Attributing Hacks

Ziqi Liu
Xi'an Jiaotong Univ.
Ant Financial Group

Alexander J. Smola
Amazon

Kyle Soska
CMU

Yu-Xiang Wang
CMU

Qinghua Zheng
Xi'an Jiaotong Univ.

Abstract

In this paper, we describe an algorithm for estimating the provenance of hacks on websites. That is, given properties of sites and the temporal occurrence of attacks, we are able to attribute individual attacks to joint causes and vulnerabilities, as well as estimating the evolution of these vulnerabilities over time. Specifically, we use hazard regression with a time-varying additive hazard function parameterized in a generalized linear form. The activation coefficients on each feature are continuous-time functions over time. We formulate the problem of learning these functions as a constrained variational maximum likelihood estimation problem with total variation penalty and show that the optimal solution is a 0th order spline (a piecewise constant function) with a finite number of adaptively chosen knots. This allows the inference problem to be solved efficiently and at scale by solving a finite dimensional optimization problem. Extensive experiments on real data sets show that our method significantly outperforms Cox's proportional hazard model. We also conduct case studies and verify that the fitted functions are indeed recovering vulnerable features.

1 Introduction

Websites get hacked, whenever they are subject to a vulnerability that is known to the attacker, whenever they can be discovered efficiently, and, whenever the attacker has efficient means of hacking at his disposal. This combination of *knowledge*, *opportunity*, and *tools* is quite crucial in shaping the way a group of sites receives unwanted attention by hackers.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

Unfortunately, as an observer, we are not privy to either one of these three properties. In fact, we usually do not even know the exact time t_s a site s was hacked. Instead, all we observe is only that a compromised site will eventually be listed as such on one (or more) blacklists. That is, we know that by the time a site lands on the blacklist it definitely has been hacked. However, there is no guarantee that the blacklists are comprehensive nor is there any assurance that the blacklisting occurs expediently. These blacklists also do not reveal which feature of the website was to blame.

On the other hand, meta-data do exist for each website and they allow us to measure the vulnerability of the websites quantitatively. These include specific string snippets on websites that are indicative of certain versions of software which might have been identified as vulnerabilities or containing bugs that lead to possible security breaches. An interesting method that uses these features to identify websites at risk is recently proposed in Soska & Christin (2014). However, it is unclear how each of these features contributes to the "hazard" of a particular website getting hacked at a given time.

In this paper, we propose a novel hazard regression model to address this problem. Specifically, the model provides a clear description of the probability a site getting hacked conditioned on its time-varying features, therefore allowing prediction tasks such as finding websites at risk, or inferential tasks such as attributing attacks to certain features as well as identifying change points of the activations of certain features to be conducted with statistical rigor.

Related work. The primary strategy for identifying web-based malware has been to detect an active infection based on features such as small iFrames (Mavrommatis & Monrose, 2008). This approach has been pursued by both academia (e.g., Borgolte et al., 2013, Invernizzi & Comparetti, 2012) and industry (e.g., Google, McAfee, Norton). While intuitive, this approach suffers from being overly reactive, and defenders must compete against adversaries in an arms race to detect increas-

ingly convoluted and obscure forms of malice.

Soska & Christin (2014) propose a data driven (linear classification) approach to identify software packages that were being targeted by attackers to predict the security outcome of websites.

Compared with former works, our method is able to predict the time a site will be hacked in a survival analysis framework. Our method naturally handles censoring of observations (i.e. inconsistency of exact hacking time and the time listed on blacklists). Moreover, our model automatically identifies a small number of features as exploits and allows the activation coefficients on each feature to be functions over continuous time. Furthermore, we show the optimal solution of the functions is a 0th order spline adaptively connected by a finite number of adaptively chosen knots. Finally, our hazard regression model is quite generic and much more powerful than the widely-used Cox model, therefore it can be viewed as a novel and alternative way to estimating nonparametric hazard functions at scale, and used as a drop-in replacement in many other applications.

2 Background

Our work is based on two key sets of insights: the specific way how vulnerabilities are discovered, exploited and communicated in the community, and secondly, the mapping of these findings into a specific statistical model.

2.1 Attacks on Websites

We start by describing the typical economics of hackers and websites.

Exploits are first discovered by highly skilled individual (hackers) who will use them for their own purposes for an extended period of time, as long as there is an ample supply of hackable sites that can be discovered efficiently. Once the *opportunity* for such hacks diminishes due to an exhausted supply, the appropriate vulnerabilities are often published since they're now of little value to the discoverer and since publication can convey reputation in the community.

Once this knowledge enters the public domain, the availability of available tools increases with it. It is added to the repertoire of popular rootkits, at the ready disposal of "script kiddies" who will attempt to attack the remaining sites. The increased availability of *tools* often offsets the reduced *opportunity* to yield a secondary wave of infections.

