Ensemble Models for Data-driven Prediction of Malware Infections

Chanhyun Kang[†], Noseong Park[†], B. Aditya Prakash[*], Edoardo Serra[+], and V. S. Subrahmanian[†]
[*]Department of Computer Science, Virginia Tech., Blacksburg
[†]Department of Computer Science and UMIACS, University of Maryland, College Park
[+]Department of Computer Science, Boise State University, Boise
{chanhyun, npark}@cs.umd.edu, badityap@cs.vt.edu, edoardoserra@boisestate.edu, vs@umiacs.umd.edu

ABSTRACT

Given a history of detected malware attacks, can we predict the number of malware infections in a country? Can we do this for different malware and countries? This is an important question which has numerous implications for cyber security, right from designing better anti-virus software, to designing and implementing targeted patches to more accurately measuring the economic impact of breaches. This problem is compounded by the fact that, as externals, we can only detect a fraction of actual malware infections.

In this paper we address this problem using data from Symantec covering more than *1.4 million hosts* and 50 malware spread across 2 years and multiple countries. We first carefully design domain-based features from both malware and machine-hosts perspectives. Secondly, inspired by epidemiological and information diffusion models, we design a novel temporal non-linear model for malware spread and detection. Finally we present ESM, an ensemble-based approach which combines both these methods to construct a more accurate algorithm. Using extensive experiments spanning multiple malware and countries, we show that ESM can effectively predict malware infection ratios over time (both the actual number and trend) upto 4 times better compared to several baselines on various metrics. Furthermore, ESM's performance is stable and robust even when the number of *detected* infections is low.

Keywords

cyber security; malware attacks; prediction model; information diffusion; anti-virus;

1. INTRODUCTION

The goal of this paper is to develop methods to predict the percentage of hosts in a given population (we use country in our experiments) that will be infected by a particular piece of malware, given some historical data about the malware and the hosts, but with no information whatsoever on how the hosts are connected together. This is made even more challenging by the fact that not all truly infected machines are actually detected to be infected (by say using some anti-virus software). This problem is an important one with immediate applications in web and cyber security. For example, a better prediction of the number of infections in a country will enable anti-virus companies and security firms to better deploy patches and safety measures to counter threats.

In order to achieve this, we make several contributions. First, we define a very novel set of features that are related to the ability of hosts to detect malware and patch vulnerabilities. In order to achieve this, we present a novel host-malware bipartite graph and a bi-fix-point algorithm to compute these features. These lead to a feature-based prediction model (FBP). Then, building upon the well-known SIR-model of disease spread, we develop an epidemiologically-inspired predictive model called DIPS in which each host is either in a detected, infected, patched, or susceptible state. We define the model and show how to learn the parameters of the model in a data-driven way. We also define a variant of DIPS called DIPS-Exp. The outputs of these models, as well as outputs of past work on predicting epidemic spreads, are then fed into three different ensemble models: ESM_0, $\text{ESM}_1$and FBP^{+}_{Funnel}. We study the relative predictive accuracies of all of these models. On split-sample 10-fold cross validation tests, ESM_0 provides the best performance, significantly outperforming past work by large margins, irrespective of whether we use root mean squared error (RMSE), normalized RMSE (NRMSE), or Pearson Correlation Coefficient (PCC) as our measure. All the experiments use large-scale extensive real-data from Symantec's Worldwide Intelligence Network Environment (WINE) data set.

The paper is organized as follows. We describe our dataset and set-up in the next section. Section 3 describes statistical features we develop for our task and the feature based prediction model (FBP). In Section 4, we propose a novel non-linear time-series model (DIPS). The ensemble models (ESM) to combine two suggested models (FBP and DIPS) is described in Section 5. In Section 6 we present our empirical data analysis including the prediction task. We review related work in Section 7, and we discuss the implications of our results in Section 8.

2. DATASET AND SET-UP

Symantec's WINE data is collected from real-world hosts running Symantec anti-virus software. Users of Symantec's product line have a choice of opting-in to report telemetry

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.

DOI: http://dx.doi.org/10.1145/2835776.2835834

about the security events[1] (e.g. executable file downloads, virus detection) that occur on their hosts. The WINE dataset contains information about the hosts, files of the hosts and malware recognized on a file of a host.

We selected the 100 most frequently detected malware from the WINE data and extracted infection information about hosts infected by these malware using the following records.

Anti-virus telemetry. Anti-virus (AV) telemetry records detection history of known malware over hosts. From each record, we extracted the detection time and the hash (MD5 and SHA2) of the malicious file detected. Each record indicates that the anti-virus has blocked an attack that may have resulted in an infection.

Binary reputation. The binary reputation data records all binary executable files – benign or malicious – that were downloaded/copied on end-hosts worldwide. From each record, we extracted the creation time of the malicious file in the host (infection time) and the country where the host was when the file was infected.

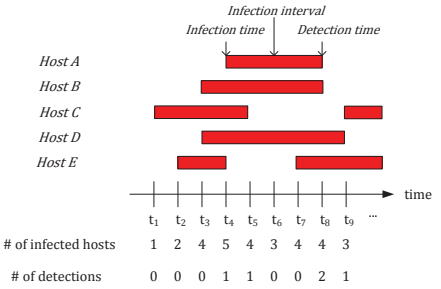


Figure 1: Examples of infections/infection intervals

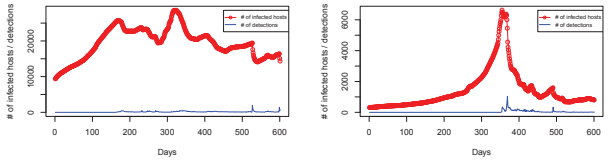


Figure 2: Examples for infection patterns of 2 malware in the US in our dataset. Number of infected hosts (bold red) and number of detection (blue) vs time (in days). Note that the number of detection are a fraction of the number of infected hosts and that both time-series are highly non-linear.

We note that an infection interval is the period from the infection time to the detection time of a malware in a host (See Figure 1). When the length of the infection interval is larger than 0, we call it an infection.[2] The number of infections over the 100 most frequently detected malware during 640 days was 3,170,669 (for the US alone, it was 989,263). The infection patterns are very different (See Figure 2). We adopt the following approach to address this problem: (a) we first carefully define and extract novel domain-based features from the WINE datasets including measures of hosts' detection ability and incompetence — we call this feature-based prediction (FBP) ; (b) we then construct a novel non-linear temporal model called DIPS (Detected, Infected, Patched, and Susceptible) based on epidemiological principles, and then (c) utilize sound regression methods to link both the features as well as the epidemiology-based models to the *dependent* variable (the number of malware infections). Because the inputs of FBP as well as multiple DIPS variants and FUNNEL[19] are themselves predictors that feed into yet another predictor, our overall framework is an ensemble prediction method.

[1]The events included in WINE are representative of events that Symantec observes around the world [23].

[2]If the anti-virus software blocked a file containing a malware immediately when the files appeared, then the detection record is not an infection in our framework.

3. FEATURE-BASED PREDICTION (FBP)

In a very dynamic environment like malware infections and patching, it is not possible to predict infection patterns solely based on infection trends. Some amount of domain knowledge has to be incorporated (for example, when the number of detected infections are small). Hence as a first step, we build a feature-based prediction model (FBP), where given past history we extract meaningful features of hosts and malware. Essentially these features encode the defending abilities of hosts and infection properties of the malware as relative scores.

