Predicting Global Trend of Cybersecurity on Continental Honeynets Using Vector Autoregression

Xing Ling and Yeonwoo Rho
Mathematical Sciences
Michigan Technological University
Houghton, MI
{xling, yrho}@mtu.edu

Chee-Wooi Ten
Electrical and Computer Engineering
Michigan Technological University
Houghton, MI
ten@mtu.edu

***Abstract*—The deployment of honeynets from around world is intended to lure attackers into their networks and hence their footprint can be extracted and studied. This global trend can be correlated based on publicly available statistics. This paper proposes a statistical analysis to identify a geospatial and temporal patterns in the cyberattacks and use this knowledge to predict future attack trend. Using a publicly available honeypot data, this work aims to (i) incroporate long range dependence in the analysis of the number of cyber-attacks, which may be a result of spread of malware agents, (ii) propose a measure on how to determine whether or not to consider dependence structure between different honeypot hosts, and (iii) establish a modeling tool that would be intuitive in a honeynet, where honeypot hosts are closely connected and related. The proposed vector autoregression approach reveals the dependencies among honeypots.**

***Index Terms*—Anomaly analytic, honeynet, machine learning, vector autoregression.**

I. Introduction

Cyberspace is a globally interconnected massive network that enables communication from one part of the world to another. The number of cybercrimes worldwide is increasing rapidly at an alarming pace. Protecting networks against intrusion and malware is a necessity in business and critical cyber infrastructure. According to Online Trust Alliance (OTA), the total volume of reported cyber incidents has doubled from 82,000 in 2016 to 159,700 in 2017, and they also speculated that the unreported volume was over 190,300 in 2017 [1]. Cybercrimes are predicted to cost $6 trillion annually by 2021 [2]. According to a survey of Wall Street Journal, the majority of top U.S. cities are covered by cyber insurance [3]. In an endeavor to pursue secure cyber systems as well as precise simulate cyber insurance models, a number of innovative ideas are proposed. Honeypots, or honeynet, are one of such endeavors. A honeypot is a computing resource that offers no legitimate service but appears to be legal. Without any production value of honeypot, any attempts to contact it are considered attacks [4]. Honeynet consists of a collection of honeypots that may appear to be real computer systems in a network. With the implementation of a well-designed honeynet system, a sample of real attack data on honeynet can be obtained. These datasets can be useful in (i) preventing the risk of being compromised by updating the firewall rules and (ii) forecasting the trend of future attacks by running statistical analysis on observed attacks. The former is already implemented in practice, but the latter aspect has not been extensively researched yet. In this paper, a procedure is proposed on how to perform a statistical analysis of honeynet-captured attack data and forecast future attacks. This would allow one to preallocate resources accordingly for improvements.

There are only a handful of existing literature on the statistical models of honeypot-captured attack data. Honeypot-captured cyber attacks tend to exhibit long range memory (LRM) because LRM-aware models tend to perform better than LRM-less models in terms of forecasting errors as well as allowing sufficient early-warning time for defenders [5]. A few others in the literature focused on modeling dependencies among honeypots and modeling the time dependencies without considering the LRM. A macroscopic analysis on the intrinsic spatio-temporal trend of cyber attacks is carried out [6], where honeypots are treated as if they were on a one-dimensional space, setting their IP addresses as their locations. In this way, the distance between different honeypots is the difference in their IP addresses. It is also found that the majority of attacks are conducted by a few major attackers, and those attacks tend to be spatially concentrated. Therefore, the attacking process can be predicted with highly accuracy considering spatial locations. With a similar dataset as used in [5], a combination of temporal model and a copula is proposed to accommodate the time dependencies and the interconnection between honeypot hosts: a generalized autoregressive conditional heteroskedasticity (GARCH) model for the time dependencies and a truncated vine copula for the dependencies among honeypots [7]. Similarly, a 4-dimensional multivariate time series collected by a network telescope, which is a tool similar with honeypot but with no interactions with attackers, is analysed using GARCH-copula approach [8]. The 4 marginal series are also fitted by an AR+GARCH model and then use vine copulas to fit the standardized residuals in order to capture the rationally symmetric dependence among these 4 series in the model.

The aforementioned literature addresses the time dependency and the interconnection of honeypots separately. There

This work was supported by US National Science Foundation (NSF) under the award 1739422, entitled "CPS: Medium: Collaborative Research: An Actuarial Framework of Cyber Risk Management for Power Grids."