An important aspect in the above scenario is the way how sites are discovered. Quite frequently this is accomplished by web queries for specific strings in sites, indicative of a given vulnerability (e.g. database, CMT, server, scripting language). In other words, string matches are excellent *features* to determine the vulnerability of a site and are therefore quite indicative of the likelihood that such a site will be attacked. Unfortunately, we are not privy to the search strings a potential attacker might issue. However, we can use existing fingerprints to *learn* such sequences, e.g. the tags and attributes in the pages of a site.

In a nutshell, the above leads to the following statistical assumptions on how vulnerability of sites and the infectious behavior occurs. Firstly, sites are only practically vulnerable once a vulnerability is discovered. Second, changes in attack behavior are discrete rather than gradual. In the following we design a statistical estimator capable of adapting to this very profile.

2.2 Hazard Regression

Hazard regression is commonly used in survival analysis of patients suffering from potentially fatal diseases. There, one aims to estimate the chances of survival of a particular patient with covariates (attributes) x, as a function of time, such as to better understand the effects of x. Unfortunately, each patient only has one life, and possibly different attributes x, hence, it is impossible to estimate the fatality rate directly.

Instead, one assumes that the hazard rate $\lambda(x,t)$ governs the instantaneous rate of dying of any x at any given time t:

$$\begin{aligned}\lambda(t) &= \lim_{dt\to 0} \frac{p(t \le T < t+dt | T \ge t)}{dt} \\ &= \lim_{dt\to 0} \frac{p(t \le T < t+dt)}{dt} \cdot \frac{1}{p(T \ge t)}\end{aligned} \tag{1}$$

That is, the density of dying at time t is given by

$$p(t|x) = \lambda(x,t) \underbrace{p\left(\text{survival until } t|x\right)}_{F(t|x)}. \tag{2}$$

This leads to a differential equation for the survival probability with solution

$$F(t|x) = \exp\left(-\int_0^t \lambda(x,\tau)d\tau\right). \tag{3}$$

Here we assumed, without loss of generality, that time starts at 0. Note that a special case of the above is $\lambda(x,t) = \lambda_0$, in which case we have $F(t,x) = e^{-t\lambda_0}$. This is the well-known nuclear decay equation (also an example of survival analysis).

In our case, death amounts to a site being infected and $\lambda(x,t)$ is the rate at which such an infection occurs. An extremely useful fact of hazard regression is that it

is additive. That is, if there are two causes with rates λ and γ respectively, (2) allows us to add the rates. We tacitly assume here that once a site is infected, the attacker will take great care to keep further attackers out, or at least, it will remain blacklisted as long as it is infected in some manner. In terms of (3) we have

$$F(t|x) = \exp\left(-\int_0^t \lambda(x,\tau) + \gamma(x,\tau) d\tau\right) \quad (4)$$
$$\text{and} \quad p(t|x) = [\lambda(x,t) + \gamma(x,t)]\, F(t,x).$$

The reason why this is desirable in our case follows from the fact that we may now model λ as the sum of attacks and can treat them as if they were independent in the way they affect sites.

One challenge in our analysis is the fact that we may not always immediately discover whether a site has been taken over. The probability that this happens in some time interval $[t_1, t_2]$ is given by $F(t_1|x) - F(t_2|x)$, i.e. by the difference between the cumulative distribution functions.

Finally, the absence of evidence (of an infection) should not be mistaken as evidence of absence of such. In other words, all we know is that the site survived until time T. By construction, their probability is thus given by $F(T|x)$. In summary, given intervals $[t_i, T_i]$ of likely infection for site i, at time T we have the following likelihood for the observed data:

$$p(\text{sites}|T) = \prod_{i \in \text{hacked}} [F(t_i, x_i) - F(T_i, x_i)] \prod_{i \notin \text{hacked}} F(T, x_i). \quad (5)$$

Most hazard regression approaches are based on the Cox's propotional hazard model Cox (1972) $\lambda(t|x) = \lambda_0(t)\exp(w^\top x)$, including parametric models, and nonparametric models with baseline hazard rate $\lambda_0(t)$ unspecified Cox (1975). The proportional assumption may not hold because of the time-varying effect of covariates Buchholz et al.. As a result, time-dependent effect models that allow $w(t)$ as functions over time for each feature are proposed. Typically people developed time functions based on fractional polynomials Sauerbrei et al. (2007), or spline functions Kooperberg et al. (1994). Due to the huge parameter space, techniques like reduced rank methods Perperoglou (2013) and structured penalized methods Tibshirani (1997), Verweij & van Houwelingen (1995), Perperoglou (2014) are proposed. However those works either search for global smoothing functions or need to pre-specify knots manually, and typically work on only tens of features.

We do not wish to make strong parametric assumptions, but since x is high-dimensional, estimating $\lambda(x,t)$ completely non-parametrically is intractable. To add to the complexity, inspired by hacking campaigns, $\lambda(x,t)$ is not a smoothly changing function, but can jump suddenly in response to certain events. It may not have a small or even bounded Lipschitz constant. We therefore constrain complexity of the function via total variations that adapts to such heterogeneous smoothness without blowing up the model complexity.

