An Ensemble Learning System to Mitigate Malware Concept Drift Attacks

Zhi Wang, Meiqi Tian, Junnan Wang, Chunfu Jia, Ilsun You*, Zheli Liu

Nankai University

Abstract. The wild adoption of machine learning algorithms in malware detection systems is based on the assumption that, training data set always conforms with real-world traffic. However, such assumption is vulnerable to well-crafted concept drift attacks, including mimicry attacks, gradient descent attacks, poisoning attacks and so on. As a result, machine learning itself could become the Achilles' heel in a malware detection system. To prevent this from happening, it is important to timely detect and prevent concept drift attacks. In this paper, we propose an ensemble learning system that combines vertical and horizontal correlation models. The significant difference between vertical and horizontal correlation models increases the difficulty of concept drift attacks. Moreover, average p-value assessment is applied to fortify the system to be more sensitive to hidden concept drift attacks. Upon recognization of concept drift attacks, SIM and DIFF assessments are introduced to locate the affected features. Then, active feature reweighting is used to mitigate model aging. The experiment results show that the hybrid system could recognize the concept drift among different Miuref variants, and reweight affected features to avoid model aging.

Key words: malware detection, machine learning, concept drift attack, vertical correlation, horizontal correlation

1 Introduction

According to AV-Test[1], over 390,000 new malicious programs are detected every day. The enormous volume of new malware variants renders manual malware analysis inefficient and time-consuming. On one hand, machine learning has been widely used in botnet detection system as a core component [1–3] to reduce human involvement. On the other hand, with financial motivation, attackers also keep up with new detection algorithms and evolve their evasion tricks. Nowadays, over 70% of the advanced malware uses one or more evasion techniques to avoid detection [2].

The assumption of machine learning algorithm is that the underlying malicious data distribution is stable for training and testing, which is vulnerable to well-crafted concept drift attacks, such as new communication channel [4–8],

[1] https://www.av-test.org/en/statistics/malware/

[2] https://go.lastline.com/webinar-protect-your-network-from-evasive-malware.html

mimicry attack [9, 10], gradient descent attack [9, 10], poison attack [11], and so on.

To build secure and sustainable detection system against evasive malware, recognizing the concept drift of underlying malicious data is very important for machine learning models. In this paper, we introduce statistical p-values to combine diverse vertical and horizontal correlation learning models, that increase prediction quality and be more sensitive to hidden concept drift attacks.

Vertical correlation model focuses on the life cycle of a single malware sample, such as BotHunter [12]; While horizontal correlation learning approach builds detection model based on the behavior similarity among a large number of malware variants, such as BotFinder [13]. There is a significant diversity among vertical and horizontal correlation learning models. Comparing the prediction results and p-values given by the two diverse models, we could obtain more insights into the hidden malware concept drifting. In a nutshell, this paper makes the following contributions:

- We propose a hybrid malware detection system that based on statistical p-values combines vertical and horizontal learning model to mine malicious data internal patterns from diverse perspectives.
- The p-value is more fine-grained than fixed empirical threshold. So our p-value based hybrid system could identify concept drift earlier than threshold based detection system.
- We use SIM and DIFF algorithm to assess the contribution of each features in different time windows and recognize the affected features when concept drift is identified, and use feature reweighting to mitigate model aging.

The remainder of this paper is outlined as follows. In Section 2, we present the related works. Section 3 presents the architecture of our hybrid botnet detection system, and describes each components. Section 4 presents our experiments performed to assess the recognition of underlying data distribution concept drift. In section 5, we discuss the limitations and future work, and in Section 6 we summarize our results.

2 Related Works

Nowadays, machine learning is widely used in malware detection system as a core component. However, Arce [14] pointed out that machine learning itself could be the weakest link in the security chain. By exploiting the knowledge of the machine learning (ML) algorithm, many well-crafted evasion approaches have been proposed to evade or mislead ML models [15]. Figure 1 shows the levels of attacker's knowledge that is extended from the graph in Srndic and Laskov [16].

Botnet attackers have begun to exploit many stealthy C&C channels, such as social network [17, 18], email protocol [19], SMS [4] and bluetooth [5]. Erhan *et al.* [17] proposed social network based botnet to abuse trusted popular websites,

such as twitter.com, as C&C server. Kapil *et al.* [19] evaluate the viability of using harmless-looking emails to delivery botnet C&C message. Social network traffic and email traffic are beyond the data collection scope of machine learning based methods, that lack of clear mitigation strategies. What makes new protocols interesting is the introduced trusted and popular websites or email servers. First, trusted websites or email servers have very good reputation and usually are listed on the white list that all traffic to such website or server will not be monitored by botnet detection methods. Second, the trusted websites or email services are very popular and have very heavy usage volume that the light-weight occasional C&C traffic is unlikely to be noticed.

However, the new botnets use the centralized architecture that all bots communicate with C&C server directly. The central C&C server is a potential single point of failure that if the C&C server is exposed to the defender, the botnet is easy to be dismantled. Mimicry attack refers to the techniques that mimic benign behaviors to reduce the differentiation between the malicious events and benign events. Wagner and Soto [20] demonstrated the mimicry attack against a host-based IDS that mimicked the legitimate sequence of system calls. Srndic and Laskov [16] presented a mimicry attack against PDFRate [21], a system to detect malicious pdf files based on the random forest classifier.

