MG-DVD: A Real-time Framework for Malware Variant Detection Based on Dynamic Heterogeneous Graph Learning

Chen Liu[1,2] , **Bo Li**[1,2*] , **Jun Zhao**[1,2] , **Ming Su**[1,2] and **Xu-Dong Liu**[1,2†]
[1]School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China
{liuchen, libo, zhaojun, suming, liuxd}@act.buaa.edu.cn

Abstract

Detecting the newly emerging malware variants in real time is crucial for mitigating cyber risks and proactively blocking intrusions. In this paper, we propose MG-DVD, a novel detection framework based on dynamic heterogeneous graph learning, to detect malware variants in real time. Particularly, MG-DVD first models the fine-grained execution event streams of malware variants into dynamic heterogeneous graphs and investigates real-world meta-graphs between malware objects, which can effectively characterize more discriminative malicious evolutionary patterns between malware and their variants. Then, MG-DVD presents two dynamic walk-based heterogeneous graph learning methods to learn more comprehensive representations of malware variants, which significantly reduces the cost of the entire graph retraining. As a result, MG-DVD is equipped with the ability to detect malware variants in real time, and it presents better interpretability by introducing meaningful meta-graphs. Comprehensive experiments on large-scale samples prove that our proposed MG-DVD outperforms state-of-the-art methods in detecting malware variants in terms of effectiveness and efficiency.

1 Introduction

Malware has become one of the largest catastrophes endangering information systems, which consistently permeates and attacks information systems to steal sensitive information, take control of the target system, and collect ransom [Gandotra *et al.*, 2014]. Purifying the network environment and keeping information systems away from malware attacks has become a serious challenge faced by security communities and researchers [Ye *et al.*, 2019].

During the last decade, a large volume of malware detection approaches has been proposed. The existing malware detection methods can be roughly divided into two types: signature-based [Kang *et al.*, 2016; Gaviria de la Puerta and Sanz, 2017; Raff *et al.*, 2018; Zhang *et al.*, 2019; Singh and Singh, 2020] and behavior-based [Pascanu *et al.*, 2015; Sun *et al.*, 2016; Bartos *et al.*, 2016; Zhang *et al.*, 2018; Wang *et al.*, 2019b]. Particularly, signature-based methods rely on feature engineering to manually extract malware fingerprints from known samples, which hardly identify new malware variants [Ye *et al.*, 2019]. Behavior-based detection methods focus on investigating the independent API sequence rather than considering their interactive call relationships, inevitably resulting in high false positives and expensive time cost [Zhang *et al.*, 2020].

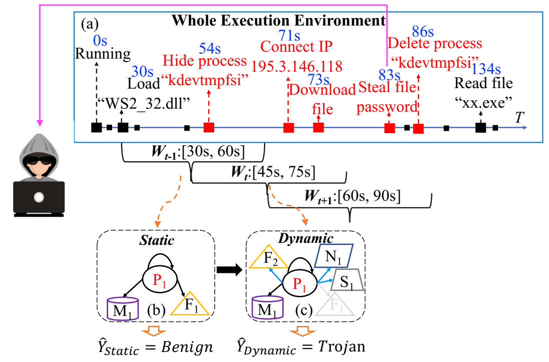
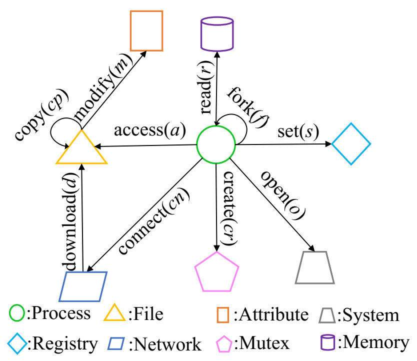


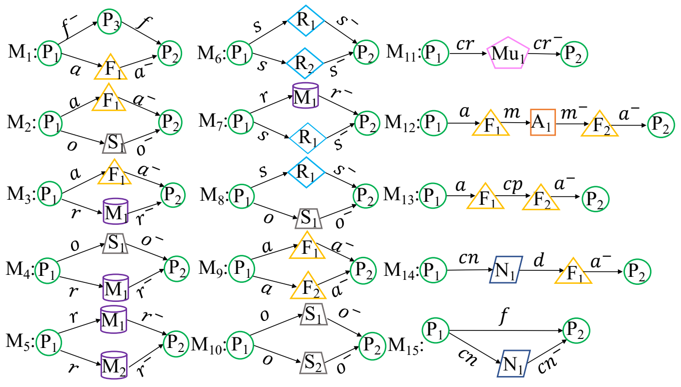
Figure 1: Illustration of dynamic malware variant detection: (a) depicts execution events stream of "*kdevtmpfsi*", in which red rectangles and black rectangles represent malicious behaviors and normal behaviors, respectively; (b) represents the static-based detection method (e.g., MatchGNet [Wang *et al.*, 2019b]), which only models the execution events at a specific time (e.g., W_{t-1}) into a static heterogeneous graph but cannot investigate newly added events in real time, result in misjudging malicious behaviors as benign. (c) is our proposed MG-DVD, which dynamically updates from (b) by adding the blue directed edges (e.g., $P_1 \rightarrow F_2$, $P_1 \rightarrow N_1$, and $P_1 \rightarrow S_1$) and removing the grey directed edges (e.g., $P_1 \rightarrow F_1$). MG-DVD can dynamically learn more comprehensive representations of malware and capture the evolutionary patterns between malware and variants to detect malicious behaviors in real time (before 75s).