2.3 Trend Filtering

Trend filtering (Kim et al., 2009, Tibshirani, 2014) is a class of nonparametric regression estimators that has precisely the required property. It is minimax optimal for the class of functions $[0,1] \to \mathbb{R}$ whose kth order derivative has bounded total variation. In particular, it has the distinctive feature that when $k = 0, 1$ it produces piecewise constant and piecewise linear estimates (splines of order 0 and 1) and when $k \geq 2$ it gives piecewise smooth estimates. The local adaptivity stems from the sparsity inducing regularizers that chooses a small but unspecified number of knots. When $k = 0$, trend filtering reduces to the fused lasso which solves

$$\operatorname*{argmin}_\beta \mathcal{L}(\beta) + \gamma \sum_{t=1}^{T-1} |\beta_{t+1} - \beta_t|.$$

for a given loss function $\mathcal{L}$. The advantage of this model is that each discrete change in the rate function effectively corresponds to the discovery or the increased (or decreased) exploitation of a vulnerability — after all, the *rate* of infection should not change unless new vulnerabilities are discovered or a patch is released.

3 Attributing Hacks

We will now assemble the aforementioned tools into a joint model for attributing hacks.

3.1 Additive hazard function and variational maximum likelihood

Given the hazard function $\lambda(t, x_i)$ of each website $i \in \{1, ..., n\}$ with feature vector $x_i(t) \in \mathbb{R}^d$ at time t, we have the following survival problem:

$$\max \prod_{i \in \text{hacked}} p(t_i \leq \tau_i < T_i) \prod_{i \notin \text{hacked}} p(\tau_i > T)$$

where τ_i is the *unobserved* random variable indicating the exact time that website x_i is being hacked. All we know for websites on the blacklist[1] is the time already

[1] A blacklist in this context is a list of websites maintained by a third party which are confirmed to be either malicious, compromised, or otherwise adversarial according to the expertise of the curator. Entries in a blacklist always contain the website in question, but are also furnished timestamps of the security event and information regarding the nature of the malice.

been hacked T_i and the last time it was alive t_i. This is what we call an "interval-censored" observation. Time T denotes the end of the observation interval, e.g., now. Websites that were still alive at T are considered "right censored" because all we know is that their hack time will be beyond of T. Under the survival analysis framework, we have

$$\begin{aligned} p(\tau_i > T) =& e^{-\int_0^T \lambda(t,x_i(t))dt} \\ p(t_i \le \tau_i < T_i) =& p(\tau_i \ge t_i) - p(\tau_i \ge T_i) \\ =& e^{-\int_0^{t_i} \lambda(t,x_i(t))dt} - e^{-\int_0^{T_i} \lambda(t,x_i(t))dt} \end{aligned} \tag{6}$$

It remains to specify the hazard function. In our setting, x is a high-dimensional feature vector, so we need to impose further structures on the hazard function $\lambda(x,t)$ to make it tractable. We thus make an additive assumption and expand the hazard function into an inner product

$$\lambda(x,t) = \langle x(t), w(t) \rangle = w_0(t) + \sum_{i=1}^{d} x_i(t) w_i(t).$$

This is still an extremely rich class of functions as $x_i(t)$ can be different over time and $w_i(t)$ is allowed to be any univariate nonnegative function (denoted by $\mathcal{F}$). Leaving it completely unconstrained will inevitably overfit any finite data set. In order to allow for sharp changes of $w_i(t)$, we choose to constrain the complexity of the function class via a total variation (TV) penalty. Then we can learn the model by solving the variational penalized maximum likelihood problem below:

$$\begin{aligned} \min_{(w_0,w_1,\ldots,w_d)\in\mathcal{F}^d} & \sum_{i=1}^{n} \ell(\{x_i, z_i, \psi_i\}; w) + \gamma \sum_{j=0}^{d} \mathrm{TV}(w_j) \\ \text{s.t.} \quad & w_j(t+\delta) - w_j(t) \ge 0 \\ & \text{for any } j \in [d], t \in \mathbb{R}, \delta \in \mathbb{R}_+ \end{aligned} \tag{7}$$

where z_i is the indicator of censoring type for observation x_i, i.e. interval-censored or right-censored; $\psi_i := \{t_i, T_i, T\}$ is the associated censoring time; $w_j(t)$ is the evaluation of function w_j at time t. The monotonic constraints are optional. We call the model class "non-monotone" when we drop the constraints from (7). Note that our method is a much richer representation comparing to Cox's proportional hazard model (Cox, 1972). Our method handles time-varying coefficients and feature vectors while Cox model is static. Also, the semiparametric nature of Cox model by construction leaves out the baseline hazard $\lambda_0(t)$ such that it becomes non-trivial to produce a proper survival distribution. For example, a common trick is to parametrize the baseline hazard rate $\lambda_0(t)$ by a either a constant or a log-Weibull density. Our formulation does not require a parametric assumption and produces a nonparametric estimate of it to account for all the effects that are not explained by the given feature. There, only issue is that (7) is an infinite dimensional function optimization problem and could be very hard to solve.

3.2 Variational characterization

The following theorem provides a finite set of simple basis functions that can always represent at least one of the solution to (7).