Next, we describe the domain-based features we constructed for our analysis in more detail. We define $\mathcal{H}$ and $\mathcal{M}$ as the set of hosts and the set of malware, respectively. For each host $h \in \mathcal{H}$, $\mathcal{F}(h)$ denotes the set of files of h. For $f \in \mathcal{F}(h)$, t_f^h is the time stamp representing the time when f appears on h. For each host $h \in \mathcal{H}$, the malware detection set $dH(h)$ of h is a set of tuples of the form (f, m, t) where $f \in \mathcal{F}(h)$, $m \in \mathcal{M}$ and t is the time stamp when the file f infected by the malware m is discovered on the host h. By using the malware detection sets for each host, we define the set of host detection $dM(m)$ of the malware m as $dM(m) = \{(f, h, t) | h \in \mathcal{H}, f \in \mathcal{F}(h), (f, m, t) \in dH(h)\}$.

Vulnerabilities: Let V be the set of all vulnerabilities. Given a malware $m \in \mathcal{M}$ the set $Vu(m)$ is the set of all vulnerabilities used by this malware. For each vulnerability $v \in V$, two time stamps are important [5]:

- **Anti-virus signatures released.** Once the vulnerability is disclosed, anti-virus vendors release new signatures for ongoing attacks and created heuristic detection for the exploit. After this point, the attacks can be detected on end-hosts with updated A/V signatures. We denote this as t_v^s. We can retrieve this information from Symantec website or the National Vulnerabilities Database (NVD)[3] maintained by NIST (National Institute of Standards and Technology of the US Government).
- **Patch released.** On the disclosure date, or shortly afterward, the software vendor releases a patch for the vulnerability. After this point, the hosts that have applied the patch are no longer susceptible to the exploit. We denote this as t_v^p. This information is readily available from NVD.

3.1 Incompetence

Given a host $h \in \mathcal{H}$ and a tuple $(f, m, t) \in dH(h)$, we define two classes of features: hosts' incompetence in detecting malware and incompetence in patching (DI and PI, respectively). For each, we consider an absolute case and a relative case (i.e ADI, API, RDI and RPI). These four measures measure the incompetence of a host in detecting and patching the malware m associated with file f. The

[3]https://nvd.nist.gov/

only difference between the relative case and absolute case is that the absolute case does not consider the time t_v^s for each vulnerability v contained in a malware.

$$\begin{aligned} ADI(h,f,m,t) &= t - t_f^h \\ RDI(h,f,m,t) &= \max_{v \in Vu(m)} \max(0, t - \max(t_v^s, t_f^h)) \\ API(h,m) &= (\max_{(f,m,t) \in dH(h)} t) - (\min_{(f,m,t) \in dH(h)} t_f^h) \\ RPI(h,m) &= \max_{v \in Vu(m)} \max(0, (\max_{(f,m,t) \in dH(h)} t \\ &\quad - \max(t_v^p, (\min_{(f,m,t) \in dH(h)} t_f^h)))) \end{aligned}$$

For instance, absolute detection incompetence, $ADI(h,f,m,t)$ of host h w.r.t. a file f of malware m and the time t when f was detected by host h is simply $(t - t_f^h)$, i.e. the time that passed from the detection time to the time of original infection. The relative detection incompetence $RDI(h,f,m,t)$ looks at all possible vulnerabilities v that malware m exploits. For each such vulnerability, it looks at the time elapsed between when the file f was detected by host h and when either the file infected the host or when an updated anti-virus signature could have detected it. The longer this time gap, the more incompetent the host. The absolute and relative (API, RPI) patching incompetence are similarly defined.

Estimation by using only the WINE *datasets.* The relative detection and patch incompetence formulas can be rewritten as:

$$\begin{aligned} RDI(h,f,m,t,) &= \max(0, t - \max((\min_{v \in Vu(m)} t_v^s), t_f^h))) \\ RPI(h,m) &= \max(0, (\max_{(f,m,t) \in dH(h)} t - \\ &\quad \max((\min_{v \in Vu(m)} t_v^p), (\min_{(f,m,t) \in dH(h)} t_f^h)))) \end{aligned}$$

For a malware m, $t^s(m) = \min_{v \in Vu(m)} t_v^s$ denotes the earliest time at which a host downloaded an anti-virus signature for the vulnerability (we call this "time signature"), showing that it might have become aware of the malware, and $t^p(m) = \min_{v \in Vu(m)} t_v^p$ denotes the earliest time the host applied a patch for a vulnerability for malware m (we call this "patch signature"). Both these quantities are recorded by WINE. Then, the two measures becomes the following:

$$\begin{aligned} RDI(h,f,m,t,) &= \max(0, t - \max(t^s(m), t_f^h)) \\ RPI(h,m) &= \max(0, (\max_{(f,m,t) \in dH(h)} t) - \\ &\quad \max(t^p(m), (\min_{(f,m,t) \in dH(h)} t_f^h)))) \end{aligned}$$

A possible way to estimate time signature and patch signature of a malware in WINE is:

$$\begin{aligned} t^s(m) &= \min_{v \in Vu(m)} t_f^m \\ &\approx \min_{h \in \mathcal{H}, (f,m,t) \in dH(h)} t \\ t^p(m) &= \min_{v \in Vu(m)} t_v^p \\ &\approx \min_{h \in \mathcal{H}, (,m',t') \in dH(h)} \max_{(f,m,t) \in dH(h)} t \end{aligned}$$

Average Incompetence. Average/relative detection/patching incompetence of a host h can be defined by aggregation.

$$\begin{aligned} AADI(h) &= \frac{\sum_{(f,m,t) \in dH(h)} ADI(h,f,m,t)}{|dH(h)|} \\ ARDI(h) &= \frac{\sum_{(f,m,t) \in dH(h)} RDI(h,f,m,t)}{|dH(h)|} \\ AAPI(h) &= \frac{\sum_{m \in \{m' | (,m',) \in dH(h)\}} API(h,m)}{|\{m' | (,m',) \in dH(h)\}|} \\ ARPI(h) &= \frac{\sum_{m \in \{m' | (,m',) \in dH(h)\}} RPI(h,m)}{|\{m' | (,m',) \in dH(h)\}|} \end{aligned}$$

For instance, average patch incompetence (AAPI) is obtained by adding up the average patch incompetence of the host w.r.t. all malware files that it patched, and dividing this by the number of such malware files associated with that host.

3.2 Features on Bipartite Host-Malware (HM) Graphs

Given a training set of historical WINE data, we can derive a bipartite graph whose vertices are hosts and malware, respectively. An edge links a host and malware if the training data shows that the malware has infected the host. Edges can be weighted with quantities like ADI, RDI, API, and RPI as defined earlier in this section.