Indeed, most malware will perform variants over time to evade detection, and their malicious behaviors are triggered in a very short time [Wang *et al.*, 2019b]. As shown in Figure 1, the malicious behaviors of "*kdevtmpfsi*" are submitted between 54s and 86s. However, the existing detection approaches cannot meet the dynamic detection requirements

*Corresponding Author
†Corresponding Author



(a) Network schema



(b) Meta-graphs

Figure 2: Network schema and meta-graphs for MG-DVD.

and misjudge malicious behaviors as benign. In summary, the existing malware detection approaches expose three major challenges. First, they [Wang *et al.*, 2019b; Chen *et al.*, 2018; Wu *et al.*, 2019] are inefficient to indiscriminately retrain the entire graph whenever a new node or event is injected, resulting in their incapacity of detecting malware variants in real time. Second, static random walk strategy [Wang *et al.*, 2019b; Fu *et al.*, 2017; Fan *et al.*, 2018] often generates a large search space since long-term walks for all nodes, causing an expensive and unacceptable time cost. Third, feature-based [Kang *et al.*, 2016; Gaviria de la Puerta and Sanz, 2017; Raff *et al.*, 2018; Zhang *et al.*, 2019; Pascanu *et al.*, 2015; Sun *et al.*, 2016; Bartos *et al.*, 2016; Zhang *et al.*, 2018] and metapath-based [Wang *et al.*, 2019b] detection approaches are incapable of characterizing high-order and expressive representations of malware variants.

To tackle these challenges, we present a novel MetaGraph-guided Dynamic Variant Detection (MG-DVD) framework based on dynamic heterogeneous graph learning, which models execution events stream of the target variant into a series of dynamic heterogeneous graphs and identifies the variant in a real-time and interpretable way. The contributions of this paper are summarized as follows:

- We propose a dynamic detection framework, namely MG-DVD, consisting of two encoders involving DWIUE and CHGAE. MG-DVD utilizes the overlapping information of adjacent sliding windows to dynamically generate graph embedding, which provides malware warnings in real time to keep information systems from malicious intrusions.
- MG-DVD implements an efficient dynamic walk strategy, which concentrates on the newly injected nodes in the heterogeneous graph at the target sliding window and learns their node representations by exploring the emerging meta-graphs between them. As a result, MG-DVD is able to effectively alleviate the pressure of search space and time overhead.
- MG-DVD presents remarkable interpretability by introducing the meaningful meta-graphs depicting interactive relationships between malware and corresponding variants, which can effectively learn the evolutionary patterns of variants from different malware families.
- We conduct extensive experiments on large-scale real-world samples. Experimental results verify that our proposed MG-DVD outperforms the state-of-the-art approaches in detecting new malware variants.

2 Preliminary

In this section, we recap important definitions used in our work, such as DHGS, network schema, and meta-graph.

Definition 1. *A **dynamic heterogeneous graphs sequence (DHGS)** over the execution events stream of malware variant is a graph set* $\mathbf{G}$=$\{G_1, G_2, \ldots, G_T\}$, *each* $G_t = (V_t, E_t)$ *with an entity type mapping* ϕ: $V_t \rightarrow A$ *and a relation type mapping* ψ: $E_t \rightarrow R$, *where* V_t *and* E_t *denote the entity set and the relation set of* G_t*, respectively, and A and R denote the entity type and relation type, respectively. Among them,* $|A| > 1$, $|R| > 1$. *The **network schema** [Sun and Han, 2012] of DHGS, denoted as* $T_G = (A, R)$, *is a graph with nodes as entity type from A and edges as relation type from R.*

To characterize high-order semantic information and capture the more discriminative malicious patterns of various malware types, we investigate the real-world meta-graphs for dynamic malware variants detection, defined as below:

Definition 2. *A **meta-graph** [Zhao* et al.*, 2017] M is a directed acyclic graph with a single source node* n_s *(i.e., with in-degree 0) and a single target node* n_t *(i.e., with out-degree 0), defined on a DHGS* $\mathbf{G}$ *with schema* $T_G = (A, R)$*, then a meta-graph can be defined as* M=$(V_M, E_M, A_M, R_M, n_s, n_t)$*, where* $V_M \in V$, $E_M \in E$ *are constrained by* $A_M \in A$ *and* $R_M \in R$*, respectively.*

Particularly, Figure 2(a) demonstrates the schema of MG-DVD, which consists of 8 types of entities and 10 types of relations. With the schema, we first enumerate all meta-graphs of MG-DVD, which start and end with processes (i.e., P_1 and P_2). Then the frequency of each meta-graph is measured on large-scale samples, and the final meta-graphs are obtained,