Theorem 1 (Representer Theorem)**.** *Assume no observations are uncensored, feature $x_i(t)$ for each site is piecewise constant over time with finite number of change points. Let $s_\tau(t) = 1(t \ge \tau)$ be the step function at τ. There exists an optimal solution $(w_1, ..., w_d)$ of the above problem such that for each $j = 1, .., d$,*

$$w_j(t) = \sum_{\tau \in \mathcal{T}} s_\tau(t) c_\tau^{(j)}$$

for some set $\mathcal{T}$ that collects all censoring boundaries and places where feature $x_j(t)$ changes, and coefficient vector $c^{(j)} \in \mathbb{R}^{|\mathcal{T}|}$.

The proof, given in the supplementary, uses a variational characterization due to De Boor (1978), Mammen et al. (1997) and a trick that reparameterizes our problem using the cumulative function $W_i(t) = \int_0^t w_i(t)dt$. Extra care was taken to handle the nonnegativity and monotone constraints. We remark that the above result also applies to the case when $\gamma = 0$ (unpenalized version), and/or the case when there is no monotone constraints.

The direct consequence of Theorem 1 is that we can now represent piecewise constant functions by vectors in $\mathbb{R}^{|\mathcal{T}|}$ and solve (7) by solving a tractable finite dimensional fused lasso problem (with an optional isotonic constraints) of the form:

$$\begin{aligned} \min_{w_0,w_1,\ldots,w_d\in\mathbb{R}^{|\mathcal{T}|}} & \sum_{i=1}^{n} \ell(\{x_i, z_i, \psi_i\}; w) + \gamma \sum_{j=0}^{d} \|Dw_j\|_1 \\ \text{s.t.} \quad & w_j(\ell+1) - w_j(\ell) \ge 0 \\ & \text{for any } j = 1, ..., d, \ell = 1, ..., |\mathcal{T}| - 1. \end{aligned} \tag{8}$$

where we abuse the notation to denote w_j as evaluations of function w_j at sorted time points in $\mathcal{T}$; and $D \in \mathbb{R}^{(|\mathcal{T}|-1)\times|\mathcal{T}|}$ is the discrete difference operator.

Although the above result does not cover the cases when we penalize the log penalty $\sum_{\ell=1}^{|\mathcal{T}|-1} \log\left(|w_j(\ell+1) - w_j(\ell)| + \epsilon\right)$ instead of the ℓ_1 norm in (8), we can still restrict our attention to the class of piecewise constant functions, which is a sensible reformulation anyway. The reason we consider such a nonconvex penalty is, say two small changes we would rather prefer one large change. Using an ℓ_1 norm this is sometimes difficult to

accomplish, because when minimizing $|a-b|+|b-c|$, any value of $b \in [a,c]$ will be a minimizer (for fixed boundaries $a < c$). A nonconvex penalty, on the other hand, allows for such changes.

Remark 1 (Higher order Trend Filtering)**.** *For kth order trend filtering with $k \geq 1$, we do not get the same variational characterization. Although we can still show that there is a spline W_j^* that is optimal, there is no guarantee that the knots of W_j^* are necessarily a subset of $\mathcal{T}$. Fortunately, by Proposition 7 of Mammen et al. (1997), restricting our attention to the class of splines with knots in $\mathcal{T}$ will yield a spline that is very close to the W_j^* at every $\tau \in \mathcal{T}$ and it has total variation of its kth derivative on the same order as* $\mathrm{TV}(W_j^*)$*. In addition, a spline is uniformly approximated by the class of functions that can be represented by the falling factorial basis (Tibshirani, 2014, Wang et al., 2014), therefore, the function from kth order trend filtering defined on $\mathcal{T}$ will be a close approximation to the optimal solution of the original variational problem.*

Remark 2 (A sparse and memory-efficient update scheme)**.** *The theorem suggests a memory-efficient scheme for optimization, as one can only keep track of the coefficients of the step functions rather than representing the dense vectors $w_0, ..., w_d$. Moreover, each stochastic gradient update will be sparse since each user has only a handful of changes in his feature vectors over time and at most two censoring brackets.*

Remark 3 (On statistical error rates)**.** *The model we put together is in fact an* additive *trend filtering model. A recent manuscript (Sadhanala & Tibshirani, 2017) showed that the minimax rate of such models is $dn^{-\frac{2k+2}{2k+3}}$ under a number of assumptions. It does not directly apply to our problem, due to our non-convex loss functions and additional non-negativity and isotonic constraints.*

3.3 Algorithms

Due to the interval-censoring in problem (8), the loss functions are not convex, and the penalty is either convex (ℓ_1) or nonconvex ($\log(|\cdot|+\epsilon)$) but non-smooth. In addition, there are non-negativity and possibly isotonic constraints. In our experiments, we find that proximal gradient approach with a stochastic variance reduced gradient approximation (Johnson & Zhang, 2013) works well for our purpose and it allows us to scale up the method to work with at least hundreds of thousands of data points and features.