We now define some novel features using HM-graphs. The absolute detection ability (ADA) of a host h captures the host's ability to detect malware, while the absolute detection hardness (ADH) of a malware m captures the difficulty in detecting that malware. Clearly, these two quantities are closely intertwined. We define them below in a mutually recursive manner.

$$\begin{aligned} ADA(h) &= \sum_{(f,m,t) \in dH(h)} w_{12}(h,f,m,t) \cdot ADH(m) \\ ADH(m) &= \sum_{(f,h,t) \in dM(m)} w_{21}(m,f,h,t) \cdot ADA(h) \end{aligned}$$

where $h \in \mathcal{H}, m \in \mathcal{M}$ and

$$\begin{aligned} w_{12}(h,f,m,t) &= \frac{\frac{1}{ADI(h,f,m,t)}}{\sum_{(f',h',t') \in dM(m)} \frac{1}{ADI(h',f',m,t')}} \\ w_{21}(m,f,h,t) &= \frac{ADI(h,f,m,t)}{\sum_{(f',m',t') \in dH(h)} ADI(h,f',m',t')} \end{aligned}$$

By way of explanation, consider the definition of $ADA(h)$. To do this, we look at every (f,m,t) triple in the training set. If h can detect m, then the harder m is to detect (captured by $ADH(m)$ in the formula), the better h's detection abilities are. Thus, when considering a specific (f,m,t) triple, the definition of $ADA(h)$ multiplies $ADH(m)$ by a quantity w_{12} which is the relative competence (inverse of absolute detection incompetence $ADI(h,f,m,t)$). A similar rationale applies to the definition of ADH. As in the previous section, we can define relative variants, RDA, RDH as well.

$$\begin{aligned} RDA(h) &= \sum_{(f,m,t) \in dH(h)} w_{12}(h,f,m,t) * RDH(m) \\ RDH(m) &= \sum_{(f,h,t) \in dM(m)} w_{21}(m,f,h,t) * RDA(h) \end{aligned}$$

where $h \in \mathcal{H}, m \in \mathcal{M}$ and

$$\begin{aligned} w_{12}(h,f,m,t) &= \frac{\frac{1}{RDI(h,f,m,t)}}{\sum_{(f',h',t') \in dM(m)} \frac{1}{RDI(h',f',m,t')}} \\ w_{21}(m,f,h,t) &= \frac{RDI(h,f,m,t)}{\sum_{(f',m',t') \in dH(h)} RDI(h,f',m',t')} \end{aligned}$$

3.2.1 Fix-point Computation and Convergence Result

As ADA, ADH are mutually recursively defined, one may wonder if they are well-defined. We show below, an iterative fix-point computation procedures that describes how this may be achieved. We now define the iterations ADA^i, ADH^i as follows. Let $ADA^0(h) = \frac{1}{|\mathcal{H}|}$, $ADH^0(m) = \frac{1}{|\mathcal{M}|}$ for $h \in \mathcal{H}, m \in \mathcal{M}$,then for iteration $i > 0$,

$$\begin{aligned} ADA^i(h) &= \sum_{(f,m,t) \in dH(h)} w_{12}(h,f,m,t) * ADH^{i-1}(m) \\ ADH^i(m) &= \sum_{(f,h,t) \in dM(m)} w_{21}(m,f,h,t) * ADA^{i-1}(h) \end{aligned}$$

We can capture the above computations as a bi-fix-point algorithm, Algorithm 1 that maintains two lists ADA, ADH. The entry $ADA(h)$ specifies the value of the ADA quantity computed thus far - similarly for the entry $ADH(m)$. These two lists are manipulated. We assume the existence of an equivalence relation $\equiv$ on ADA and ADH lists, telling us when two ADA lists and two ADH lists are essentially the same. For instance, we can decide that ADA lists ℓ_1, ℓ_2 are

similar whenever a measure of distance, such as cosine distance or Jaccard distance, between the two vectors is below a threshold.

Algorithm 1: BiFixpoint

Input : $\mathcal{H}, \mathcal{M}, T$ ($*T$ is a training set $*$)
Output: ADA, ADH

```
forall h ∈ H, ADA(h) ← 1/|H| (* initialize *)
forall m ∈ M, ADH(m) ← 1/|M|
change ← true;
while change do
    ADA'(h) ← Σ_{(f,m,t)∈dH(h)} w12(h,f,m,t) * ADH(m)
    ADH(m) ← Σ_{(f,h,t)∈dM(m)} w21(m,f,h,t) * ADA(h)
    if ADA' ~ ADA and ADH' ≡ ADH then
        change ← false
    else
        ADA ← ADA' and ADH ← ADH'
    end
end
return ADA, ADH
```

The following result shows that this algorithm is guaranteed to converge.

THEOREM 1. *For any sets $(\mathcal{H}, \mathcal{M}, T)$, $BiFixpoint(\mathcal{H}, \mathcal{M}, T)$ is guaranteed to converge and the returned solution is unique.*

PROOF. The recursive equations describing RDA and RDH have the same structure of those of a bipartite Markov Chain where the first partition is the set of hosts $\mathcal{H}$ and the second is the set of malware $\mathcal{M}$. In fact we can rewrite the equations in the following way $RDA(h) = \sum_{m\in\{m'|(,m',)\in dH(h)\}} p_{12}(h,m) * RDH(m)$ and $RDH(m) = \sum_{h\in\{f|(,h,)\in dM(m)\}} p_{21}(m,h) * RDA(h)$ where $p_{12}(h,m) = \sum_{(f,m,t)\in dH(h)} w_{12}(h,f,m,t)$ and $p_{21}(m,h) = \sum_{(f,h,t)\in dM(m)} w_{21}(m,f,h,t)$. Because of the definition of the weights w_{12} and w_{21}, it follows that for each $m \in \mathcal{M}$, $\sum_{h\in\{h'|(,h',)\in dM(m)\}} p_{12}(h,m) = 1$ and for each $h \in \mathcal{H}$, $\sum_{m\in\{m'|(,m',)\in dH(h)\}} p_{21}(m,h) = 1$. In addition both the $p_{1,2}(h,m)$ (with $m \in \mathcal{M}$ and $h \in \{h'|(,h',) \in dM(m)\}$) and $p_{2,1}(m,h)$ (with $h \in \mathcal{H}$ and $m \in \{m'|(,m',) \in dH(h)\}$) are greater than zero. Moreover, by construction, if we have an edge (h,m) also the edge (m,h) exists, both with value (probability) greater than zero. The last statement implies that for each pair of nodes (a,b) in each connected components in this Markov chain the probability to reach a from b (and vice versa) after a finite number of steps is greater than zero. This implies that, for each connect component, the associate transition matrix is regular, it follows that the $BiFixpoint(\mathcal{H}, \mathcal{M}, T)$ algorithm converges and the returned solution is unique. □

3.3 Patch Ability

Exactly analogous to the measures defined above, we can define two patch abilities: absolute and relative. The absolute patch ability (APA) and absolute patch hardness (APH) are defined in the following way:

$$\begin{aligned} APA(h) &= \sum_{m\in\{m'|(,m',)\in dH(h)\}} w_{12}(h,m)\cdot APH(m) \\ APH(m) &= \sum_{h\in\{h'|(,h',)\in dM(m)\}} w_{21}(m,h)\cdot APA(h) \end{aligned}$$

where $h \in \mathcal{H}$, $m \in \mathcal{M}$, and

$$\begin{aligned} w_{12}(h,m) &= \frac{\frac{1}{API(h,m)}}{\sum_{h''\in\{h'|(,h',)\in dM(m)\}} \frac{1}{API(h'',m)}} \\ w_{21}(m,h) &= \frac{API(h,m)}{\sum_{m''\in\{m'|(,m',)\in dH(h)\}} API(h,m'')} \end{aligned}$$

The relative patch ability (APA) and relative patch hardness (APH) are defined in the following way:

$$\begin{aligned} RPA(h) &= \sum_{m\in\{m'|(,m',)\in dH(h)\}} w_{12}(h,m) * RPH(m) \\ RPH(m) &= \sum_{h\in\{h'|(,h',)\in dM(m)\}} w_{21}(m,h) * RPA(h) \end{aligned}$$

where $h \in \mathcal{H}$, $m \in \mathcal{M}$ and

$$\begin{aligned} w_{12}(h,m) &= \frac{\frac{1}{RPI(h,m)}}{\sum_{h''\in\{h'|(,h',)\in dM(m)\}} \frac{1}{RPI(h'',m)}} \\ w_{21}(m,h) &= \frac{RPI(h,m)}{\sum_{m''\in\{m'|(,m',)\in dH(h)\}} RPI(h,m'')} \end{aligned}$$

In order to compute the patch ability we can use the $BiFixpoint(\mathcal{H}, \mathcal{M}, T)$ algorithm with opportune modifications according the specific equations. By following the same proof methodology of Theorem 1, is possible to prove that the resulting algorithm converge and the returned solution is unique.