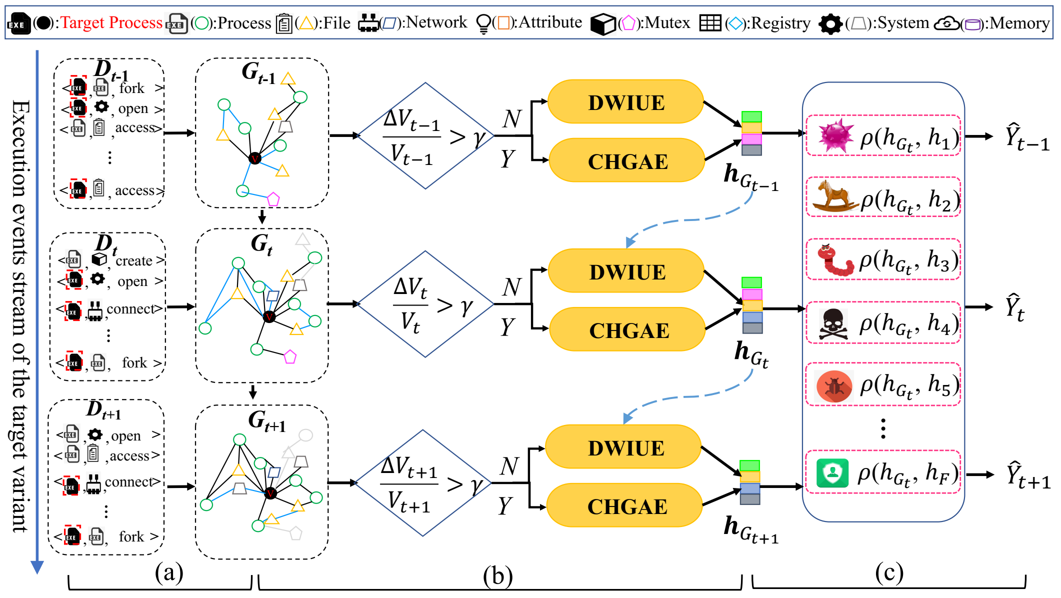


Figure 3: Framework of MG-DVD: (a) Dynamic heterogeneous graphs constructing: G_t is updated based on G_{t-1} and D_t, in which the grey nodes and grey solid lines indicate the expired events, and blue solid lines indicate newly-created events (Section 3.1); (b) Dynamic graph learning: generate the graph embedding $\boldsymbol{h}_{G_t}$ under each sliding window by encoder DWIUE or CHGAE based on the ratio $\Delta V_t / V_t$ (Section 3.2); (c) Real-time detection: calculate the Pearson correlation coefficients between the graph embedding of the target variant and the graph embeddings of all samples to obtain real-time prediction result $\widehat{Y}_t$ (Section 3.3).

as shown in Figure 2(b), which hold high-frequency and rich semantics depicting the evolutionary patterns between malware and their variants.

3 Framework

Figure 3 shows the framework of MG-DVD, which consists of three major components, detailed below.

3.1 Dynamic Heterogeneous Graphs Constructing

MG-DVD models the execution events stream of the malware variant into a *DHGS*=$\{G_1, G_2, \ldots, G_T\}$ with different sliding windows [Wang *et al.*, 2019a]. As shown in Figure 3, G_t can be updated from G_{t-1} as the new execution events are joined, and expired events are removed, and then the heterogeneous graph G_t at W_t is obtained, represented as adjacency matrix A_t. MG-DVD utilizes overlapping information at adjacent sliding windows (e.g., G_{t-1} and G_t) to receive richer variant evolution behaviors rather than reconstructing the entire graph under each sliding window, which offers better efficiency. Experimentally, the detection performance depends on the sliding window size (i.e., W) and sliding step (i.e., p), and it is proved that MG-DVD could reach the desired performance when W is set to 60s and p=1/2W.

3.2 Dynamic Graph Learning

Unlike the static frameworks [Chen *et al.*, 2018; Hamilton *et al.*, 2017; Wu *et al.*, 2019], MG-DVD aims to implement a dynamic malware variants detection based on the overlapping information of heterogeneous graphs under adjacent sliding windows. To this end, MG-DVD proposes Dynamic Walk Incremental Updating Encoder (DWIUE) and Combined Hierarchical Graph Attention Encoder (CHGAE), which can discriminatively update or retrain the graph embedding at each target sliding window based on the threshold $\Delta V_t/V_t$. Particularly, given G_t and ΔV_t at sliding window W_t, the graph embedding of G_t can be generated under the two paradigms. On the one hand, if $\Delta V_t/V_t \leq \gamma$, MG-DVD utilizes DWIUE to straightly assemble the representation matrix based on the strongly correlated $\boldsymbol{h}_{G_{t-1}}$ and dynamic neighbors; on the other hand, MG-DVD presents CHGAE to partially learn the graph embedding by discriminatively aggregating newly joined neighbors. Our presented dynamic walk strategy only traverses the changed nodes rather than all nodes in the target graph, which markedly alleviates the search space and time cost compared with the static random walk [Fu *et al.*, 2017; Fan *et al.*, 2018]. Definitely, the set of changed nodes on the target graph is denoted as a dynamic node set ΔV_t, which is formalized as below:

$$\begin{aligned}\Delta V_t = \bigcup \{v_x \in V_t | (\exists e_{x,y} = (v_x, v_y) \notin E_{t-1}) \\ \vee (\exists e_{x,y} = (v_x, v_y) \in E_{t-1} \backslash E_t)\},\end{aligned} \tag{1}$$

where v_x and v_y are nodes in V_t, $e_{x,y}$ is the relation between them. $E_{t-1} \backslash E_t$ denotes the difference set of E_{t-1} and E_t.

Dynamic Walk Incremental Updating Encoder

As mentioned above, DWIUE focuses on dynamically updating graph embedding when $\Delta V_t/V_t \leq \gamma$, which involves three major steps: (A) Construct dynamic walk neighbor set; (B) Incrementally fuse representation matrix; and (C) Aggregate graph embedding.