Note that the probability of each interval-censored data $-\log(p(t_i \leq \tau_i < T_i))$ in (6) can be decomposed as $\int_0^{t_i} \lambda(t,x_i)dt - \log\Big(1-\exp\big(-\int_{t_i}^{T_i}\lambda(t,x_i)dt\big)\Big)$. As a result we only need to calculate the integral between t_i and T_i for interval-censored data while evaluating the gradients:

$$\nabla_{g_i(w)} = \begin{cases} \frac{1}{1-\exp\left(\int_{t_i}^{T_i}\lambda(t,x_i)dt\right)} D\tau^\top, & \text{if } t_i \leq t < T_i. \\ D\tau^\top, & \text{otherwise.} \end{cases}$$

where $g_i(\cdot)$ is the negative log-likelihood function on x_i; $\tau \in \mathbb{R}^{d,\mathcal{T}}$ denotes sorted times in $\mathcal{T}$ for each feature $j \in \{1, ..., d\}$. We have the following updates:

$$\begin{aligned} z^{t+1} &= z^t + \nabla^2_{g_i(w^t)}, \\ w^{(1)} &= w^t - \frac{\alpha}{\beta+\sqrt{z^{t+1}}}(\nabla_{g_i(w^t)} - \nabla_{g_i(\tilde{w})} + \tilde{\mu}), \\ w^{(2)} &= \min_{w \text{ is isotonic}} \|w^{(1)} - w\|_2 + \|Dw^{(1)}\|_1, \\ w^{t+1} &= \min_{w \text{ is nonnegative}} \|w^{(2)} - w\|_2 + \|w\|_1. \end{aligned}$$

Here, $\tilde{w}$ means a snapshot of w after every m iterations; $\tilde{\mu}$ means the average of the $\nabla_{g_i(\tilde{w})}$ over all x_i. The third update can be solved in linear time by dynamic programming (Johnson, 2013). Lastly, Reddi et al. (2016) proved fast convergence rate of proximal SVRG to a stationary point for nonconvex loss functions, which guarantees that only $O(1/\epsilon)$ proximal operators calls and $O(n + n^{2/3}/\epsilon)$ incremental gradient computation are needed to get to ϵ accuracy.

4 Experiments

In this section, we show the accuracy of the inference of the learned latent hazard function evaluated on synthetic data with ground truth and by conducting case studies on real data via domain expert's knowledge. We also evaluate the out-of-sample predictive power measured by log-likelihood, that significantly outperforms Cox's proportional methods.

4.1 Simulation study

We simulate two kinds of attacks. First one is attacks with strict monotonic hazard rate. It corresponds to our statistical model with a monotone constraint on the hazard rate. This is easy to understand because once a vulnerability is known it will become easier and easier for hackers to attack as more tools are available. The second one is simulated attacks without monotonic hazard rate. This leads to our "non-monotone" model. It's a practical assumption because in reality, the attack campaigns could be complex. We will talk about both the cons and pros of these two schemes in the analysis of real-world data.

In both cases, we simulate the data as follows. We generate 40 features among of which 4 features are vulnerable exploits that are potentially under attack. To simulate the true attacks, we assume there could be

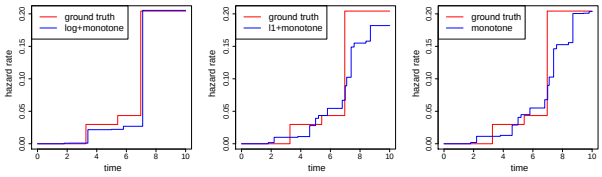

Figure 1: Estimated hazard rate on one exploit: log+monotone(**left**), l1+monotone (**middle**) and monotone (**right**).

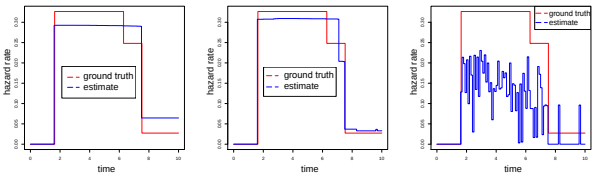

Figure 2: Estimated hazard rate on one exploit: log+l1(**left**), l1 (**middle**) and non-monotone (**right**).

several attack campaigns for each exploit. We randomly pick change points over time (cast as real numbers in $[0, 10.0]$) each of which corresponds to one attack campaign. The hazard rate for each campaign are randomly sampled too. Given the ground true hazard rate we sample the hacked times for each of 1000 data points. Independently, we assume another uniformly sampled checking points served as censoring times. We finally obtain our experimental interval-censored times by finding the nearest censoring times around each hacked time.

The results for monotonic hazard rates are reported in Figure 1 and 3. We denote "ℓ_1" as ℓ_1 penalized Total Variation, and "log" as log penalized Total Variation. The convergence in Figure 3 shows that compared with "ℓ_1" and monotone, "log" penalty works a bit better. The reason can be seen from Figure 1 (1 out of 4 exploit) where the "log" penalty produces much sharper hazard curves and approximate the ground truth quite well.