3.4 Collaborative Features

Let H be a set of all hosts and H' be a subset of H. We use a memory based collaborative filtering approach to produce features whose meaning is close to the number of malware infections in a specific set of hosts H', i.e. all the hosts of a specific country. The basic intuition under this collaborative filtering is that if host h is infected by malware m and host h' similar to h, then h' could also be infected by m. We assume the existence of a function $sim : H \times H \to [0,1]$ that similarity between hosts. Given a host $h \in H'$, the percentage of infected hosts in H (weighted by the similarity) by malware m is

$$infected(h,m) = \frac{\sum_{h'\in H,\exists(,m,)\in dH(h')} sim(h,h')}{|H|} \qquad (1)$$

where $\exists(,m,) \in dH(h')$ is used to check if h is infected by m. Intuitively $infected(h,m)$ estimates the possibility that the host will be infected by m. In fact, the greater the percentage of hosts similar to h infected by m, the greater is the possibility (or probability) that h will be infected by m. Since we want to predict the expected number of infected hosts in H', our resulting variable is the following:

$$\sum_{h\in H'} infected(h,m)$$

If we assume that $infected(h,m)$ is a correct estimation of the probability that the host h will be infected by m, the value $\sum_{h\in H'} infected(h,m)$ estimates the expected number of infected host in H'. Since H can be huge (millions of hosts), we consider a reduced subset $\tilde{H}$ obtained by uniformly sampling H, and then we replace H in Eq 1 by $\tilde{H}$.

Similarity Measures. We define different notions of similarity – each of these notions yields a different feature estimating the expected number of infected hosts. The definitions below show some examples.

$$\begin{aligned} sim1(h,h') &= 1 - |ADA(h) - ADA(h')| \\ sim2(h,h') &= 1 - |RDA(h) - RDA(h')| \\ sim3(h,h') &= 1 - |APA(h) - APA(h')| \\ sim4(h,h') &= 1 - |RPA(h) - RPA(h')| \\ sim5(h,h') &= \\ 1 &- \sqrt{\frac{(ADA(h)-ADA(h'))^2+(APA(h)-APA(h'))^2}{2}} \\ sim6(h,h') &= \\ 1 &- \sqrt{\frac{(ADA(h)-RDA(h'))^2+(RPA(h)-RPA(h'))^2}{2}} \end{aligned}$$

3.5 The FBP Model

The FBP Model is a general framework to predict the ratio of hosts infected by malware m in country c with a given regression method r such as GPR (Gaussian Process Regression), Elastic Net, Ridge, Lasso, and etc. The dependent and independent variables for the regression method are following:

Dependent variable. The dependent variable for each country -malware pair (c, m) is the percentage of hosts in country c infected by malware m.

Independent variables (features). For each country-malware pair (c, m), we compute ADI, API, RDI, RPI, $AADI$, $ARDI$, $AAPI$, $ARPI$, ADA, RDA, APA and RPA of hosts in country c, APH and RPH of malware m and the *collaborative variables* of the hosts in c with the 6 similarity measures. Other features are the percentages of the hosts which were not detected for m in countries which are economically similar to country c.[4] Two measures, Human Development Index (HDI) and Per Capita GDP PPP of countries are used to determine the similar countries. The features are computed for each day d with respect to four different time intervals from 0, $d-60$, $d-30$, $d-10$ to d day.

Nearest Neighbors (additional feature selection) To make the learned model more robust, we also take features of the $k-$nearest neighbors. For a given country-malware pair (c,m) and similarity measure S, we find the k *most similar* country-malware pair(s) P_k to (c,m) with respect to the past infection pattern. We then use the features of P_k as well (in addition to those of (c,m)) for predicting. Various existing techniques can be used for measuring similarity between two time-series e.g. Dynamic Time Warping (DTW), Euclidean distance etc.

4. TIME-SERIES MODEL BASED PREDICTION (DIPS)

In this section we propose a novel non-linear time-series model called DIPS (short for Detected, Infected, Patched, Susceptible) which is a variant of the SIR model of disease spread tailored for malware. Typical infection patterns (see Figure 2) are highly non-linear. Hence intuitively linear approaches are ill-suited to this task. Malware spread has multiple similarities to information diffusion and epidemiology (indeed, some of the earliest papers on this topic try to model it as a disease). Inspired by these models, DIPS adapts a recent model [19] to handle malware infections. The most challenging part is to deal with difference in the *detected* and *actual* infections. We assume that in the training dataset we have the true (as well as the detected) number of infections but in the test dataset, we are only given the number of actual detections.

4.1 Model Parameters

Similar to fundamental models like the Susceptible-Infected-Recovered (SIR) 'mumps-like' model, our model assumes that the machine population can be in one of the following states:

- *Susceptible:* In this state, a machine is not patched and hence is vulnerable to an attack by a particular malware.
- *Infected:* In this state, the machine is now infected, and spreads the malware by infecting other susceptible hosts.
- *Detected:* In this state, we have detected that the machine is infected (by say an anti-virus software running on the machine). Note that not all infected machines may be detected as infected.
- *Patched:* As the name suggests, in this state the machine is patched, and cannot get infected.

We temporally track the number of hosts in each of these states. In addition, we assume we have a variable $DET(t)$ which is the true number of detection at time t.

4.2 The DIPS Model

We are now ready to describe the DIPS model. There are a total of N potential susceptible hosts for the malware (one may also assume that there is an inflow of additional hosts at every time tick). Each host starts out in the Susceptible state. From the susceptible state, a host may move into the Patched state (at the rate of $\theta(t)$) or the Infected state. We assume each infected host may infect a susceptible host at a certain rate (called $\beta(t)$), which is also periodic and dictated by users' habitual behavior. An infected host may move into the Detected state upon detection, after which the host may become susceptible again (if it is not patched), or be patched (at the rate of $\delta(t)$) and never get infected again — these transitions are shown in Figure 3.