978-1-5386-8218-0/19/$31.00©2019 IEEE

has not been a research that addresses dependence structure between honeypot hosts and the time dependencies in one model. In addition, the LRM has not been addressed with consideration of dependency structure among honeypots. This paper seeks to address these two gaps in literature. In this paper, a vector autoregression (VAR) model is proposed, which (i) allows one to handle the two types of dependencies (in time and among honeypot hosts) at the same time, and (ii) allows LRM be naturally integrated. However, with the increase of the number of honeypot hosts, the VAR model becomes impossible to estimate because of the large number of parameters. BigVAR is well-suited for prediction of high-dimensional time series. The remainder of this paper is organized into three parts: section II presents the method applied. Section III describes the data and the results of case studies. Section IV summarizes and concludes the proposed method.

II. Proposed Methodology

This section addresses the statistical fundamentals of autocorrelation for Long Range Memory (LRM), vector autoregression (VAR) model, and Least Absolute Shrinkage and Selection Operator (LASSO) method for VAR. BigVAR, the combination of VAR and LASSO, is introduced as how to select its tuning parameter λ.

A. Long Range Memory

Long Range Memory (LRM), also known as Long Range Dependence (LRD), can be modeled using the fractional integration. With LRM, the autocorrelation decays more slowly as lag increases than exponential while the stationary condition still holds. The fractional integration provides a tool to concisely model the LRM. This part provides an explanation on how to add and remove fractional integration. This approach is slightly different from the usual approach using gamma functions as in [9] and may be more intuitive for readers who are not familiar with gamma functions. Consider a binomial series expansion,

$$(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = \sum_{k=0}^{\infty} \frac{\prod_{a=0}^{k-1}(d-a)}{k!}(-B)^k,$$

where B is a backshift operator and d takes values between -0.5 and 0.5. Then a fractional-difference from a time series X_t is evaluated as follows:

$$\begin{aligned} Y_t &= X_t(1-B)^d \\ &= X_t - dX_{t-1} + \frac{d(d-1)}{2!}X_{t-2} + \cdots \\ &= (X_t, X_{t-1}, X_{t-2}, \cdots) \begin{pmatrix} 1 \\ -d \\ \frac{d(d-1)}{2!} \\ \vdots \end{pmatrix}. \end{aligned} \tag{1}$$

Similarly,

$$Y_{t-1} = (X_{t-1}, X_{t-2}, X_{t-3}, \cdots) \begin{pmatrix} 0 \\ 1 \\ -d \\ \frac{d(d-1)}{2!} \\ \vdots \end{pmatrix}.$$

Combining all Y_t's into a row vector,

$$Y^\intercal = X^\intercal A, \tag{2}$$

where $X^\intercal = \begin{pmatrix} X_t & X_{t-1} & \cdots \end{pmatrix}$, $Y^\intercal = \begin{pmatrix} Y_t & Y_{t-1} & \cdots \end{pmatrix}$ and $A = \begin{pmatrix} 1 & 0 & \cdots \\ -d & 1 & \cdots \\ \frac{d(d-1)}{2!} & -d & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$. Denote $A = \begin{pmatrix} a_1 & 0 & \cdots \\ a_2 & a_1 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$, which is a half-infinite triangular Toeplitz matrix. By the property of triangle Toeplitz matrix,

$$A^{-1} = \begin{pmatrix} b_1 & 0 & \cdots \\ b_2 & b_1 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$

where $b_1 = 1/a_1$, and $b_k = -1/a_1 \sum_{j=1}^{k-1} a_{k+1-j} b_j$ for $k > 1$. Because $(b_k)_{k=1}^{\infty}$ decay to zero [10], only $b_1, \cdots, b_j$ need computation for some tolerance level j and the followings $(b_k)_{k=j+1}^{\infty}$ are all zero's. Using the property of decay, the LRM can be removed by $X^\intercal = Y^\intercal A^{-1}$ and added back by (2).