The gradient descent is an optimization process to iteratively minimize the distance between malicious points and benign points. Srndic and Laskov [9] applied a gradient descent-kernel density estimation attack against the PDFRate system that uses SVM and random forest classifier. Biggio et al. [10] demon-

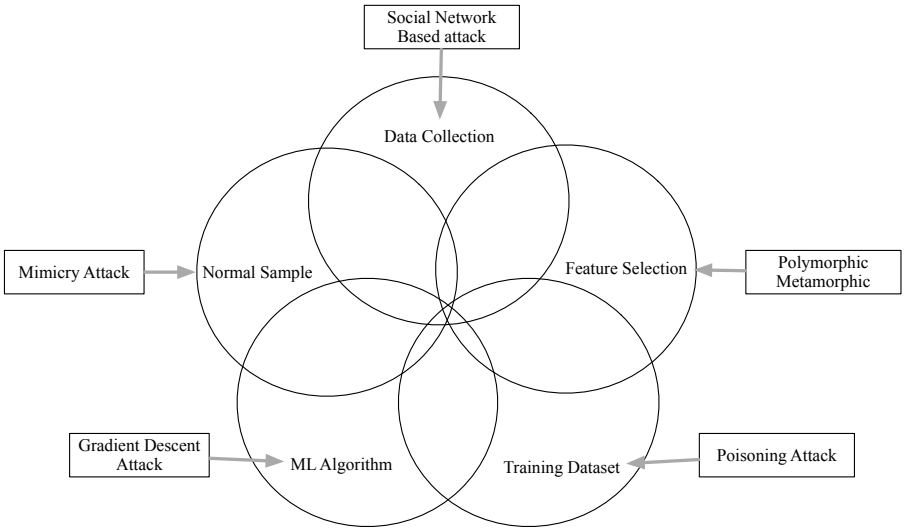


Fig. 1. Attack methods against machine learning detection approaches at different knowledge levels

strated a gradient descent component against the SVM classifier and a neural network.

Poisoning attacks work by introducing carefully crafted noise into the training data. Biggio et al. [11] proposed poisoning attacks to merge the benign and malicisous clusters that make learning model unusable.

Therefore, malware problems are not stable but change with time. For machine learning based malware detectors, they are designed under the assumption that the training and testing data follow the same distribution which make them vulnerable to concept drift problem that the underlying data distribution are changing with time. One of the concept drift mitigation approach is to recognize and react to recent concept changes. Demontis et al. [11] proposed an adversary-aware approach to proactively anticipates the attackers. Deo et al. [22] presented a probabilistic predictor to assess the underlying classifier and retraining model when it recognized concept drift. In this paper we will introduce conformal evaluation into the horizontal correlation learning model that assess the prediction quality and understand the statistical distribution of data.

3 Ensemble Malware Detection System

Driven by financial motivation, malware authors keep evolving malware perpetually using evasion tricks to avoid detection, especially to bypass widely deployed learning-based models. Many learning-based detection models calculate a score to a new approaching sample describing the relationship between the known malware samples and the new one. Then detectors compare the score with a fixed and empirical threshold to make a decision if it is malicious. The threshold usually fits the old training dataset very well, even overfits, however, the performance degenerates to the new ever-changing malicious dataset with time. In this paper, we propose a hybrid botnet detection system (HBDS) based on statistical p-values using vertical life-cycle algorithm and horizontal traffic similarity algorithm as the underlying scoring classifiers that is robust to malware concept drift attacks. And average p-value assessment is introduced to recognize the traditional learning-based models to mitigate model aging challenge through introducing conformal prediction.

Figure 2 depicts the framework of HBDS which includes five components: non-conformity measure (NCM), conformal learning, concept drift recognition, feature assessment, and active reweighting. The HBDS is an open frame that any machine learning model based on fixed empirical threshold can be integrated into HBDS as a underlying independent NCM. The diverse NCMs model the malware data distribution from different perspectives. In this paper, we select vertical correlation based classifier BotHunter and horizontal correlation based classifier BotFinder as the underlying NCMs. The conformal learning component uses p-values to carry out the further statistical analysis based on NCM scores. The p-value is more fine-grained than threshold that can used to observe the gradual decay of detection model, and p-value is comparable between different models while the NCM scores are not comparable among different models. The

concept drift recognition component uses the average p-value (APV) algorithm to recognize the concept drift of malware data distribution between two different time windows. Feature assessment component introduces SIM and DIFF algorithm to locate the features that are affected by identified concept drift. Active reweighting component dynamically adapts the weight of affected features to update model before the cumulative radical concept drift.

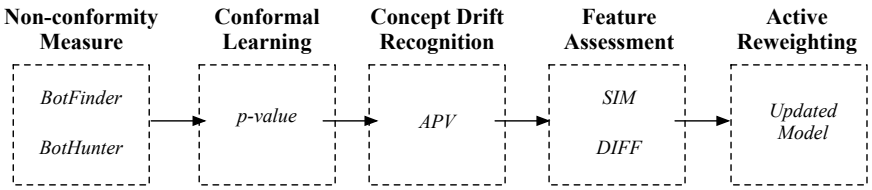


Fig. 2. The framework of hybrid detection system

3.1 Non-conformity Measure

Many machine learning algorithms are in fact scoring classifiers: when trained on a set of observations and fed with a test object x, they could calculate a prediction score $s(x)$ called scoring function. Any scoring classifiers using a fixed and empirical threshold can be introduced into our system as a underlying NCM. Each NCM uses different machine learning algorithm, such as classification, clustering, to model the malware data distribution from different perspectives. Currently, we select BotHunter and BotFinder as the NCMs. BotHunter models the life cycle of botnet from the vertical perspective, while BotFinder selects time related features and traffic volume features to build detection model from the horizontal perspective. The diversity of selected NCM increases the complexity of the successful concept drift attack, since attackers need to obtain more knowledge to construct concept drift attacks against HBDS than the traditional single model detection systems.