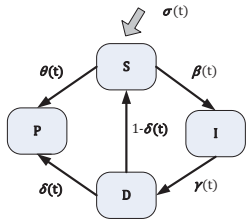


Figure 3: DIPS model. D, I, P and S are the Detected, Infected state, Patched and Susceptible states

Infection Rate $\beta(t)$: This is the rate at which an infected host infects susceptible machines. We assume $\beta(t)$ has the following form:

$$\beta(t) = \beta_0(1 + P_a \cos(\frac{2\pi}{P_p}(t + P_S))) \quad where,$$

- β_0: is the infection rate by users' (who are using the hosts) habitual behavior averaged over days
- P_p: stands for the period of a cycle. Infection data, like information diffusion data, and epidemiological data, typically shows obvious periodicity (like weekly) which we want to capture.
- P_a: represents the amplitude of the fluctuation.
- P_S: represents the phase shift in the cycle.

Patching rates $\delta(t)$ and $\theta(t)$: As mentioned above, we assume that susceptible hosts get patched at the same rate $\theta(t)$ and hosts in the detected state get patched at rate $\delta(t)$. More formally:

$$\theta(t) = \quad 0 \ (t < t_p) \quad or \ \ \theta_0 \ (t \geq t_p)$$
$$\delta(t) = \quad 0 \ (t < t_p, t_d) \quad or \ \ \delta_0 \ (t \geq t_p, t_d)$$

where t_p is the start time of the patching and t_d is the first detection time.

[4]We define that country C and C' are similar w.r.t. measure T when $|T(C) - T(C')|/(T(C) + (T(C')) * 0.5 \leq p$ where $p = \{3\%, 6\%, 9\%, 12\%, 15\%, 18\%\}$.

Error rate $\gamma(t)$: This captures the fact that the number of true detections $DET(t)$ may not exactly match the number of detection in our model (i.e. the number of hosts in the $D(t)$ state).

Finally, putting everything together, the DIPS model can be expressed using the following differential equations:

$$\begin{aligned} S(t+1) &= S(t) - \beta(t)S(t)I(t) + (1-\delta(t))D(t) - \theta(t)S(t) \\ I(t+1) &= I(t) + \beta(t)S(t)I(t) - \gamma_0 DET(t) \\ D(t+1) &= \gamma_0 DET(t) \\ P(t+1) &= P(t) + \delta(t)D(t) + \theta(t)S(t) \end{aligned} \quad (2)$$

Intuitively, $S(t), I(t), D(t), P(t)$ capture the number of hosts in a susceptible, infected, detected, and patched state, respectively.

The DIPS*-exp model.*

We have another variant of the DIPS model, called DIPS-exp where we have an inflow of additional susceptible hosts at every time-tick ($\sigma(t)$). All the equations in Equation 2 remain the same, except for the equation defining $S(t+1)$ which changes to:

$$S(t+1) = S(t) - \beta(t)S(t)I(t) + (1-\delta(t))D(t) - \theta(t)S(t) + \sigma(t)$$

4.3 Model Learning

The parameters of the DIPS model $\Theta = \{N, \beta_0, P_a, P_S, \gamma_0, t_p, \theta_0, \delta_0\}$ (DIPS-exp additionally has a parameter of σ_0) need to be learned. Given two real time-sequences, $I_{true}(t)$ and $D_{true}(t)$, of the number of actual infections and detection (resp.) at time-tick t, we find the parameters which optimize the following equation:

$$\Theta^* = \arg\min_{\Theta} \sum_t (I_{true}(t) - I_\Theta(t))^2.$$

The learning algorithm is given in Algorithm 2. Basically, there are two phases to learn the parameters. The first step is to adjust the initial parameters to fit them on overall infection time-sequences ($I^{sum}_{true}(t)$ and $D^{sum}_{true}(t)$) and fix parameter t_p,θ_0, P_a and P_S. The learning order of the parameters is $N \rightarrow \beta_0 \rightarrow \gamma_0 (\rightarrow \sigma_0) \rightarrow t_p \rightarrow \delta_0 \rightarrow \theta_0 \rightarrow P_a \rightarrow P_S$[5]. The second step is to fit the learned parameters on the specific infection pattern of country c for malware m. Here, we only update the following parameters: $N \rightarrow \beta_0 \rightarrow \gamma_0 (\rightarrow \sigma_0) \rightarrow \delta_0$. The $LearnParameters(\Theta, I(t), D(t), flag)$ is the function to update the given parameter values Θ applying parameter $p \in \Theta \leftarrow \arg\min_p \sum_t (I(t) - I_\Theta(t))^2$ and to return the updated parameters, Θ' and the error, $\sum_t (I(t) - I_{\Theta'}(t))^2$. The flag represents the learning order of each step (if $flag = 1$, the first step, else the second step). Initially, $S(0) = N - D(0) - I(0), P(0) = 0, D(0) = DET(0)$, and $I(0) = I_{true}(0)$. We leverage the Levenberg-Marquardt (LM) algorithm [16] for each *argmin* in *LearnParameters*.

Algorithm 2: DIPS learning algorithm

Input : country c, malware m and $X(m) = \{(c', I_{true}(t), D_{true}(t)) | c' \in Countries, I_{true}(t)$ and $D_{true}(t)$ for c' and $m\}$
Output: $\Theta = \{N, \beta_0, P_a, P_S, \gamma_0, t_p, \theta_0, \delta_0, (\sigma_0)\}$

$\Theta \leftarrow \{N', \beta_0', P_a', P_S', \gamma_0', t_p', \theta_0', \delta_0', (\sigma_0')\}$ // Assign default parameter values
$I^{sum}_{true}(t), D^{sum}_{true}(t) \leftarrow \sum_{c' \in Countries, (c', I(t), D(t)) \in X(m)} I(t), D(t)$ // Sum of # of infections/detection over countries w.r.t. each time
$err \leftarrow infinite, err' \leftarrow 0, \Theta' \leftarrow \Theta$
while $err > err'$ **do**
  // Learn all parameters
  $< err', \Theta' > \leftarrow LearnParameters(\Theta', I^{sum}_{true}(t), D^{sum}_{true}(t), 1)$
  if $err > err'$ **then**
    $\Theta \leftarrow \Theta', err \leftarrow err'$
    $err' \leftarrow 0$
  end
end
Let $(c', I_{true}(t), D_{true}(t)) \in X(m)$ where c' equals c
$\Theta' \leftarrow \Theta, err' \leftarrow 0$
while $err > err'$ **do**
  // Learn $N, \beta_0, \gamma_0(\sigma_0), \delta_0$ only
  $< err', \Theta' > \leftarrow LearnParameters(\Theta', I_{true}(t), D_{true}(t), 0)$
  if $err > err'$ **then**
    $\Theta \leftarrow \Theta', err \leftarrow err'$
    $err' \leftarrow 0$
  end
end
return Θ

5. ENSEMBLE MODELS (ESM)

We have suggested three models: feature based prediction (FBP), and time-series model based prediction (DIPS and DIPS-exp). We now propose ensemble models, ESM which combine both of them to improve performance. We add the expected number of infections we get from the DIPS and DIPS-exp models as features for our feature prediction model. We do not use the *learned* parameters Θ^* of DIPS directly as features. This is because DIPS is highly non-linear; to take advantage of it, we use the *output* of DIPS instead.

To summarize, we extract all the features, add the output of DIPS as additional features, and then use a regression model to perform prediction. We used a variety of models like logistic regression and Gaussian process regression.

[5]The order affects the prediction performance a lot.