B. VAR and LASSO

BigVAR model is applied with a popular penalty function as in [11], [12]. A vector autoregression (VAR) model with order p is presented as

$$\mathbf{y}_t = \mathbf{v} + \sum_{l=1}^{p} \Phi^{(l)} \mathbf{y}_{t-l} + \mathbf{u}_t$$

where $(\mathbf{y}_t)_{t=1}^{T}$ is a k-dimensional multivariate time series and $\mathbf{v}$ is a k-dimensional non-random vector representing the intercept of the VAR model. Each $\Phi^{(l)}$ represents a $k \times k$ endogenous coefficient matrix at lag $l = 1, \cdots, p$; that is, the (i,j)th element $\Phi^{(l)}(i,j)$ represents the coefficient between $Y_{i,t}$ and $Y_{j,t-l}$. The noise vector $\mathbf{u}_t$ is assumed to be independent identically distributed with mean 0 and some covariance matrix Σ_ϵ.

If k, the dimension of $\mathbf{y}_t$, is not so large, the parameters of a VAR model can be estimated using the multivariate least squares method solving the following optimization problem:

$$\text{minimize}_{\ \mathbf{v},\Phi} \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{v} - \sum_{l=1}^{p} \Phi^{(l)} \mathbf{y}_{t-l}\|^2.$$

The objective function involves $k + pk^2$ parameters. In high-dimensional cases with large k, this optimization becomes unstable if not infeasible. To achieve a sparse estimation, a

LASSO-type penalized regression approach is applied here. The new objective function is

$$\text{minimize}_{\mathbf{v},\Phi} \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{v} - \sum_{l=1}^{p} \Phi^{(l)} \mathbf{y}_{t-l}\|^2 + \lambda \|\Phi\|_1,$$

where $\Phi = (\Phi^{(1)}, \cdots, \Phi^{(p)})$ is the coefficient matrix, $\lambda \geq 0$ is a penalty parameter and the penalty form $\|\cdot\|_1$ is the L_1 norm, as proposed in [13]. This L_1 penalty is chosen because of the computational tractability [14].

C. Model Selection

This section explains the details of how to choose the regularization parameter λ by cross-validation. First, the data is divided into training set and testing set. The followings are dealt in the training set. To begin the search, a smallest value of λ is initiated to make all components of $\|\Phi\|$ to be zero. This is used as the ending value of the grid, or the depth of the grid. Then a grid of potential penalty parameter $\lambda_1, \cdots, \lambda_n$ is selected, letting the distance between neighboring gridpoints increase geometrically. The finer the grid is, the higher the computation costs are. After achieving enough computation accuracy, increased depth in the grid does not improve the model accuracy. As suggested by [4], ten grid points ranging from 10 to 50 depth are enough to achieve the highest accuracy.

The optimal penalty parameter $\widehat{\lambda}$ is chosen by the minimizer of 1-step ahead mean square forecasting error (MSFE), following [15]:

$$\text{MSFE}(\lambda_i) = \frac{1}{T_2 - T_1} \sum_{t=T_1}^{T_2-1} \|\widehat{\mathbf{y}}_{t+1}^{\lambda_i} - \mathbf{y}_{t+1}\|^2$$

where $\mathbf{y}_t$ is a $k \times 1$ vector, $\|\cdot\|$ is the L_2 norm, T_1 and T_2 are the time indices and λ_i's are the gridpoints. This procedure divides the training set (from 1 to T_3) into three parts: the initialization set (from 1 to T_1), the penalty parameter selection set (from $T_1 + 1$ to T_2) and the forecast evaluation set (from $T_2 + 1$ to T). Each grid point of λ is used to construct a model by the initialization set, which will then be used to calculate one predicted value. Next, repeat the above steps with adding one time period ahead to the observation set . Repeat this procedure $T_2 - T_1$ times until it reaches the tail of the selection set. Then obtain an MSFE by comparing the predicted values and the actual data on the selection set. This rolling method lets the MSFE be robust to random variation in data. Repeat the above process for all gridpoints and obtain all corresponding MSFEs. Choose the parameter which gives the smallest MSFE. MSFE rather than information criteria (AIC or BIC) is applied because the estimated residual sample covariance matrix of IC's might perform poorly in high dimensional settings [12].