The input of the NCM is a known sample set and an unknown sample, and the output is a score that describes the similarity or dissimilarity of the unknown sample to the known sample set.

This paper hybrids two different machine learning models: BotHunter and BotFinder, as shown in table 1.

BotHunter is a multi-dialog-based vertical correlation algorithm. First, BotHunter establish botnet life cycle model according to the behavior sequence pattern of botnets; Then it maps a set of host dialogues to a pre-learned life-cycle model and calculate a score to describe how close between the dialog and the model. When the dialog correlation algorithm shows that a host dialog pattern maps sufficiently close to the life-cycle model based on a threshold, the host is

Table 1. The non-conformity measures of hybrid botnet detection system

	Object	Features	ML Type	Algorithm
BotHunter	Dialog	5	Clustering	CLUES
BotFinder	Trace	7	Classification	Life-cycle model

declared infected. According to the introduction of BotHunter, we construct a botnet life-cycle model in 4 layers and 7 states as shown in Fig. 3.

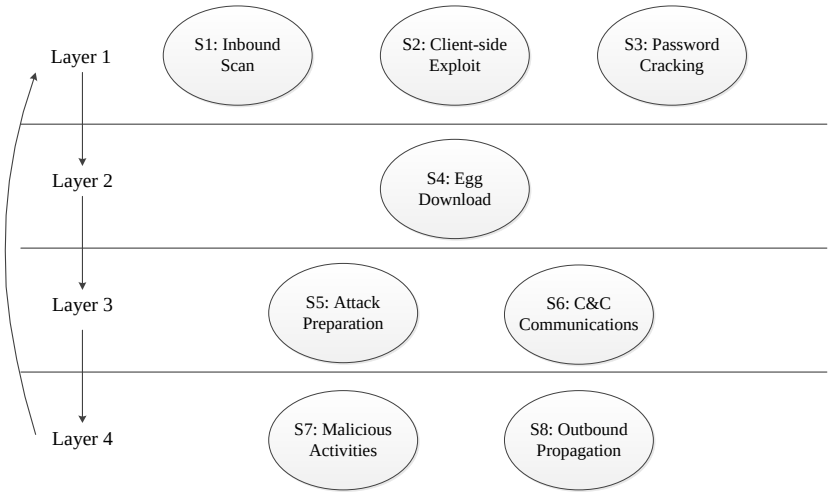


Fig. 3. The architecture of life-cycle model

BotFinder is a detection method that does not require deep packet inspection. First, BotFinder groups netflows with the same source IP, destination IP, destination port number, and communication protocol into trace; Then, it extracts traffic volume features from Trace, such as the average number of sent bytes , the average number of received bytes, and time related features of trace, such as the average time interval, the average duration , and frequency calculated by Fast Fourier Transformation algorithm. BotFinder uses the CLUES algorithm to cluster the similar traces of a botnet family, and builds detection model for each class of this family. This method can effectively identify the botnet network traffic similarity between different malware variants, and give a prediction based on the optimal threshold fitting the training dataset.

3.2 Conformal Learning

Once NCMs are selected, conformal learning component computes a p-value p_{z^*}, which in essence for a new object z^*, represents the percentage of objects in $\{x \in C, \forall C \in \mathbb{D}\}$,(i.e., the whole dataset) that are equally or more estranged to C as p_{z^*}, and we will get a number between 0 and 1.The algorithm is shown in Algorithm 5.

Algorithm 1 P-value calculation used in Conformal Predictor

Require: Dataset $D = \{z_1, , z_n\}$, sequence of objects $C \subset D$, non-conformity measure A, and new object z^*
Ensure: p-value p_{z^*}
1: Set provisionally $C = C \cup \{z^*\}$
2: **for** $i \leftarrow 1$ ***to*** n **do**
3: $\quad \alpha \leftarrow A(C \setminus z_i, z_i)$
4: **end for**
5: $p_{z^*} = \frac{|\{j:\alpha_j \geq \alpha_{z^*}\}|}{n}$

P-value measures the fraction of objects within $\mathbb{D}$, that are at least as different from a class C as the new object z^*. For instance, if C represents the set of malicious activities, a high p-value p_{z^*} means that there are a significant part of the objects in this set is more different than z^* with C, on the other words, z^* is more similar to these malicious activities than the objects that already marked malicious. Therefore, the prediction result based on a high p-value shows a high credibility. P-values are directly involved in our discussion of concept drift.

The p-value is comparable between different learning models while the NCM scores are not comparable among different models, as shown in Fig. 4. The p-value is more fine-grained than threshold that is more sensitive to concept drift attacks. The concept drift recognition component uses the average p-value algorithm to recognize hidden concept drift.

3.3 Concept Drift Recognition

We use the average p-value (APV) algorithm based on time windows to recognize concept drift attacks, as shown in Fig. 5. We group the malware samples into different time windows according to their time stamps in the timeline. We calculate p-values for each malware sample in a time window, and compute the APV value for each time window. Note that, the number of APV for each time window depends on the number of selected NCMs. In this paper, each time window has two APV for the vertical and horizontal NCMs.