Figures 2 and 4 illustrate the results on data without monotonic hazard rates constraint. Both the "ℓ_1" and "log" penalties work well. It is well expected that the non-monotone model without any regularization would overfit the data badly. The minor difference between "ℓ_1" and "$\ell_1 + \log$" is that "$\ell_1 + \log$" produces sharper curves but tends to ignore weak signals (e.g. the second knot) when the signal-to-noise ratio is relatively small, i.e. prefer significant signals.

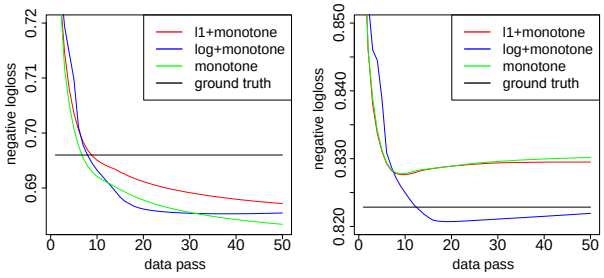

Figure 3: Convergence on training data (**left**) and test data (**right**) respectively. (monotone hazard rate)

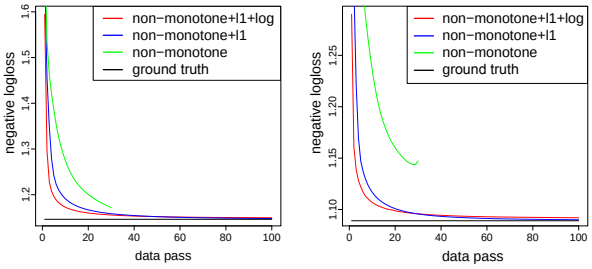

Figure 4: Convergence on training data (**left**) and test data (**right**) respectively. (non-monotone hazard rate)

4.2 Real-World Data

The data used for evaluation was sourced from the work of Soska & Christin (2014) and was compromised as a collection of interval censored sites from blacklists and right censored sites randomly sampled from .com domains [2]. As a consequence of the time-varying distribution of software deployed on the web, all the samples were drawn from The Wayback Machine[3] when archives were available at appropriate dates.

One of the blacklists that was sampled was PhishTank, a blacklist of predominately phishing[4] websites for which 11,724,276 unique links from 91,155 unique sites were observed between February 23, 2013 and December 31, 2013. The Wayback Machine contained usable archives for *34,922* (38.3%) domains. The other blacklist that was used contains websites that perform search redirection attacks Leontiadis et al. (2014) and was sampled from October 20, 2011 to September 16, 2013. In total the sample contained 738,479 unique links, from 16,173 unique domains. Amazingly, the

[2] A .com zone file is the list of all registered .com domains at the time.

[3] The Wayback Machine is a service that archives parts of the web.

[4] A phishing website is a website that impersonates another site such as a bank, typically to trick users and steal credentials.

Wayback Machine contained archives in the acceptable range for *14,425* (89%) of these sites.

These two blacklists are particularly well suited for providing labeled samples of attacked websites as manual inspection has shown, an overwhelmingly large proportion of these sites were compromised by a hacker.

Lastly, the .com zone file from January 14th, 2014 was randomly sampled, ignoring cases where an image of the site was not available in The Wayback Machine. In total *336,671* archives distributed uniformly between February 20th, 2010 and September 31st, 2013 were collected. These samples were checked against our blacklists as well as Google Safe Browsing to ensure that as few compromised sites remained in the sample as possible.

We automatically extracted raw tags and attributes from webpages, that served as features (a total of *159,000* features). These tags and attributes could be like
, or <meta> WordPress 2.9.2</meta>. They are useful for indicating the presence of code that is vulnerable or may be the target of adversaries.

4.3 Real-World Numeric Results

To estimate the actual hazard rate we first estimate the approximate distributions over hacked websites and still not hacked websites during that period. There are 120 million websites registered in .com zone file at end. We reweigh the non-hacked websites by 200 times. To report the results, we randomly select 80% for training and validation, and the rest as test data.

The baseline method is the classic Cox Proportional model (Cox, 1972) which has been extensively used for hazard regression and survival analysis ever since its invention, is still considered a "gold standard" in epidemiology, clinical trials and biomedical study today (see e.g., Woodward, 2013). Cox model is parametrized based on the features just like we do, but is not time-varying. As has been discussed in section 3.1, to estimate the survival probabilities we specify a uniform distribution for the baseline hazard function.

An experimental comparison between our models and Cox on the aforementioned dataset are shown in Figure 5. Comparing to our models, the Cox model underfits the data quite a bit. Our "monotone" model that allows only non-decreasing hazard rate underfits the data a little but still significantly outperforms Cox. Moreover due to much smaller parameter space need to search, we find that it converges faster than "ℓ_1+non-monotone" model. Our "log+monotone" model performs nearly the same convergence (overlapped). Again it is well expected that "non-monotone" model without any constraint overfits the data severely. "ℓ_1+non-monotone" model which is well-regularized performs the best.