(A) Construct dynamic walk neighbor set $N^i_{t_v}$: Given the heterogeneous graph G_t and the meta-graphs M=$\{M_1,$

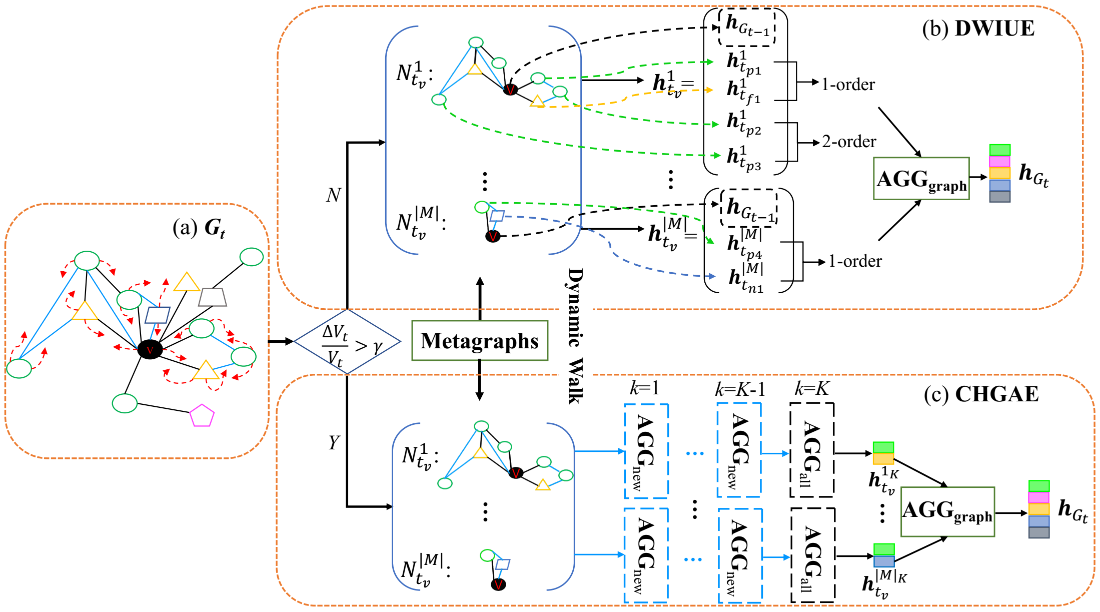


Figure 4: Dynamic graph learning: (a) Heterogeneous graph G_t at sliding window W_t, where the dotted lines with the red arrows indicate the dynamic nodes that need to perform random walks; (b) Dynamic Walk Incremental Updating Encoder (DWIUE); (c) Combined Hierarchical Graph Attention Encoder (CHGAE), where AGG_{new}, AGG_{all}, and AGG_{graph} denote dynamic node aggregator, node aggregator, and meta-graph aggregator, respectively.

$M_2, \ldots, M_{|M|}\}$, the path-relevant neighbor set $N^i_{t_v}$ of the target process v can be obtained with the guidance of each M_i.

(B) Incrementally fuse representation matrix $\boldsymbol{h}^i_{t_v}$: As shown in Figure 4(b), a representation matrix of the target process v can be established by fusing $\boldsymbol{h}_{G_{t-1}}$ at W_{t-1} and its new neighbors' feature vectors at W_t, which leverages both the embedding of itself and the information of its new neighbors in $N^i_{t_v}$. Innovatively, DWIUE selects the intersection between $N^i_{t_v}$ and ΔV_t as the meaningful dynamic neighbors, and the representation matrix $\boldsymbol{h}^i_{t_v} \in \mathcal{R}^{d\times d}$ is expressed as:

$$\mathbf{h}^i_{t_v} = [\mathbf{h}_{G_{t-1}}, \mathbf{h}^i_{t_{u1}}, \ldots, \mathbf{h}^i_{t_{un}}]^T, \tag{2}$$

where $u1 \neq un \in (N^i_{t_v} \wedge \Delta V_t)$, and they are sorted by the node types (i.e., process, file, memory, registry, system, mutex, attribute, and network.) and the distance to node v.

(C) Aggregate graph embedding $\boldsymbol{h}_{G_t}$: Actually, different meta-graphs show distinctive contributions in characterizing malware variants. Inspired by [Wang *et al.*, 2019b], we implement a self-attention mechanism to learn the weight of each meta-graph for detecting malware variants. Concretely, the attention weight θ^i_t of M_i is formalized as below.

$$\theta^i_t = \frac{exp(LeakyReLU(\mathbf{W}^l[\mathbf{h}^i_{t_v}, \mathbf{h}^j_{t_v}] + b^l))}{\sum_{m\in|M|} exp(LeakyReLU(\mathbf{W}^l[\mathbf{h}^i_{t_v}, \mathbf{h}^m_{t_v}] + b^l))}, \tag{3}$$

where $i \neq j \in \{1, \ldots, |M|\}$, $\boldsymbol{h}^i_{t_v}$ and $\boldsymbol{h}^j_{t_v}$ denote the node representations of the target process node v under meta-graph M_i and M_j, respectively. $\boldsymbol{W}^l$ and b^l are the trainable weight and bias parameters. Naturally, the graph embedding of G_t with different meta-graphs can be aggregated as follows.

$$\mathbf{h}_{G_t} = \sum_{i=1}^{|M|} \theta^i_t \times \mathbf{h}^i_{t_v}. \tag{4}$$

Combined Hierarchical Graph Attention Encoder

As shown in Figure 4(c), CHGAE is presented to handle the graph embedding of G_t when threshold $\Delta V_t/V_t > \gamma$. Particularly, CHGAE also consists of three key steps: (A) Construct dynamic walk neighbor set; (B) Aggregate node embedding; (C) Aggregate graph embedding. Notably, steps (A) and (C) of CHGAE are similar to DWIUE. Here, we detail step (B), which can discriminatively aggregate newly joined and known nodes.