The p-values are comparable, and the APV scores are also comparable between different time windows and even in the same window with different underlying NCMs. The change of APV value between different time window reflects the change of underlying malware data distribution with time that can identify

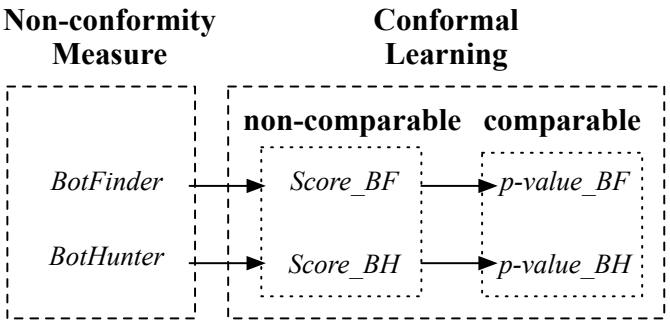


Fig. 4. The conformal learning component transform non-comparable NCM scores to comparable statistical p-values

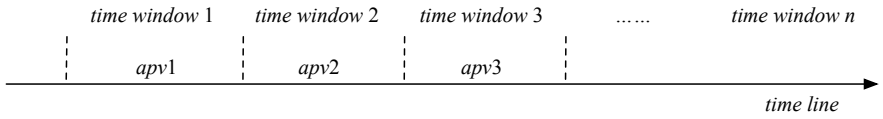


Fig. 5. The conformal learning component calculates APV for each time windows

gradual moderate drift. In the same time window, the difference between APV scores calculated from different NCMs reflect the affection of concept drift to different learning models which can detect the sudden drift.

If the APV score of a certain detection model decreases with time, it shows that the current concept of the malware data distribution is gradually different from the old concept learnt from previous malware data, and indicates that the detection model is suffering from concept drift attack. But the decay of threshold based detection performance may not be observed immediately when concept drift is found. The concept of drift attack is a gradual process. Only when the variation of the underlying data distribution exceeds the boundary of the threshold, the detection model starts make poor decisions. If the APV score does not decrease in the new time window, it means in the current time window, the distribution of malware data does not have concept drift from this observed perspective.

3.4 Feature Assessment

When a detection model finds concept drift attack in current time window, we will use the SIM and DIFF algorithms to locate the features that affected by concept drift. The SIM and DIFF algorithms are used to evaluate the contribution of the features in a data set, as shown in Algorithm 2 and Algorithm 3. $SIM[i]$ represents the effect of the i^{th} feature on the average distance between

two samples with the same label, $DIFF[i]$ represents the effect of the i^{th} feature on the average distance between two samples with different labels.

If $DIFF[i]$ increases, it means that the concept drift affects the i^{th} feature, leave the sample away from the known sample set in the current time window. The value of $SIM[i]$ indicates the aggregation degree of the sample at the i^{th} feature, the smaller value of $SIM[i]$ indicates that the sample is more stable at this feature; the larger value of $SIM[i]$ means the greater noise from this feature.

Algorithm 2 SIM algorithm

Require: feature vectors x_1,x_2,...,x_n,labels y_1,y_2,...,y_n
Ensure: SIM coefficients
  SIM =[]
  X ={x_1,x_2,...,x_n}
  uY =entries in Y without repetition
  for $i = 1$ to d **do**
    $X_i = i^{th}$ column of X
    $SIM[i] = 0$
    for class in elements of uY **do**
      fromclass = X_i[where Y == class]
      pdist = pairwise_distances(fromclass)
      $SIM[i] = SIM[i]$ + (sum(pdist)/length(pdist))
    end for
    $SIM[i] = SIM[i]$/length(uY)
  end for

Algorithm 3 $DIFF$ algorithm

Require: feature vectors x_1,x_2,...,x_n,labels y_1,y_2,...,y_n
Ensure: $DIFF$ coefficients
  $DIFF$ =[]
  X ={x_1,x_2,...,x_n}
  uY = unique entries in Y: our classes
  for $i = 1$ to d **do**
    $X_i = i^{th}$ column of X
    $DIFF[i] = 0$
    for class in elements of uY **do**
      $fromclass = X_i$[where Y == class]
      $notclass = X_i$[where Y != class]
      pdist = respective distanced between elements in $fromclass$ and elements in $notclass$
      $DIFF[i] = DIFF[i]$ + (sum(pdist)/length(pdist))
    end for
    $DIFF[i] = DIFF[i]$/length(uY)
  end for

3.5 Active Reweighting

When concept drift is recognize to a detection model, we will reweight the affected features according to the SIM and DIFF results to actively update the model before the cumulative radical drift. The formula for the feature reweighting is:

$$W_i = 1/SIM[i] + DIFF[i] \tag{1}$$

By updating the weight, we can reduce the weight of the feature that is significantly influenced by concept drift attack, and increase the anti-aging ability of the detection model.

4 Experiment

In this paper, we use the public CTU datasets for our experiment that is provided by Malware Capture Facility project[3]. They capture long-live real botnet traffic and generate labeled netflow files that is publish for malware research. The traffic dataset is from 2011 to present.