6. EMPIRICAL ANALYSES

In this section, we empirically analyze the performance of the proposed prediction methods. Specifically, we study three questions:

- Q1: Can we predict the number of hosts in a country that are infected by a malware?
- Q2: How does predictive accuracy change with the number of infections?
- Q3: Does the prediction performance of the models depend on the number of hosts monitored?

Experiment setting. We focused on top 50 most infectious malware in the 100 malware infection data extracted from WINE, and the top 40 countries (by GDP), leading to 2000 (50x40) country-malware pairs. This data includes over 1.45M unique hosts and 2.99M infections. The number of reported hosts varies from just 634 (Iran) to 554,969 (United States). We ran the experiment over 541 days from 60th to 600th day among total 640 days. Because infection data is not complete for the head and the tail periods, we eliminated the first 60 and last 40 days of the 640-day data set.

Performance metrics. We checked RMSE (root mean squared error), NRMSE (normalized RMSE), and Pearson correlation coefficient (PCC) to evaluate performance. $NRMSE$

$= RMSE\ /(max(X) - min(X))$, where X is a set of ground true infection ratios of each malware-country pair over all days. Due to space constraints, we focus on NRMSE[6].

Train/Test Split. Each prediction model was trained in the first l days (on the features defined earlier) and tested for the remaining days. We used 80% of the data for training and the rest for validation.

Prediction models. We tested 7 different prediction models, FUNNEL [19][7], FBP, DIPS, DIPS-EXP, two different ESM models (ESM_0 and ESM_1) and FBP^+_{Funnel}[8]. FUNNEL is a recently proposed time-series model which has been used to predict large scale disease patterns (and like DIPS, it too is based on epidemiological models like SEIR). However, intuitively, FUNNEL is too tailored to model disease trends (like taking into account strong seasonality and locality) and not malware attacks. For example, it does not distinguish between detected and patched states, and its parameters are not suitable to represent malware infection. ESM_0 refers to ESM using FBP with the expected values of the two DIPS models as additional features for each pair. ESM_1 uses FBP with FUNNEL and DIPS models. To estimate the performance of ESM more precisely, we also tested FBP^+_{Funnel} which uses FBP with only FUNNEL (but without DIPS).

For FBP, we tested various regression methods such as GPR (Gaussian Process Regression), Elastic Net, Ridge, Lasso, and etc with k more country-malware pairs whose infection patterns are similar with each country-malware. We tested with $k \in \{0, 1, 2\}$. The similarity was measured by Dynamic Time Wrapping (DTW). For FBP, DTW showed best performance in our testing among several similarity measures, such as Euclidean distance, average correlation coefficient, DTW and etc. In testing with 2000 (country,malware) pairs, FBP showed the best performance with GPR and $k = 2$ w.r.t. the average RMSE and NRMSE. The average correlation coefficients were similar for all. We did not consider $k > 2$ because the performance improvement from $k = 1$ to $k = 2$ was very trivial. In the following experiment sections, we fix k = 2 and GPR for FBP and the FBP setting was used for ESM_0, ESM_1 and FBP^+_{Funnel} also.

6.1 Performance over all countries and malware

Answering Q1 with overall performance. The overall performance of the prediction models is shown in Table 1. FBP is clearly better than FUNNEL with respect to all four measures, showing that the features defined in Section 3 are useful to predict infection patterns generally. Also, DIPS shows good prediction performance. The average MAE*, RMSE and NRMSE are 25%, 30% and 73% respectively of FUNNEL. Moreover the average correlation coefficient is more than 6 times that of FUNNEL. ESM models (FBP+DIPS) shows significant performance improvement from FBP and the best performance with respect to NRMSE. DIPS models show slightly better prediction performance w.r.t. MAE*, RMSE and Correlation coefficient than ESM_0 and ESM_1. We can also confirm that FBP^+_{Funnel} (FBP with FUNNEL only) and

[6]The other two measures provide similar results.

[7]The code was downloaded from http://www.cs.kumamoto-u.ac.jp/∼yasuko/software.html

[8]The source code for the models suggested in this paper are available at http://www.umiacs.umd.edu/research/CDIG/projects/CCAFE.html

Model	MAE*	RMSE	NRMSE	C.C.
FBP	73.74	0.00170	0.179	0.12
FUNNEL	127.83	0.00269	0.226	0.08
DIPS	32.36	0.00083	0.165	0.50
DIPS-EXP	36.56	0.00096	0.223	0.48
ESM_0	39.41	0.00115	0.150	0.31
ESM_1	41.84	0.00118	0.151	0.31
FBP^+_{Funnel}	79.01	0.00189	0.179	0.19

Table 1: Average of MAE, RMSE, NRMSE and Correlation coefficient over 2000 pairs in testing period. Training ratio is 0.8. The MAE* values were computed with |# of ground true infected hosts - the expected # of infected hosts|. Others were based on the infection ratio of each country.

ESM_1 (ESM_0 with FUNNEL) do not show clear improvement from FBP and ESM_0. This suggests that DIPS significantly improves upon the predictive accuracy of past work. In any case we wish to point out that FUNNEL was designed and intended for disease patterns and not malware spread.

More detailed analysis: Performance of the prediction models for each country-malware pair is shown in Figure 4. DIPS models show many cases having very high correlation coefficient and very low NRMSE. But there are some cases having very high NRMSE when the correlation coefficient is very low. On the other hand, NRMSE of FBP is relatively low even when the correlation coefficient is pretty low. Interestingly, ESM models have the both strength of FBP and $DIPS$. To analyze the performance more, kernel density estimations are shown in Figure 5. As you see in Figure 5 (a), the density of ESM models and DIPS models are high in very high correlation coefficient values. In Figure 5 (b), the density of DIPS models and ESM models are really high with very low NRMSE values. But DIPS models have very long tail having very high NRMSE values.

Answering Q2. We define the infectiousness level as the average number of infected hosts over all days. Precisely, $ln(1 + Avg.\#\ infected\ hosts\ over\ days)$, here 1 is added as a default because $Avg.\#\ infected\ hosts\ over\ days$ could be less than 1. In Figure 6, we projected values of correlation coefficients and NRMSE over the infectiousness level w.r.t. each country-malware pair in order to check how each model works in different infectiousness levels. Each dot corresponds to a certain country-malware pair — the infectiousness level varies a lot. In terms of correlation coefficient, DIPS shows the best performance across all infectiousness levels especially with very high values in highly infectiousness levels, and ESM_0 follows after it. Switching the perspective to NRMSE more interestingly, DIPS still shows very good performance in general but some glitches in relatively lower infectiousness levels. This shows the performance of DIPS may be less robust when the number of infections is low. On the other hand, NRMSE of FBP is still low in the situation. This means the two models are very complementary to each other. So, ESM models show very stable and reliable performance regardless of infectiousness levels for both measures. Shortly, FUNNEL doesn't show any good performance even in highly infectiousness levels.