(B) Aggregate node embedding $\boldsymbol{h}^{i_K}_{t_v}$: CHGAE calculates the significance of neighbor nodes for representing the target process node *v*, which specifically selects more important neighbor nodes to represent itself rather than uniformly aggregating its neighbor nodes in $N^i_{t_v}$. Formally, the weight of the neighbor node can be defined as:

$$\alpha^i_{t_u} = \frac{exp(\frac{|N^i_{t_v}.u[type]|}{|N^i_{t_v}|\times n})}{\sum_{u\in N^i_{t_v}} exp(\frac{|N^i_{t_v}.u[type]|}{|N^i_{t_v}|\times n})}, \tag{5}$$

where $N^i_{t_v}.u[type]$ denotes the node set whose type is the same as instance *u* in $N^i_{t_v}$, and *n* depicts *u* as the *n*-order neighbor of the target process node *v*.

G_t evolves from the $DHGS = \{G_1, G_2, \ldots, G_{t-1}\}$, which contains the previously known nodes before W_t and newly added nodes at W_t. Notably, the known nodes in G_t have been aggregated in G_{t-1}; thus, CHGAE only learns newly added nodes in $N^i_{t_v}$ at W_t. As shown in Figure 4(c), the former (*K*-1) layers of the proposed aggregator only tackle the new neighbors in $N^i_{t_v}$, and the last layer aggregates all neighbors in $N^i_{t_v}$, which greatly reduces the time cost of node embeddings. The former (*K*-1) layers can be defined as:

Type	Samples	Type	Samples	Type	Samples
Trojan	4,536	Virus	1,606	Worm	842
Backdoor	660	Adware	394	Exploit	338
Dropper	118	Benign	3,042		

Table 1: Sample distribution in ACT-KingKong dataset.

Entity	Number	Entity	Number
Process	2,869,361	File	1,284,625
Memory	641,945	Registry	436,773
System	284,614	Mutex	44,498
Attribute	31,608	Network	26,976

Table 2: Statistics of ACT-KingKong dataset.

$$\begin{aligned}\mathbf{h}_{t_v}^{i_k} &= AGGREGATE_k(\mathbf{h}_{t_v}^{i_{k-1}}, \{\mathbf{h}_{t_u}^{i_{k-1}}\}_{u\in(N_{t_v}^i \wedge \Delta V_t)}) \\ &= \sigma((1+\epsilon_k)\mathbf{h}_{t_v}^{i_{k-1}} + \sum_{u\in(N_{t_v}^i \wedge \Delta V_t)} \alpha_{t_u}^i \mathbf{W}_k \mathbf{h}_{t_u}^{i_{k-1}}), \quad (6)\end{aligned}$$

where $k \in \{1, 2, \ldots, K\text{-}1\}$ denotes the index of the layer, $\boldsymbol{h}_{t_v}^{i_k}$ and $\boldsymbol{h}_{t_u}^{i_k}$ are the feature vectors of target process node v and neighbor node u at the k_{th} layer, respectively, and they can initialize by their state vectors. ϵ_k is a trainable trade-off parameter. Moreover, the K_{th} layer of the proposed aggregator is similar to the previous ones, which aggregates all neighbors in $N_{t_v}^i$ to represent the final node embedding.

3.3 Real-time Detection

In this section, we leverage the Pearson correlation coefficient [Feng *et al.*, 2019] between the target variant and all samples to detect the target variant in real time. Intuitively, the real intentions (e.g., stealing sensitive information, etc.) of the new variant is similar to that of the original malware. Formally, the Pearson correlation coefficient is defined as:

$$\rho(\mathbf{h}_{G_t}, \mathbf{h}_{G_f}) = \frac{E[(\mathbf{h}_{G_t} - \mu_{\mathbf{h}_{G_t}})(\mathbf{h}_{G_f} - \mu_{\mathbf{h}_{G_f}})]}{\sigma_{\mathbf{h}_{G_t}}\sigma_{\mathbf{h}_{G_f}}}, \quad (7)$$

where $\mu_{\mathbf{h}_{G_t}}$ and $\mu_{\mathbf{h}_{G_f}}$ are expectations of $\boldsymbol{h}_{G_t}$ and $\boldsymbol{h}_{G_f}$, respectively. $\sigma_{\mathbf{h}_{G_t}}$ and $\sigma_{\mathbf{h}_{G_t}}$ are standard deviations of $\boldsymbol{h}_{G_t}$ and $\boldsymbol{h}_{G_f}$, respectively.

MG-DVD outputs the type of the sample with the highest Pearson correlation coefficient exceeded the threshold τ as the detection result. Particularly, if $\rho(\boldsymbol{h}_{G_t}, \boldsymbol{h}_{G_f}) \geq \tau$, MG-DVD continues to slide forward until the two detection results are consistent. Inversely, if the Pearson correlation coefficients with all samples are less than τ, it is essential to slide forward and learn a richer embedding to continue detecting.

Finally, the objective function of dynamic malware variants detection is formulated as below.

$$l = \sum_{p=1}^{|P|}(\rho(\mathbf{h}_{G_{p1}}, \mathbf{h}_{G_{p2}}) - y_p)^2, \quad (8)$$

where y_p=1 if $\boldsymbol{h}_{G_{p1}}$ and $\boldsymbol{h}_{G_{p2}}$ are graph embeddings from the same malware; otherwise, label y_p=-1. Here, we optimize l with Adam optimizer [Kingma and Ba, 2014].

Method	Recall	Precision	ACC	F-1	AUC
SVM+RBF	0.767	0.878	0.764	0.819	0.701
RNN+LR	0.847	0.911	0.851	0.878	0.868
MalConv	0.833	0.908	0.843	0.869	0.839
CNN+BPNN	0.873	0.919	0.882	0.895	0.884
MatchGNet	0.913	0.937	0.917	0.925	0.916
MG-DVD	**0.965**	**0.981**	**0.976**	**0.973**	**0.952**

Table 3: Performance on malware variants detection.