To recognize the sudden radical concept drift between different botnet families is not the focus of this paper. We plan to recognize the hidden and gradual concept drift between different variants in the same family that is not noticed by traditional models using fixed and empirical threshold. So we select the Miuref family for our experiment from CTU dataset, because Miuref has 4 variants and 8 different traffic records which is more than other families in the public CTU dataset. Miuref redirects web browser to carry out click fraud or download other malware. The four variants of Miuref (V1, V2, V3, and V4) are listed in the table /reftable:miuref.

Table 2. The network traffic of 4 variants in Miuref family

CTU file name	variant	time window
127-1	2015.06.01-2015.06.07	V1
127-2	2015.06.09-2015.07.08	V1
128-1	2015.06.01-2015.06.07	V2
128-2	2015.06.09-2015.07.19	V2
169-1	2016.08.03-2016.08.04	V3
169-2	2016.08.04-2016.08.04	V3
169-3	2016.08.03-2016.08.11	V3
173-1	2016.08.04-2016.08.11	V4

According to the time stamps of each time window, we order the variants for the experiments of concept drifting recognize and feature assessment and active

[3] Garcia, Sebastian. Malware Capture Facility Project. Retrieved from https://stratosphereips.org

reweighting. First, we split the 4 variants into 2 time windows that V1 and V2 are grouped into the first time window whose malware data is collected in 2015, and V3 and V4 are grouped into the second time window whose malware data is captured in 2016.

To assess the consecpt drift between different time windows and from different perspectives, we use dimension reduction algorithm tSNE [23] and statistical p-values to see the underlying data distribution.

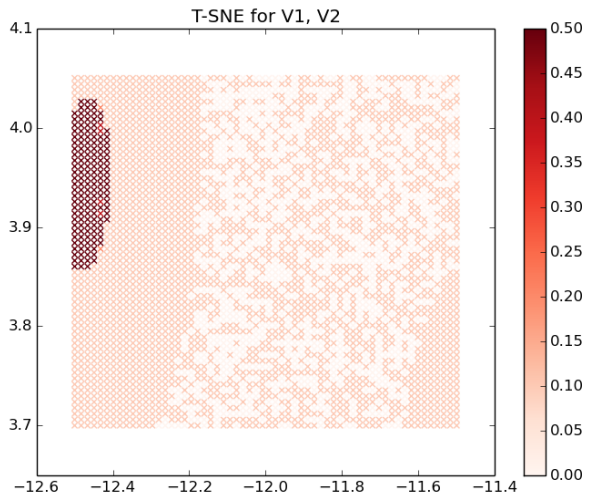
The tSNE is a kind of reduced dimension visualization algorithm, which maps the multi-dimensional features to two or three dimensions. The goal of tSNE is to make the distance similar to the elements on the low dimension remain close to each other.

Figure 6 shows the underlying data distribution and p-value significant levels of Miuref family in two different time windows for vertical correlation model. The subfigure 6(a) shows the data distribution of V1 and V2 in tSNE space and the p-values for each point. The labelled colors are for the various p-values that the dark red means the point has high p-values, while the light red means the point is far from the Miuref V1 and V2 variants. The subfigure 6(b) shows the data distribution and p-value significant levels of all Miuref variants in the tSNE space. We can see that from the vertical perspective, the Miuref family has very slight concept drift between the two time windows in different years, because the malware data distribution and p-value significant level are almost stable without much change. In addition, the characteristics of Miuref becomes more remarkable and centralized after absorbing the malware data of V3 and V4 variants captured in 2016, because in the middle of subfigure 6(a) there are some points with week p-value significant level change to be almost zero in the subfigure 6(b).

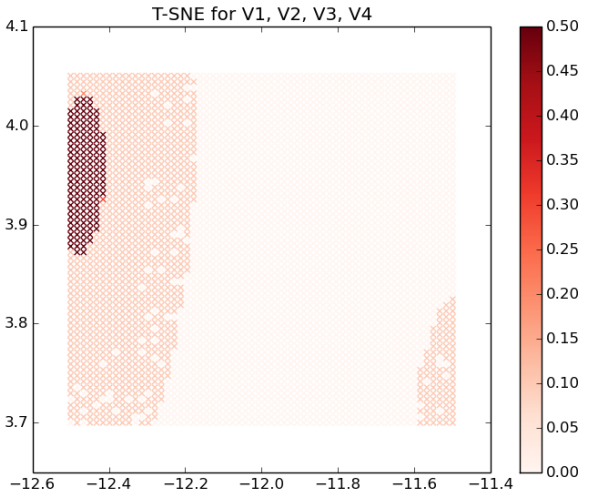
Figure 7 shows the changes of APVs of 4 Miuref variants for vertical correlation model. All 4 APVs are at high APV level, and the APV of V4 is even higher than 0.8 that is consistent with the Miuref data distribution and p-value significant levels in tSNE space. So from the vertical observing perspective, the Miuref family data distribution has not much concept drift, and vertical correlation model is still effective to detect Miuref variants.

Figure 8 shows the underlying data distribution and p-value significant levels of Miuref family in two different time windows for horizontal correlation model. The subfigure 8(a) shows the data distribution of V1 and V2 in tSNE space and the p-values for each point for horizontal model. The subfigure 8(b) shows the data distribution and p-value significant levels of 4 Miuref variants in the tSNE space for horizontal model. We can see that from the horizontal perspective, the Miuref family has significant concept drift between the two time windows in different years, because between the two subfigures, the malware data distribution and p-value significant level are obviously changed, especially at the upper left corner in the figure.