6.2 Performance for certain countries

Answering Q3. The number of reported hosts over countries in our data is very different. Figure 7 shows the performance of the models according to each country. The order

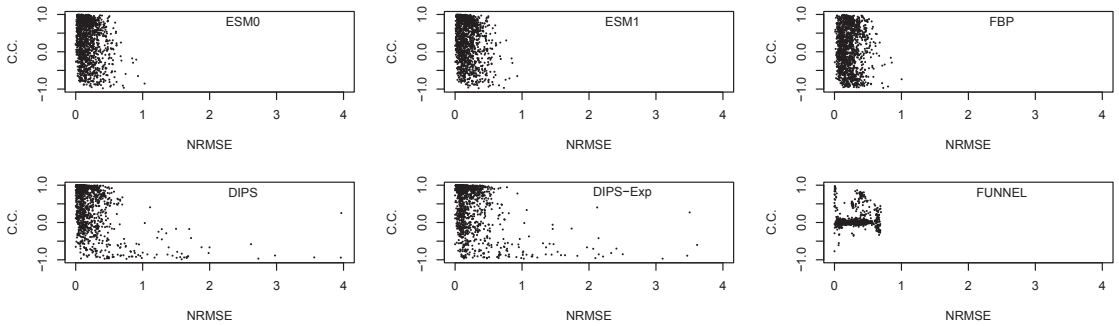
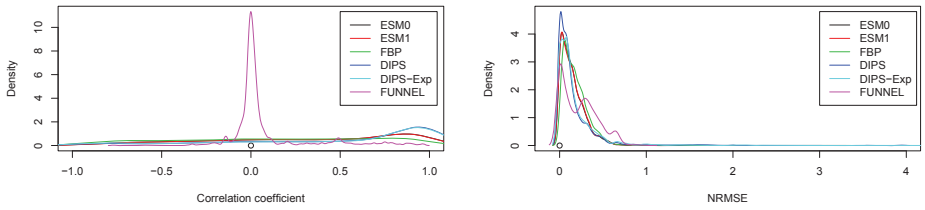
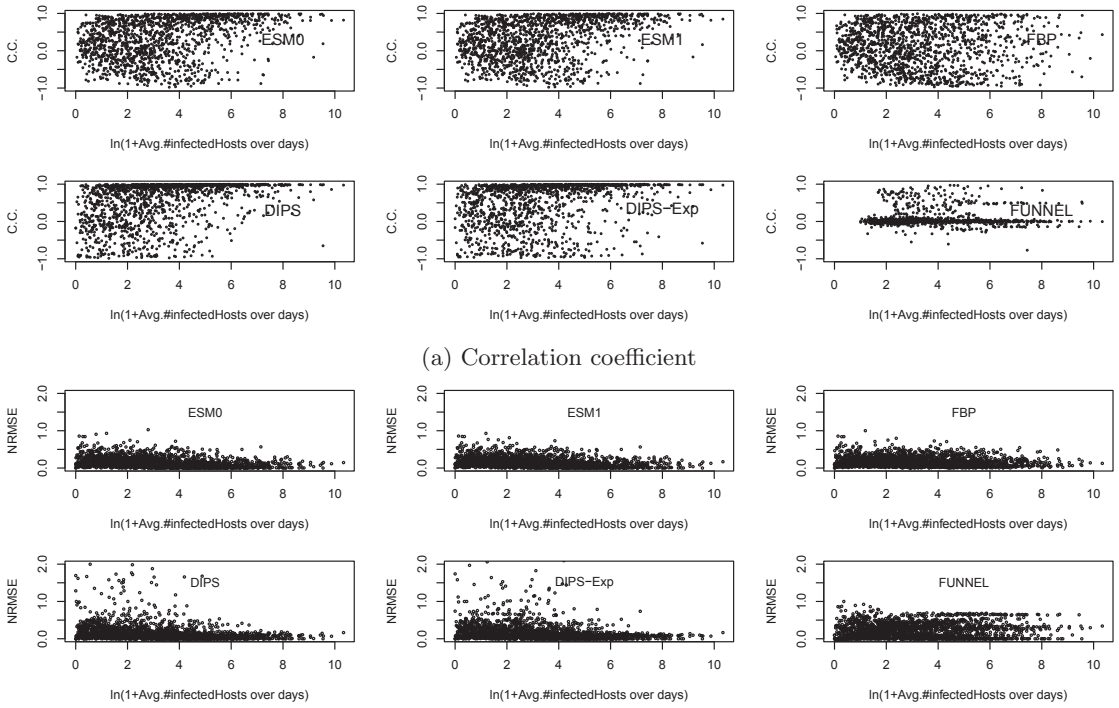


Figure 4: Scatter plot for NRMSE and Correlation coefficient over 2000 pairs. Each point represents the performance for each country-malware pair.



(a) Correlation coefficients (b) NRMSE

Figure 5: Kernel Density of (a) Correlation coefficient and (b) NRMSE over 2000 pairs



(a) Correlation coefficient

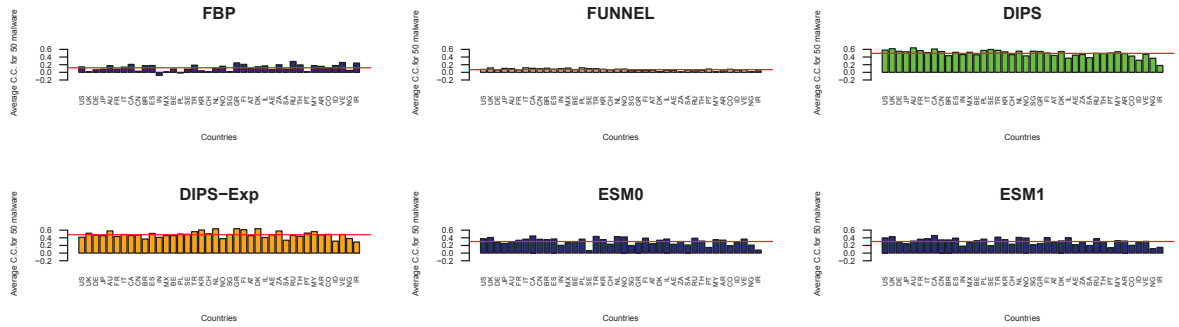
(b) NRMSE

Figure 6: (a) Correlation coefficient and (b) NRMSE by ln(1+ Avg.# infected hosts over all days)

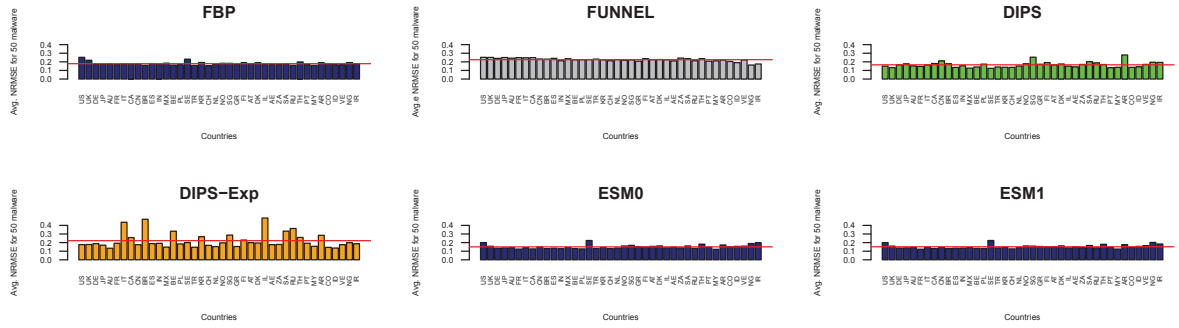
of countries from left to right is the descending order w.r.t. the number of hosts in each country. DIPS models show good correlation coefficients. ESM models show stable and good performance for NRMSE over countries whose number of monitored hosts are very different. The overall prediction performance of DIPS and ESM do not depend on the size of monitored hosts except the cases when the number of monitored hosts are really small relatively. But for the countries having really small number of hosts, such as Nigeria (704) and Iran (634), the models show slightly worse correlation coefficient.