D. Forecast Evaluation

Once $\widehat{\lambda}$ is chosen by the traning set, predicted values on the testing set can be calculated using the direct multi-step ahead approach by [16]. Solve $\widehat{\mathbf{v}}$ and $\widehat{\Phi}$ by least squares for a h-step ahead forecast:

$$\text{minimize}_{\mathbf{v},\Phi} \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{v} - \sum_{l=1}^{p} \Phi^{(l)} \mathbf{y}_{t-h-l+1}\|^2 + \lambda \|\Phi\|_1.$$

The h-step ahead predicted values are calculated as:

$$\widehat{\mathbf{y}}_{t+h} = \widehat{\mathbf{v}} + \sum_{l=1}^{p} \widehat{\Phi}^{(l)} \mathbf{y}_{t-l+1}.$$

MSFE is applied to evaluate the prediction accuracy on the testing set. Mean absolute deviation percent (MADP) is not suggested since MADP may be misleading when there are observations close to zero, which may inflate MADP. The MSFE of h-step ahead forecast for marginal series k is defined as

$$\text{MSFE}_k = \frac{1}{T - T_3 - h + 1} \sum_{t=T_3}^{T-h} (\widehat{\mathbf{y}}_{k,t+h} - \mathbf{y}_{k,t+h})^2.$$

where $\mathbf{y}_{k,t+h}$ denotes the kth element of $\mathbf{y}_{t+h}$.

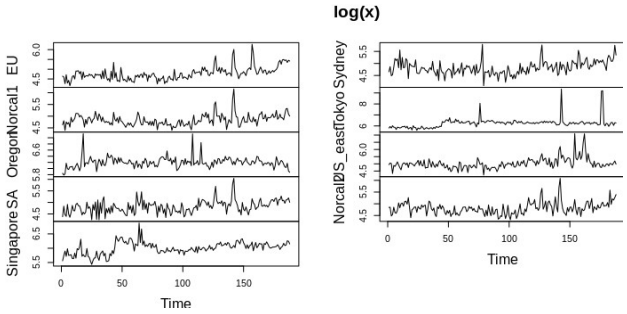

Fig. 1. Logged number of attacks for 9 servers per day.

III. Case Study

A. Data Description

This case study analyzes the data collected by Daniel Blander, a contract information security and risk management specialist who setup up several instances by Amazon Web Service (AWS) virtual honeypots across the world and let `iptables` capture packets. `iptables` is a program recording the firewall traffic. This data was provided to the blog of a book named "Data Driven Security [17]." This open source AWS Honeypot dataset contains information about the attack time, targeted host, attackers' IP addresses and origin countries [18].

This dataset with 9 honeypot hosts (EU, Norcal1, Oregon, SA, Singapore, Sydney, Tokyo, US-east and Norcal2) was captured throng 03/03/2013 to 09/08/2013 (451,581 data points contained). The first and last days are dropped because observation time did not last 24 hours. Fig. 1 presents the time series plots of log-transformed number of attacks per day for these 9 servers.

B. Data Pre-Processing

This study begins with a pre-processing filtering of raw data of 9 hosts with 188 observation days. The data were severely right-skewed, so log-transformation is taken. The data (X_t), a 188×9 matrix, is divided into two sets: the training set from day 1 to 150 and the testing set from day 151 to 188. The training set is divided further into three equal parts: the initialization set from day 1 to 50, the penalty parameter selection set from day 51 to 100 and the forecast evaluation set from day 101 to 150.

Similarly with how the penalty parameter λ is chosen, the LRD parameter d is calculated by Brent's `fmin` Algorithm [19] at each step for each marginal series. The rolling cross-validation technique is also applied to determine $\hat{d}$ by the minimzier of 1-step ahead MSFE on the selection set. Then LRD is removed as $Y_{k,t} = (1-B)^{-\hat{d}} X_{k,t}$, where $k = 1, \cdots, 9$ for the 9 hosts used in this study.

The BigVAR procedure is used on the series X_t as well as the LRD removed series Y_t. Standardization with mean 0 and unit variance is required for these 9 series to ensure that the LASSO conducts variable selection correctly. The number of lags for the VAR model is selected as $p = 4$. When $p = 7$, the coefficient matrices for lags greater than 4 appear to be almost zeros. The penalty parameter for BigVAR, λ, is set to have a depth of 50 with 10 grid points.

The predicted values are transformed back to the original scale by multiplying the standard deviation and adding mean. If LRM was removing in the beginning, it was added back by using the procedure in Section II-A.

IV. Computational Results

This section presents (i) whether LRM is necessary and (ii) whether dependence between honeypot hosts should be included in modeling the honeypot data.