Figure 9 shows the changes of APVs of 4 Miuref variants for horizontal correlation model. We can see that the APV drops dramatically on V4 from 0.7 to 0.4, which means that the underlying V4 data distribution changed significantly



(a) V1 and V2 in the tSNE space for vertical correlation model



(b) V1 and V2 and V3 and V4 in the tSNE space for vertical correlation model

Fig. 6. The data distribution and p-value significant level for vertical correlation model in the tSNE space.

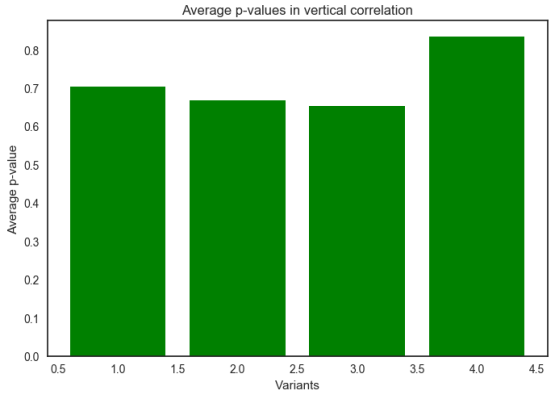
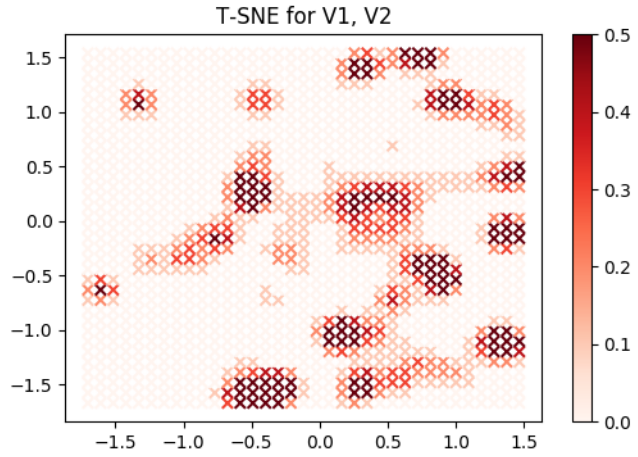


Fig. 7. The change of APVs of 4 variants for vertical correlation model

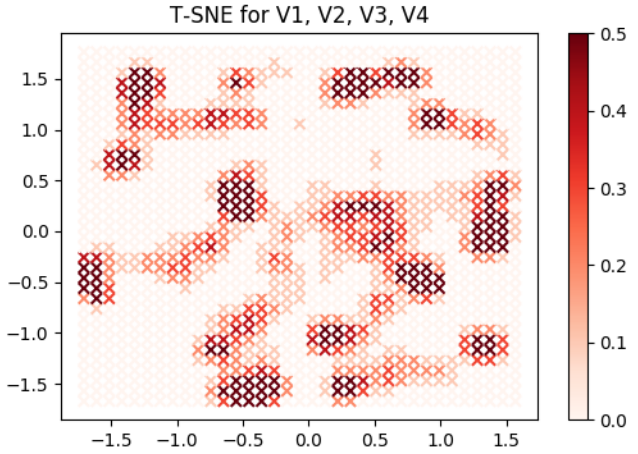
observed from horizontal correlation perspective. Let us understand the concept drift from the cumulative distribution of p-values. As shown in figure 10, most p-values of V4 data is less than 0.4, while the p-values of V1 and V2 and V3 are much higher than 0.4. It can be inferred from figure 8 9 10 that the variant V4 is not consistent of family characteristics and V4 data distribution occurs concept drift.

After recognized concept drift for horizontal correlation model, we use SIM and DIFF algorithms to assess the affection of concept drift to each predictive feature. SIM score represents the distance between the observed samples that belonged to the same class. DIFF score represents the distance between the observed samples that belonged to the same class and the samples belonged to all other samples. SIM and DIFF scores reflect the contributions of features in identifying the new variant to its family. Figure 11 shows the SIM and DIFF average scores of five features used by horizontal correlation model. Figure 11(a) shows that the average SIM and DIFF scores of features for V3 to V1 and V2 are lower than 0.8. In figure 11(b), the average DIFF scores of the second feature and the forth feature are higher than 1, especially, the average DIFF score of forth feature is up to 1.75. This result confirms that the concept drift happened , and mainly caused by the second feature and the forth feature. The average SIM and DIFF scores not only can assess the contribution of features, and provide foundation for reweighting the affected features to mitigate the decay of horizontal correlation model to Miuref family.

In conclusion, concept drift is the significant factor of causing the model aging problem. We hybrid two learning models: horizontal and vertical correlation model to analyze malware data from two diverse perspectives that makes our detection system more robust to the concept drift attacks and increases the complexity for evading learning models. We map concept drift of underlying malware data distribution to tSNE space with p-value significant levels and APV scores, so it will be easier for us to understand and recognize concept drift



(a) V1 and V2 in the tSNE space for horizontal correlation model



(b) V1 and V2 and V3 and V4 in the tSNE space for horizontal correlation model

Fig. 8. The data distribution and p-value significant level for horizontal correlation model in the tSNE space.