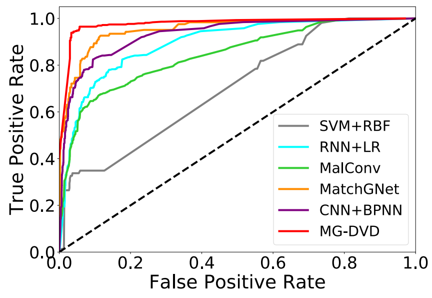


Figure 5: ROC curves on malware variants detection.

4 Experiments

4.1 Dataset

We collect a large number of real-world executable (PE) files from VirusTotal,[1] which contains the latest Windows malware and variants from Mar 2019 to Oct 2019. Afterwards, the execution events streams dataset of all PE files is captured using the KingKong system, namely ACT-KingKong.[2] More specifically, the sample distribution and basic statistics for our ACT-KingKong dataset are presented in Table 1 and Table 2, respectively. For all samples, we randomly divide the training set, validation set, and test set into 6:2:2.

4.2 Result Analysis

In this section, we validate the effectiveness, efficiency, and interpretability of MG-DVD in detecting malware variants by comparing it with five strong baseline methods, where SVM+RBF [Gaviria de la Puerta and Sanz, 2017], MalConv [Raff *et al.*, 2018] and CNN+BPNN [Zhang *et al.*, 2019] belong to signature-based detection methods, RNN+LR [Pascanu *et al.*, 2015] and MatchGNet [Wang *et al.*, 2019b] belong to behavior-based detection methods. We implement or utilize the source codes shared by the authors and adopt the same parameters in their works.

Effectiveness Evaluation

As shown in Table 3 and Figure 5, our proposed MG-DVD outperforms all baseline methods in terms of all metrics. The improvement of MG-DVD can be attributed to the following traits. First, comparing with feature-based methods (e.g., SVM+RBF, MalConv, RNN+LR, and CNN+BPNN), MG-DVD models the fine-grained execution events streams of malware variants into dynamic heterogeneous graphs sequences to effectively learn the evolutionary patterns between

[1]https://www.virustotal.com.

[2]https://github.com/yidun1027/ACT-KingKong.

Method	SVM+RBF	RNN+LR	MalConv	CNN+BPNN	MatchGNet	MG-DVD$_{CHGAE}$	MG-DVD$_{staticwalk}$	**MG-DVD**
Total time(s)	108.13	156.44	79.78	202.53	414.35	1083.07	1035.81	**345.50**
Detection time(s)	5.60	3.37	3.21	7.12	13.79	25.21	19.33	**8.84**

Table 4: Runtime comparison.

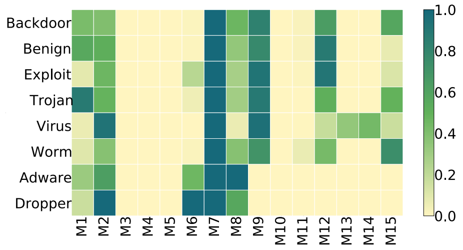
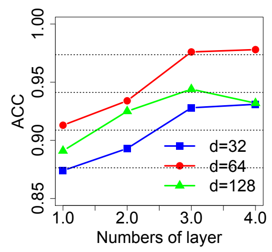
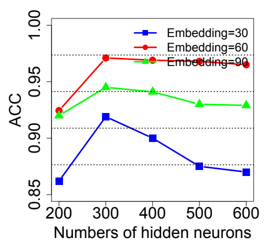


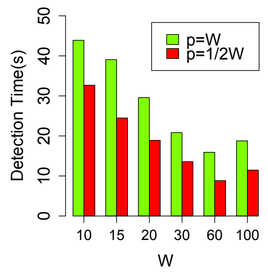
Figure 6: Interpretability for MG-DVD.



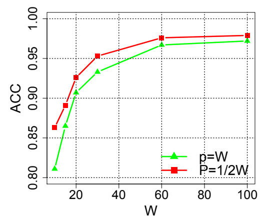
(a) Number of layers and dimension of matrix vs. ACC



(b) Number of hidden neurons and embedding size vs. ACC



(c) W and p vs. detection time



(d) W and p vs. ACC

Figure 7: Parameter sensitivity.

malware and their variants. Notably, our MG-DVD achieves a 9–21% improvement in terms of ACC against feature-based methods. Second, compared with MatchGNet, MG-DVD implements a metagraph-guided dynamic graph learning method, which can learn more fine-grained representations of malware variants to accurately distinguish the new malware variants. Indeed, MG-DVD generates fewer false positives (i.e., 21) than MatchGNet (i.e., 46), indicating that MG-DVD can relieve alarm fatigue.

Efficiency Evaluation

Here we evaluate the efficiency of MG-DVD from two aspects. Firstly, we compare it with its two variants, namely MG-DVD$_{staticwalk}$ and MG-DVD$_{CHGAE}$. As shown in Table 4, MG-DVD is 2.18× and 3× faster than MG-DVD$_{staticwalk}$ in terms of detection time and total time. Moreover, MG-DVD is significantly faster than MG-DVD$_{CHGAE}$ since the encoder DWIUE in MG-DVD can skillfully reduce the time for node aggregation compared to the CHGAE. Secondly, we compare MG-DVD with the existing feature-based and metapath-based methods. As we presented in Table 4, feature-based detection methods, such as SVM+RBF, RNN+LR, MalConv, and CNN+BPNN, are faster than MG-DVD due to the fact that they do not consider the complex graph structure from various types of entity and relation. On the contrary, the static MatchGNet is slower than MG-DVD, which means that the metagraph-guided dynamic embedding encoders in our MG-DVD are beneficial to real-time malware variant detection.