A. Results of LRM

The fractional-differencing parameter, denoted by d for each host computed by maximum likelihood method is presents in Table I at the last column. It is noted that the parameter d for three hosts located in Tokyo, US-east and Oregon are very small so that the prediction accuracy does not improve with LRM. All other hosts have $d > 0.2$, and their MSFEs have decreased when LRM are considered. In particular, Sydney's d is estimated to be very close to 0.5, which is on the borderline to ensure stationarity. This suggests there possibly be a very strong long memory behavior in terms of number of cyberattacks. Any models without considering LRM may result in more complicated than what is necessary using the fractional integration. Based on the datasets, when LRM is not considered, there are 148 nonzero coefficients chosen by the BigVAR. In contrast, only 39 nonzero coefficients are needed when LRM is considered.

The LRM can also be found from autocorrelation function (ACF) plots. Fig. 2 presents two ACF plots. The one on the left is for the logged number of attacks per day for Singapore before taking the LRM off, and the one on the right is the acf after taking the LRM off. The ACF is persistent before taking the LRM off, which would typically requires more complicated models to accommodate the time dependency. On the other hand, the ACF after taking the LRM off decrease to zero fast. A simpler model usually is enough to model the leftover time dependency in the data.

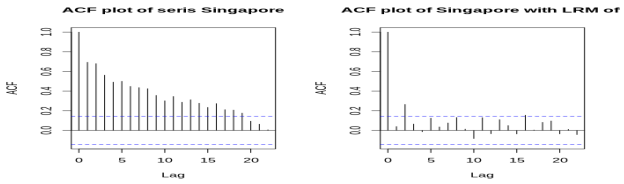

Fig. 2. ACF plots for Singapore before and after taking LRM off.

The findings suggest that LRM exist for most cyberattack data. This phenomena may be a result of how an actual malware spread out in the Internet. If there is one successful malware, its effect would stay on the Internet for a considerable amount of time. Therefore, considering LRM using fractional integration would allow concise modeling and better prediction.

B. Results of Dependence Between Honeypots

As shown in Table I, when LRM are considered, all the hosts except Sydney have very close MSFE's with full and diagonal models. It suggests that there is very weak, if any, linear dependence among these 9 hosts. Since these 9 host are not directly connected, this result is not surprising. As a result, the simpler BigVAR model with diagonal coefficient matrices is enough to establish the honeypot study.

It is worth noting that Sydney has obvious advantages with diagonal modeling. The MSFE decreased from 0.1374 of full model to 0.0638 of diagonal model. This may be explained by the very strong d value of the LRM, which suggests that this time series is possibly nonstationary. In this case, the fractional integration is not enough to model the time dependence. Additional autoregressive (AR) lags are needed and information from other hosts would be unnecessary. Fig.3 presents the sparsity plots for full coefficient matrix and diagonal coefficient matrix, respectively, both with considering

TABLE I
MSFE of Full and Diagonal BigVAR coefficient matrix with and without LRM for 9 hosts

Hosts	LRM not considered		LRM considered		
	Full	*Diag*	*Full*	*Diag*	*d*
EU	0.2697	0.2716	**0.2540**	0.2544	0.345
Norcal1	0.0512	0.0496	**0.0445**	0.0448	0.375
Oregon	0.0138	**0.0137**	0.0138	0.0138	0.119
SA	0.0626	0.0617	**0.0567**	**0.0567**	0.244
Singapore	0.0155	0.0154	0.0133	**0.0132**	0.401
Sydney	0.1407	0.1414	0.1374	**0.0638**	0.499
Tokyo	0.4734	**0.4733**	**0.4733**	**0.4733**	0.000
US east	0.3829	**0.3578**	0.3848	0.3609	0.022
Norcal2	0.0734	0.0718	0.0692	**0.0688**	0.265

[a] This is analyzed based on 9 dimensions.
[b] Boldface blue numbers indicate the best forecast for the corresponding host.

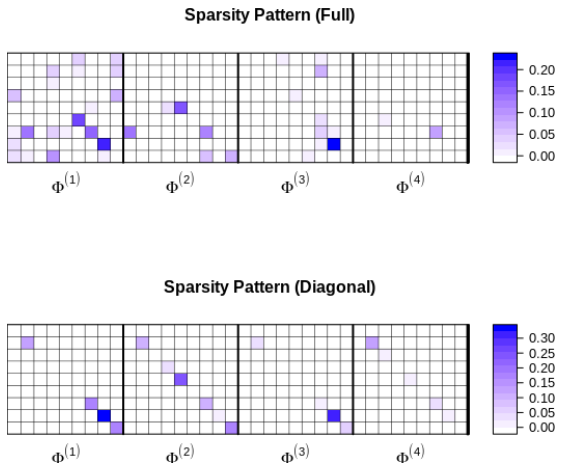

Fig. 3. Sparsity plots of full and diagonal coefficient matrix for BigVAR.