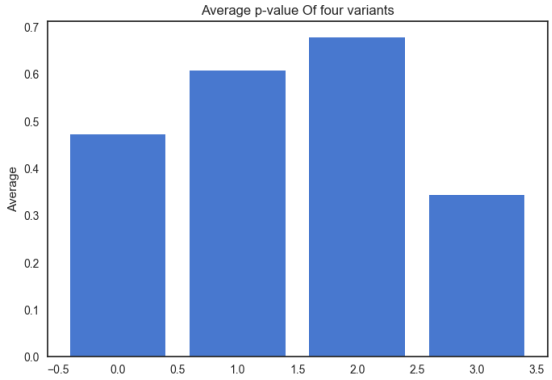


Fig. 9. The change of APVs of 4 variants for horizontal correlation model

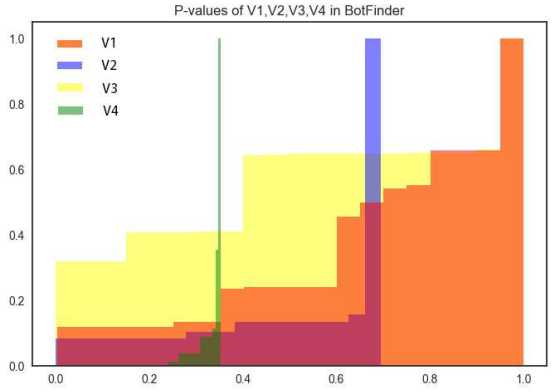
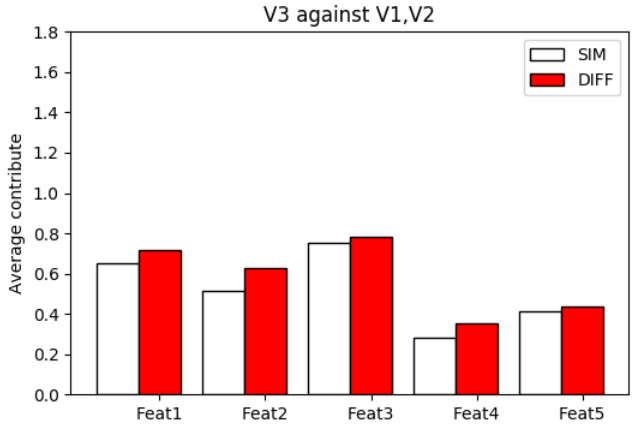


Fig. 10. The cumulative distribution of p-values of 4 variants for horizontal correlation model

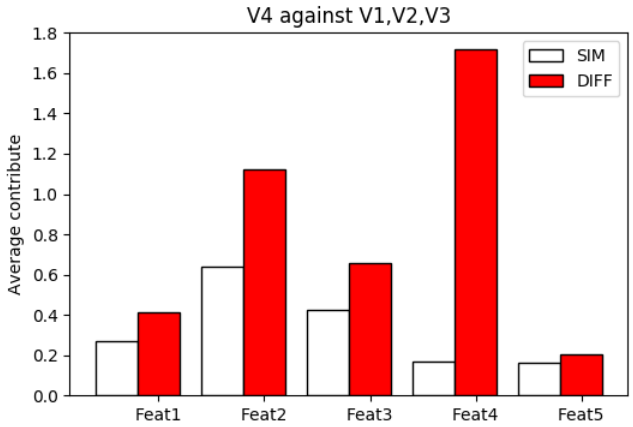
attacks. If concept drift happened, features will be actively reweighting based on the scores of SIM and DIFF to adapt to new malware variants quickly.

5 Discussion

Building machine learning models of malware behaviors is widely used as a panacea towards effective, scalable, and automatic malware detection. For the sake of survival and financial motivation, attackers keep learning the latest machine learning based detection systems and evolving evasion techniques to generate new sustainable variants. Concept drift is the well-known vulnerability of machine learning which is exploited by attackers to launch well-crafted concept drift attacks artificially, such as mimicry attacks, gradient descent attacks and poisoning attacks.



(a) The average contributions of 5 predictive features used by horizontal correlation model for recognizing V3 data.



(b) The average contributions of 5 predictive features used by horizontal correlation model for recognizing V4 data.

Fig. 11. Feature assessment results using SIM and DIFF for V3 and V4.

The real world malware problem concepts are not stable but change with time rapidly, so that machine learning models should quickly recognize and adapt to the hidden changes in the underlying malware data distribution. There are two types of concept drift: sudden drift and gradual moderate drift. Sudden drift means radical changes in the target concept. Single learning model is vulnerable to sudden drift, because single model only observes a particular perspective of malware data distribution.

To handle sudden drift, ensemble learning is needed that hybrid a set of diverse concept descriptions. In this paper, we maintains two much diverse learning

models that observe the malware data distribution from both of vertical life-cycle perspective and horizontal traffic similarity perspective simultaneously. The hybrid model is robust to single concept drift attacks. In the future, we are going to introduce more learning models into our system based on statistical p-value against more and more sophisticated concept drift attacks.

The gradual moderate drift induce less radical changes than sudden drift, but the change is more hidden and difficult to be detected. To recognize and react gradual moderate drift, we introduce statistical p-values to replace fixed, empirical threshold. The p-value gives us the insights of the underlying malware data distribution that is sensitive to gradual moderate drift attacks. SIM and DIFF algorithms can assess the gradual moderate drift affection to each feature, and feature reweighting can update the detection model actively before the cumulative radical drift.