Interpretability Evaluation

Figure 6 shows the frequency matrix of the top-k meta-graphs with the largest weights for different types of malware, which improves interpretability for malware variants detection, from which we can learn that: (i) several meta-graphs show crucial significance on detecting all types of malware variants, such as M_1, M_2, M_7, and M_8; (ii) several meta-graphs only focus on specific types, such as M_{13} and M_{14} are only involved in Virus, which depicts that the Virus is usually active in connecting to the network and infecting files.

4.3 Parameter Sensitivity

In this section, we conduct a large volume of experiments to analyze the sensitivity of different parameters in MG-DVD, including the number of layers (i.e., K), dimension of representation matrix (i.e., d), number of hidden neurons, embedding size, window size (i.e., W), and sliding step (i.e., p).

As shown in Figure 7(a), ACC stably increases with K increases from 1 to 3, when K= 4, the framework is more complex, yet has little improvement on the ACC. Moreover, the performance of our MG-DVD is outstanding in terms of accuracy and stability when the dimension of the representation matrix is set to 64. Similarly, as shown in Figure 7(b), our MG-DVD is the most stable and effective when the number of hidden neurons is set to 300 and embedding size = 60. As demonstrated in Figure 7(c), the detection time when p=1/2W is shorter than that of p=W, which can be attributed to adjacent sliding windows having more overlapping behaviors when p=1/2W, reducing the time cost of node retraining. Moreover, the larger W, the less detection time; however, when W is greater than the 60s, the detection time becomes longer because the number of sliding times is fixed when W reaches a certain value (i.e., 60s), but the more neighbors need to walk and aggregate in larger windows, causing more time cost. In addition, in Figure 7(d), we can observe that the improvement of ACC becomes slow after W=60s. Conse-

Method	Walk	Embedding	Meta-structure	ACC
MatchGNet	static	HAGNE	meta-path	0.917
MG-DVD$_{staticwalk}$	static	DWIUE+CHGAE	meta-graph	0.958
MG-DVD$_{CHGAE}$	dynamic	CHGAE	meta-graph	0.924
MG-DVD$_{DWIUE}$	dynamic	DWIUE	meta-graph	0.853
MG-DVD	dynamic	DWIUE+CHGAE	meta-graph	0.976

Table 5: A summary of the ablation study.

quently, we set W=60s and p=1/2W in MG-DVD to hold the trade-off between detection accuracy and efficiency.

4.4 Ablation Study

We perform a detailed ablation study comparing different variants of our MG-DVD, investigating the effects of different modules in the model. Table 5 summarizes our experiments. We first evaluate the two metagraph-guided dynamic embedding encoders by comparing MG-DVD with MatchGNet [Wang *et al.*, 2019b], which indicates the importance of the meta-graph and two dynamic embedding encoders (i.e., DWIUE and CHGAE) to improve detection performance. We also study the effect of the dynamic walk strategy by comparing MG-DVD with MG-DVD$_{staticwalk}$ that only adopted static random walk strategy. Table 5 shows that the dynamic walk strategy in MG-DVD outperforms the existing static random walk strategy since the dynamic walk strategy only traverses the newly added nodes rather than all nodes in the target graph. Additionally, we examine the discriminative effects of each metagraph-guided dynamic embedding encoder by experimenting with MG-DVD$_{CHGAE}$ and MG-DVD$_{DWIUE}$, respectively. As can be seen from Table 5, the detection accuracy of MG-DVD$_{CHGAE}$ is a significant boost than that of MG-DVD$_{DWIUE}$, which can be attributed to employing aggregators to aggregate high-order neighborhoods in CHGAE can more effectively capture semantic information than directly expanding the state vector of the target node in DWIUE.

5 Related Work

5.1 Signature-based Malware Detection

There have been numerous studies concerning signature-based malware detection [Gandotra *et al.*, 2014; Kang *et al.*, 2016; Gaviria de la Puerta and Sanz, 2017; Raff *et al.*, 2018; Zhang *et al.*, 2019; Singh and Singh, 2020]. Concretely, [Kang *et al.*, 2016; Gaviria de la Puerta and Sanz, 2017; Raff *et al.*, 2018] extracted the opcode or bytecode to detect malware, yet they rely on manually extracting features from known malware samples, which hardly identify new types of malware. Zhang et al. [Zhang *et al.*, 2019] presented a hybrid model consisting of CNN and BPNN to extract more advanced malware features from raw opcodes and APIs. However, they are vulnerable to code obfuscation techniques, inevitably leading to a drop in malware variants detection performance.

5.2 Behavior-based Malware Detection

Recently, extensive studies were implemented on detecting malware based on execution behaviors [Pascanu *et al.*, 2015; Sun *et al.*, 2016; Bartos *et al.*, 2016; Zhang *et al.*, 2018; Wang *et al.*, 2019b; Zhang *et al.*, 2020]. Pascanu et al. [Pascanu *et al.*, 2015] and Zhang et al. [Zhang *et al.*, 2018] verified that the API sequences of malware are significantly different from that of the legitimate programs, which provides a new avenue for behavior-based malware detection. Nevertheless, these methods ignored the interactive relationships among various malware objects and cannot learn the evolutionary patterns between malware and their variants.