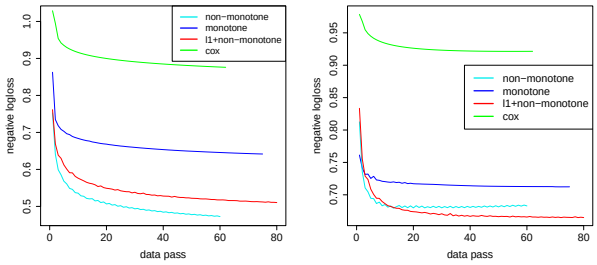

Figure 5: Convergence on training data (**left**) and test data (**right**) respectively.

Methods	Empirical model size
non-monotone	$2 \cdot 10^6$
monotone	$4.04 \cdot 10^5$
ℓ_1+nonmonotone	$5.16 \cdot 10^5$
Cox	$1.59 \cdot 10^5$

Table 1: Empirical model size (active breakpoints for our methods, number of parameters for Cox) estimated by different statistic models.

Due to the sparsity of our models, table 1 shows that with only around 3 times parameter storage our models can give significantly better estimates compared with the Cox model. Most importantly, identifying the changes of each feature's susceptibility over time can help people understand the latent hacking campaigns and leverage these insights to take appropriate action. We will discuss more in section 4.3.1.

Finally it is imperative that the model does not assign non-zero hazard rate to features that are uncorrelated with the security outcome of a website. The hazard curve for 200 random features believed to be uncorrelated with security (such as tags for custom font colors, styles, and links to unique images) was manually studied, 182 (9% false positive rate) of which generated a hazard value of 0 for the enture duration of the experiment. Of the 18 features that were assigned a non-zero hazard curve, all of them reported a value of less than 0.04 which can be ignored.

4.3.1 Real-World Case Study

In this section, we manually inspect the model's ability to automatically discover known security events. To this end, the model was trained on the aforementioned dataset and $\lambda_i(t)$ was measured for features i that corresponded directly to websites that were known to be the victim of attacks.

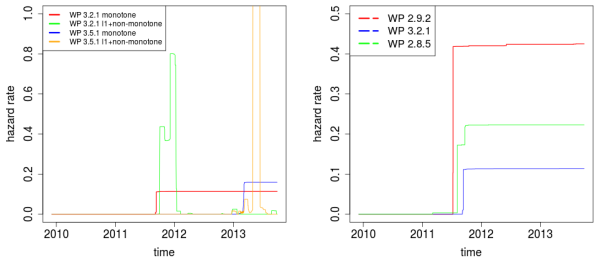

Figure 6: $\lambda_t(i)$ of a feature known to correspond directly to instances of Wordpress 3.2.1 and Wordpress 3.5.1.(**left**); $\lambda_t(i)$ of features known to correspond directly to different versions of the Wordpress content management system that were attacked in the summer of 2011.(**right**)

Figure 6 (**left**) demonstrates some of the differences between the monotone and non-monotone models by following the hazard assigned to features that correspond to Wordpress 3.5.1. In early 2013, our dataset recorded a few malicious instances of Wordpress 3.5.1 sites (among some benign ones). These initial samples appeared to be part of a small scale test or proof of concept by the adversary to demonstrate their ability to exploit the platform. Both models detect these security events and respond by assigning a non-zero hazard.

Following the small scale test was a lack of activity for a few weeks, during which the non-monotone model relaxes its hazard rate back down to zero, just before an attack campaign on a much larger scale is launched. This example illustrates the importance of not letting a guard down in the context of security. Once a vulnerability for a software package is known, that package is always at risk, even if it is not actively being exploited.

Despite not taking the most prudent approach to security, the non-monotone model captures the notion that adversaries tend to work in batches or attack campaigns. Previous work Soska & Christin (2014) has shown that it is economically efficient for adversaries to compromise similar sites in large batches, and after a few attack campaigns, most vulnerable websites tend to be ignored. This phenomena is shown in Figure 6 where Wordpress 3.2.1 was attacked in late 2011 and then subsequently ignored with the exception of a few small attacks that were likely the work of amateurs or password guessing attacks which are orthogonal to the underlying software and any observable content features. The monotone model in this case is very prudent while the non-monotone model captures the notion that the software is not being targeted.

It can be observed from Figure 6 (**right**) that a number of distinct Wordpress distributions experienced a change-point in the summer of 2011 (between July 8th 2011 and August 11th 2011). This phenomena was present in several of the most popular versions of Wordpress in the dataset including versions 2.8.5, 2.9.2 and 3.2.1.

This type of correlation between the hazard of features corresponding to different versions of a software package is expected. This correlation often occurs when adversaries exploit vulnerabilities which are present in multiple versions of a package, or plugins and third party add-ons that share compatibility across the different packages.

Manual investigation revealed that a number of impactful CVEs[5] such as remote file inclusion and privilege escalation were found for these versions of Wordpress as well as a particular plugin around the time of July 2011. While it is impossible to attribute with certainty any particular vulnerability, the observed behavior is consistent with vulnerabilities that impact large number of consecutive iterations of software.

