir and E. Dogdu, "Malware classification using deep learning methods," in *Proceedings of the ACMSE 2018 Conference*, 2018, pp. 1–5.

[18] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*. IEEE, 2018, pp. 1–5.

[19] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] W. Fleshman, E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Non-negative networks against adversarial attacks," *arXiv preprint arXiv:1806.06108*, 2018.

[21] M. Krčál, O. Švec, M. Bálek, and O. Jašek, "Deep convolutional malware classifiers can learn from raw executables and labels only," 2018.

[22] X. Sun and S. Sun, "Adversarial robustness and attacks for multi-view deep models," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104085, 2021.

[23] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.

[24] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv preprint arXiv:1804.02391*, 2018.

[25] M. Anand, P. Kayal, and M. Singh, "On adversarial robustness of synthetic code generation," *arXiv preprint arXiv:2106.11629*, 2021.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Advances in neural information processing systems*, vol. 28, 2015.

[27] Y. Li, Z. Shen, J. Li, and Z. Xu, "A deep learning method based on srn-yolo for forest fire detection," in *2022 5th International Symposium on Autonomous Systems (ISAS)*. IEEE, 2022, pp. 1–6.

[28] Y. Guo, D. Zhou, W. Li, J. Cao, R. Nie, L. Xiong, and X. Ruan, "Identifying polyadenylation signals with biological embedding via self-attentive gated convolutional highway networks," *Applied Soft Computing*, vol. 103, p. 107133, 2021.

[29] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[30] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 984–996, 2013.