To address the limitations, Wang et al. [Wang *et al.*, 2019b] proposed MatchGNet, and they characterized execution events of malware into a static heterogeneous graph and extracted metapath-based features to detect malware, which can effectively identify the unknown malware from a large number of benign samples. However, MatchGNet exposes two serious deficiencies. First, it focused on learning metapath-based malware features, which is unable to capture high-order and fine-grained malware variant patterns, resulting in a drop in malware variants detection performance and high false positive. Secondly, MatchGNet learned a long-term representation from the static heterogeneous graph, which cannot meet the dynamic detection requirements.

6 Conclusion

In this paper, we proposed MG-DVD, a dynamic malware variants detection framework. MG-DVD first utilized dynamic heterogeneous graphs sequence (DHGS) to model the execution events streams of malware variants and introduced meta-graphs to characterize the interactive relationships between malware objects. To realize real-time malware variants detection, MG-DVD then proposed two dynamic walk-based encoders, which leverages overlapping information between adjacent sliding windows to dynamically learn node embeddings. Finally, MG-DVD trained a Pearson-based model to automatically detect malware variants. Extensive experiments show that MG-DVD is effective and efficient in dynamically detecting new malware variants. In the future, we would expand the MG-DVD to a more high-level dynamic detection method by considering the temporal correlation between various behaviors.

Acknowledgements

This work was supported by the 2020 Industrial Internet Innovation and Development Project-the Key Project of Intelligent Connected Vehicle Safety Inspection Platform (Tender No. TC200H01S), the Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003020-ZC), and the Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

[Bartos *et al.*, 2016] Karel Bartos, Michal Sofka, and Vojtech Franc. Optimized invariant representation of network traffic for detecting unseen malware variants. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 807–822, 2016.

[Chen *et al.*, 2018] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[Fan *et al.*, 2018] Yujie Fan, Shifu Hou, Yiming Zhang, Yanfang Ye, and Melih Abdulhayoglu. Gotcha-sly malware! scorpion a metagraph2vec based malware detection system. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 253–262, 2018.

[Feng *et al.*, 2019] Wanli Feng, Quanyin Zhu, Jun Zhuang, and Shimin Yu. An expert recommendation algorithm based on pearson correlation coefficient and fp-growth. *Cluster Computing*, 22(3):7401–7412, 2019.

[Fu *et al.*, 2017] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1797–1806, 2017.

[Gandotra *et al.*, 2014] Ekta Gandotra, Divya Bansal, and Sanjeev Sofat. Malware 2014 survey-analysis and classification. *Journal of Information Security*, 5(2):56–64, 2014.

[Gaviria de la Puerta and Sanz, 2017] José Gaviria de la Puerta and Borja Sanz. Using dalvik opcodes for malware detection on android. *Logic Journal of the IGPL*, 25(6):938–948, 2017.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[Kang *et al.*, 2016] BooJoong Kang, Suleiman Y Yerima, Kieran McLaughlin, and Sakir Sezer. N-opcode analysis for android malware classification and categorization. In *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*, pages 1–7. IEEE, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Pascanu *et al.*, 2015] Razvan Pascanu, Jack W Stokes, Hermineh Sanossian, Mady Marinescu, and Anil Thomas. Malware classification with recurrent networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1916–1920. IEEE, 2015.

[Raff *et al.*, 2018] Edward Raff, Jon Barker, Jared Sylvester, Robert Brandon, Bryan Catanzaro, and Charles K Nicholas. Malware detection by eating a whole exe. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Singh and Singh, 2020] Jagsir Singh and Jaswinder Singh. A survey on machine learning-based malware detection in executable files. *Journal of Systems Architecture*, page 101861, 2020.

[Sun and Han, 2012] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[Sun *et al.*, 2016] Mingshen Sun, Xiaolei Li, John CS Lui, Richard TB Ma, and Zhenkai Liang. Monet: a user-oriented behavior-based malware variants detection system for android. *IEEE Transactions on Information Forensics and Security*, 12(5):1103–1112, 2016.

[Wang *et al.*, 2019a] Jinyan Wang, Chen Liu, Xingcheng Fu, Xudong Luo, and Xianxian Li. A three-phase approach to differentially private crucial patterns mining over data streams. *Computers &amp; Security*, 82:30–48, 2019.

[Wang *et al.*, 2019b] Shen Wang, Zhengzhang Chen, Xiao Yu, Ding Li, Jingchao Ni, Lu-An Tang, Jiaping Gui, Zhichun Li, Haifeng Chen, and Philip S. Yu. Heterogeneous graph matching networks for unknown malware detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3762–3770, 2019.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[Ye *et al.*, 2019] Yanfang Ye, Shifu Hou, Lingwei Chen, Jingwei Lei, Wenqiang Wan, Jiabin Wang, Qi Xiong, and Fudong Shao. Out-of-sample node representation learning for heterogeneous graph in real-time android malware detection. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4150–4156. AAAI Press, 2019.

[Zhang *et al.*, 2018] Jixin Zhang, Kehuan Zhang, Zheng Qin, Hui Yin, and Qixin Wu. Sensitive system calls based packed malware variants detection using principal component initialized multilayers neural networks. *Cybersecurity*, 1(1):10, 2018.

[Zhang *et al.*, 2019] Jixin Zhang, Zheng Qin, Hui Yin, Lu Ou, and Kehuan Zhang. A feature-hybrid malware variants detection using cnn based opcode embedding and bpnn based api embedding. *Computers & Security*, 84:376–392, 2019.

[Zhang *et al.*, 2020] Zhaoqi Zhang, Panpan Qi, and Wei Wang. Dynamic malware analysis with feature engineering and feature learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1210–1217, 2020.

[Zhao *et al.*, 2017] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 635–644, 2017.