6.3 Summary of experiments

ESM and DIPS have low prediction errors and relatively high correlation coefficient values on average over all 2000



(a) *Correlation*



(b) *NRMSE*

Figure 7: (a) Correlation coefficient and (b) NRMSE by each country. The order from left to right is the descending order by the number of hosts over countries. Each red line is the average over countries.

(country,malware) pairs. Both models have better prediction performance for the country-malware pairs having more infected hosts. For the cases having lower infectious levels, DIPS was not reliable and sometimes yielded very large errors. Interestingly, in the lower infectious levels, DIPS and FBP showed complementary performance. So, the combination of two techniques, ESM showed good and very stable performance for almost all cases except the very small number cases where DIPS was originally the best.

7. RELATED WORK

We considered related work in cyber-security, data mining and epidemiology. Much research has tried to model malware propagation. In July 2001, the Code Red worm infected 359,000 hosts on the Internet in less than 14 hours [22]. Code Red achieved this by probing random IP addresses (using different seeds for its pseudo-random number generator) and infecting all hosts vulnerable to an IIS exploit. Staniford et. al. [26, 25] analyzed the Code Red worm traces and proposed an analytic model for its propagation. They also argue that optimization like hit-list scanning, permutation scanning can allow a worm to saturate 95% of vulnerable hosts on the Internet in less than 2 seconds. Such techniques were subsequently employed by worms released in the wild, such as the Slammer worm [21] (infected 90% of all vulnerable hosts within 10 minutes) and the Witty worm [28]. Gkantsidis et al. study the dissemination of software patches through the Windows Update service and find that approximately 80% of hosts request a patch within a day after it is released; the number of hosts drops by an order of magnitude during the second day, and is further reduced by factor of 2 in day three [8]. Additionally, [27] [15] [17] conducted measurement studies into routing instability in Broder Gateway Protocol (BGP) routers caused by catastrophic events, such as worm outbreaks or power outages. Other recent research also includes using machine learning methods like belief propagation on the file-machine graph [6] to infer files' reputations (say malicious or benign). Papalexakis et. al. [23] propose the SharkFin and GeoSplit models of spatio-temporal propagation of malware based on an analysis of the WINE data. Their system models only the total volume of malware attacks as a whole over time using a simplified model. In contrast, in this paper we model the *magnitude* of malware attacks *per machine* in context of the *machine usage* and using many more sophisticated variables. Remotely related work also comes from the area of information diffusion. Information diffusion is the phenomena in which an action or idea becomes widely adopted due to the influence of others, typically, neighbors in some network [4, 9, 10]. There are a lot of dynamic process on graphs, all of which are related to virus propagation like blogs cascades [1, 14, 11] and information cascades [18].

Finally our DIPS model is inspired by epidemiological models. The canonical textbooks and surveys discussing fundamental epidemiological models like SIS and SIR include [12, 2]. Much work has gone into finding epidemic thresholds (minimum virulence of a virus which results in an epidemic) for a variety of networks [3, 20, 2, 13, 7, 24].

8. CONCLUSIONS

In this paper, we studied the problem of predicting the volume of actual malware infections in a country given past data. This is a challenging problem because we can only detect a fraction of malware infections. There are many potential benefits of accurately predicting malware volume, which can have implications for better anti-virus software,

and targeted patching. Using Symantec's extensive malware database in WINE, we investigated the infection patterns of 50 malware in more than 1.4 million hosts spread over two years in several countries. We present ESM, an ensemble based approach which carefully leverages well-designed domain-specific features (FBP) and a novel non-linear time-series model DIPS which incorporates detection and infections seamlessly. We show that our approach gives robust and stable predictions, even in case of low volume of infections, across all the countries over several metrics (correlation coefficients and NRMSE).

Acknowledgments: This paper is based on work partially supported by the Maryland Procurement Office under Contract No. H98230-14-C-0137, by the NEH under Grant No. HG-229283-15, by ORNL under Task Order 4000143330, by the VT College of Engineering, and a Facebook faculty gift.

We thank Symantec for providing access to the WINE platform. Other researchers may reproduce and verify our results by analyzing the reference data set we recorded in WINE (WINE-2013-001) after signing a research agreement with Symantec.

9. REFERENCES

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. *Web Intelligence*, pages 207–214, 2005.

[2] R. M. Anderson and R. M. May. *Infectious Diseases of Humans*. Oxford University Press, 1991.

[3] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, London, 1975.

[4] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change in informational cascades. *Journal of Political Economy*, 100(5):992–1026, October 1992.

[5] L. Bilge and T. Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *ACM Conference on Computer and Communications Security*, pages 833–844, 2012.

[6] D. H. P. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Polonium : Tera-scale graph mining for malware detection. In *SDM*, Mesa, AZ, April 2011.

[7] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology in spread of epidemics. *IEEE INFOCOM*, 2005.

[8] C. Gkantsidis, T. Karagiannis, and M. Vojnovic. Planet scale software updates. In *SIGCOMM*, pages 423–434, 2006.

[9] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[10] M. Granovetter. Threshold models of collective behavior. *Am. Journal of Sociology*, 83(6):1420–1443, 1978.

[11] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04*, 2004.

[12] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42, 2000.

[13] J. O. Kephart and S. R. White. Measuring and modeling computer virus prevalence. *IEEE Computer Society Symposium on Research in Security and Privacy*, 1993.

[14] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 568–576, New York, NY, USA, 2003. ACM Press.

[15] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. Analysis of BGP Update Burst During Slammer Attack. In *The 5th International Workshop on Distributed Computing*, Dec 2005.

[16] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[17] J. Li, Z. Wu, and E. Purpus. CAM04-5: Toward Understanding the Behavior of BGP During Large-Scale Power Outages. *GLOBECOM '06. IEEE*, pages 1–5, Nov. 2006.

[18] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 6–14, 2012.

[19] Y. Matsubara, Y. Sakurai, W. G. Van-Panhuis, and C. Faloutsos. Funnel: automatic mining of spatially coevolving epidemics. In *KDD*, pages 105–114, 2014.

[20] A. G. McKendrick. Applications of mathematics to medical problems. In *Proceedings of Edin. Math. Society*, volume 44, pages 98–130, 1925.

[21] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *Security & Privacy, IEEE*, 1(4):33–39, 2003.

[22] D. Moore, C. Shannon, and K. C. Claffy. Code-red: a case study on the spread and victims of an internet worm. In *Internet Measurement Workshop*, pages 273–284, 2002.

[23] E. E. Papalexakis, T. Dumitras, D. H. Chau, B. A. Prakash, and C. Faloutsos. Spatio-temporal mining of software adoption & penetration. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2013.

[24] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. In *ICDM*, 2011.

[25] S. Staniford, D. Moore, V. Paxson, and N. Weaver. The top speed of flash worms. In *WORM*, pages 33–42, 2004.

[26] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167, Berkeley, CA, USA, 2002. USENIX Association.

[27] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and Analysis of BGP Behavior under Stress. In *IMW*, 2002.

[28] N. Weaver and D. Ellis. Reflections on Witty: Analyzing the attacker. *;login: The USENIX Magazine*, 29(3):34–37, June 2004.