The malware problem is totally different from optical character recognition, speech recognition, bioinformatics and so on, where the training data could be used for many years. As time goes, the cumulative concept drift of malware will be more and more enormous, that the current concept of underlying malware data is much different to previous concept. The old malware concept will be noise to current learning model, so that sliding time window is important for malware problem to select malware data relevant to the current concept. The time window moves over recently arrived malware data, and the learnt concepts only are used for detection in the immediate future. The time window size can be fixed or heuristically determined. In the future work, we will introduce sliding time window into our system.

6 Conclusions

For the survival and financial motivation, malware keeps evolving perpetually and introducing more and more sophisticated evasion techniques. To build a sustainable and secure learning model, we need to quickly recognize and react to the concept drift of underlying malware data distribution. In this paper we proposed a hybrid botnet detection system based on statistical p-values using vertical life-cycle algorithm and horizontal traffic similarity algorithm as the underlying scoring classifiers that is robust to malware sudden concept drift attacks. And average p-value assessment is introduced to recognize gradual concept drift and feature reweigting could update detection model actively before cumulative radical concept drift.

The hybrid malware learning system is an open and scale platform that other threshold based scoring classifiers can easily be integrated into. In the future, we will integrate more diverse scoring classifiers into our system to understand the underlying malware data distribution from more diverse perspectives. And we are going to improve the efficiency of this predictor such as introducing sliding window to online learn the latest concepts and dynamically remove aging data automatically.

References

1. A. Demontis, M. Melis, B. Biggio, D. Maiorca, D. Arp, K. Rieck, I. Corona, G. Giacinto, and F. Roli, "Yes, machine learning can be more secure! a case study on android malware detection," *IEEE Trans. Dependable and Secure Computing*, 2017.
2. S. Garca, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers and Security*, vol. 45, p. 100123, September 2014.
3. S. Garca, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," *Security and Communication Networks*, vol. 7, p. 878903, May 2014.
4. Y. Zeng, K. G. Shin, and X. Hu, "Design of sms commanded-and-controlled and p2p-structured mobile botnets," in *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, (New York, NY, USA), pp. 137–148, ACM, 2012.
5. K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating bluetooth as a medium for botnet command and control," in *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA'10, (Berlin, Heidelberg), pp. 61–80, Springer-Verlag, 2010.
6. K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and Applications*, vol. 22, no. 0, pp. 113 – 122, 2015. Special Issue on Security of Information and Networks.
7. T. Yin, Y. Zhang, and S. Li, "Dr-snbot: A social network-based botnet with strong destroy-resistance," in *IEEE International Conference on Networking, Architecture, and Storage*, pp. 191–199, 2014.
8. E. J. Kartaltepe, J. A. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: Emerging threats and countermeasures," in *Applied Cryptography and Network Security, International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, pp. 511–528, 2010.
9. N. Šrndic and P. Laskov, "Practical evasion of a learning-based classifier: A case study," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP '14, (Washington, DC, USA), pp. 197–211, IEEE Computer Society, 2014.
10. B. Biggio, I. Pillai, S. Rota Bulò, D. Ariu, M. Pelillo, and F. Roli, "Is data clustering in adversarial settings secure?," in *Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security*, AISec '13, (New York, NY, USA), pp. 87–98, ACM, 2013.
11. B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning behavioral malware clustering," in *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec '14, (New York, NY, USA), pp. 27–36, ACM, 2014.
12. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, USENIX Association Berkeley, CA, USA, 2007.
13. F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: Finding bots in network traffic without deep packet inspection," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies(CoNEXT '12)*, (France), pp. 349–360, ACM New York, NY, USA, December 2012.
14. I. Arce, "The weakest link revisited," *IEEE Security and Privacy*, vol. 1, pp. 72–76, Mar. 2003.

15. A. Kantchelian, S. Afroz, L. Huang, A. C. Islam, B. Miller, M. C. Tschantz, R. Greenstadt, A. D. Joseph, and J. D. Tygar, "Approaches to adversarial drift," in *Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security*, AISec '13, (New York, NY, USA), pp. 99–110, ACM, 2013.
16. N. Srndic and P. Laskov, "Practical evasion of a learning-based classier: A case study," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (S&P)*, (SAN JOSE, CA), May 2014.
17. E. J. Kartaltepe, J. A. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: Emerging threats and countermeasures," in *Applied Cryptography and Network Security, International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, pp. 511–528, 2010.
18. T. Yin, Y. Zhang, and S. Li, "Dr-snbot: A social network-based botnet with strong destroy-resistance," in *IEEE International Conference on Networking, Architecture, and Storage*, pp. 191–199, 2014.
19. K. Singh, A. Srivastava, J. Giffin, and W. Lee, "Evaluating emails feasibility for botnet command and control," in *IEEE International Conference on Dependable Systems and Networks with Ftcs and DCC*, (Anchorage, AK), pp. 376–385, IEEE, June 2008.
20. D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, (New York, NY, USA), pp. 255–264, ACM, 2002.
21. C. Smutz and A. Stavrou, "Malicious pdf detection using metadata and structural features," in *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC '12, (New York, NY, USA), pp. 239–248, ACM, 2012.
22. A. Deo, S. K. Dash, G. Suarez-Tangil, V. Vovk, and L. Cavallaro, "Prescience: Probabilistic guidance on the retraining conundrum for malware detection," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, AISec '16, (New York, NY, USA), pp. 71–82, ACM, 2016.
23. L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.