5 Conclusion

In this paper, we propose a novel survival analysis-based approach to model the latent process of websites getting hacked over time. The proposed model attempts to solve a variational total variation penalized optimization problem, and we show that the optimal function can be linearly represented by a set of step functions with the jump points known to as ahead of time. This allows us to solve the problem by either Lasso or fused lasso efficiently using proximal stochastic variance reduced gradient algorithm. The results suggest that the model significantly outperforms the classic Cox model and is highly interpretable. Through a careful case study, we found that at least some of the active features and jump points we discovered by fitting the model to data are indeed important components of known vulnerability, and major jump points often clearly mark out the life cycles of these exploits. In the future, the same model (and variants) can be used in many other settings to study consumer spending behaviors, marriage, animal habitats and so on.

Acknowledgements

ZL was sponsored by "The Fundamental Theory and Applications of Big Data with Knowledge Engineering" with grant number 2016YFB1000903; Creative

[5]CVE stands for Common Vulnerabilities and Exposures, which is a list of publicly disclosed vulnerabilities and security risks to software.

Program of Ministry of Education (IRT13035); NSF of China (91118005, 91218301, 61428206); 863 Program (2012AA011003).

YW was supported by NSF Award BCS-0941518 to CMU Statistics, a grant by Singapore NRF under its International Research Centre @ Singapore Funding Initiative, and a Baidu Scholarship.

References

Borgolte, Kevin, Kruegel, Christopher, and Vigna, Giovanni. Delta: automatic identification of unknown web-based infection campaigns. In *ACM SIGSAC conference on Computer & communications security*, pp. 109–120. ACM, 2013.

Buchholz, Anika, Sauerbrei, Willi, and Royston, Patrick. On approaches for modelling time-varying effects in survival analysis.

Cox, David R. Regression models and life tables (with discussion). *Journal of the Royal Statistical Society.*, 34(2):187–220, 1972.

Cox, David R. Partial likelihood. *Biometrika*, 62(2): 269–276, 1975.

De Boor, Carl. *A practical guide to splines*. Springer-Verlag New York, 1978.

Google. "google safe browsing api". URL `https://code.google.com/apis/safebrowsing/`.

Invernizzi, Luca and Comparetti, Paolo Milani. Evilseed: A guided approach to finding malicious web pages. In *IEEE Symposium on Security and Privacy*, pp. 428–442. IEEE, 2012.

Johnson, Nicholas A. A dynamic programming algorithm for the fused lasso and l_0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2): 246–260, 2013.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS-13)*, 2013.

Kim, Seung-jean, Koh, Kwangmoo, Boyd, Stephen, and Gorinevsky, Dimitry. L1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

Kooperberg, Charles, Stone, Charles, and Truong, Young. Hazard Regression, 1994.

Leontiadis, Nektarios, Moore, Tyler, and Christin, Nicolas. A Nearly Four-Year Longitudinal Study of Search-Engine Poisoning. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014.

Mammen, Enno, van de Geer, Sara, et al. Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413, 1997.

Mavrommatis, Niels Provos Panayiotis and Monrose, Moheeb Abu Rajab Fabian. All your iframes point to us. In *USENIX Security Symposium*, pp. 1–16, 2008.

McAfee. "site advisor". URL `http://www.siteadvisor.com/`.

Norton. "norton safe web". URL `http://safeweb.norton.com`.

Perperoglou, Aris. Reduced rank hazard regression with fixed and time-varying effects of the covariates. *Biometrical Journal*, 55(1):38–51, 2013.

Perperoglou, Aris. Cox models with dynamic ridge penalties on time-varying effects of the covariates. *Statistics in medicine*, 33(1):170–180, 2014.

Reddi, Sashank J, Sra, Suvrit, Poczos, Barnabas, and Smola, Alex. Fast stochastic methods for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems (NIPS-16)*, 2016.

Sadhanala, Veeranjaneyulu and Tibshirani, Ryan J. Additive models with trend filtering. *arXiv preprint arXiv:1702.05037*, 2017.

Sauerbrei, Willi, Royston, Patrick, and Look, Maxime. A new proposal for multivariable modelling of time-varying effects in survival data based on fractional polynomial time-transformation. *Biometrical Journal*, 49(3):453–473, 2007.

Soska, Kyle and Christin, Nicolas. Automatically detecting vulnerable websites before they turn malicious. *USENIX Security Symposium*, 2014.

Tibshirani, Robert. the Lasso Method for Variable Selection in the Cox Model. *Statistics in Medicine*, 16(4):385–395, 1997.

Tibshirani, Ryan J. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Verweij, Pierre JM and van Houwelingen, Hans C. Time-dependent effects of fixed covariates in cox regression. *Biometrics*, pp. 1550–1556, 1995.

Wang, Yu-Xiang, Smola, Alex, and Tibshirani, Ryan. The falling factorial basis and its statistical applications. In *International Conference on Machine Learning (ICML-14)*, 2014.

Woodward, Mark. *Epidemiology: study design and data analysis*. CRC press, 2013.