LRM. The (i, j)-th square of $\Phi^{(l)}$ indicates the impact of what happened in the j-th host on the i-th host with lag l. And the vertical orders as well as the horizontal orders per lag of squares in sparsity plot are the same as the orders in Table I. The corresponding estimated coefficients are presented in different shades of blue. The color key is provided next to the plot. The blank cell indicates no impact (i.e., zero coefficients) and a darker blue shade indicates a stronger impact.

The estimated value of moderately shaded cell in (4,1) element of $\Phi^{(1)}$ is 0.05, which indicates that what happened in the 1st host (EU) yesterday has some effect on the 4th host (SA) today. The (8,8) cells in $\Phi^{(1)}$ and in $\Phi^{(3)}$ both have the darkest color with the largest estimated value 0.22, which indicates that the strength of impacts of what happens yesterday and 3 days ago on what happens today in 8-th host (US-east) is the same. The estimated coefficients has decreased significantly from 39 of full matrix to 19 of diagonal matrix while the prediction performance is trivial. This suggests that simplification of BigVAR model to 9 independent equations will not compromise the prediction accuracy.

Besides considering dependence among other hosts, it may not be a good idea to reveal its own structure. For example, Sydney's coefficients in the diagonal model are all zero because the starting value of lambda grid is 0, which means all the coefficients by least square are zero and hence the series is suggested to be a white noise, while in the full model, Sydney has moderately self-dependence at lag 1 with $\widehat{\Phi}^{(1)}(6,6) = 0.19$, slight dependence on host 7th host (Tokyo) at lag 3 with $\widehat{\Phi}^{(3)}(6,7) = -0.03$ and slight dependence on 3rd host (Oregon) at lag 4 with $\widehat{\Phi}^{(4)}(6,3) = 0.08$. However, the more complicate latter model does not improve the performance compared to the previous simpler model. Similarly, EU is also in the case that nonzero coefficients in full model turns into zero in diagonal model while the MSFE of EU is not significantly increased.

V. Conclusion and Future Work

This paper concludes with two observations from the study. First, the honeypot data exhibits long range memory of each host. Considering fractional integration helps achieve concise modeling and better performance. This phenomena may be closely related to how malwares are spread over the Internet. Second, if honeypot hosts are not directly connected in the same network, considering dependence among hosts may not necessarily improve prediction accuracy.

A potentially interesting future study is using the proposed BigVAR procedure on a honeynet data observed for a longer time period. A higher data resolution for each site and more honeynets and studies could enable a more conclusive findings. For example, it would be interesting to correlate different hosts that may experience increase in number of cyberattack. Correlation of frequency of attack occurrence can also be investigated, including geographical distance between sites and potential impacts of cyberattacks.

References

[1] Online Trust Alliance (OTA), "Cyber incident and breach trends report," Internet Society, Tech. Rep., Jan. 2018.
[2] S. Morgan, "2019 cybersecurity almanac: 100 facts, figures, predictions and statistics," https://cybersecurityventures.com/cybersecurity-almanac-2019/, Feb. 6 2019.
[3] S. Calvert and J. Kamp, "More U.S. cities brace for 'inevitable' hackers," *Wall Street Journal*, Sep. 4 2018, [Online; accessed Feb. 16, 2019].
[4] N. Provos, "A virtual honeypot framework," in *Conf. 13th on USENIX Security Symposium*, Berkeley, CA, USA, May 2004, pp. 1–14.
[5] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1775–1789, Nov. 2013.
[6] Y. Chen, Z. Huang, S. Xu, and Y. Lai, "Spatiotemporal patterns and predictability of cyberattacks," *Public Library of Science (PLOS) One*, vol. 10, no. 6, pp. 1–19, May 2015.
[7] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling multivariate cybersecurity risks," *Journal of Applied Statistics*, vol. 45, no. 15, pp. 2718–2740, Jan. 2018.
[8] M. Xu, L. Hua, and S. Xu, "A vine copula model for predicting the effectiveness of cyber defense early-warning," *Technometrics*, vol. 59, no. 4, pp. 508–520, Nov. 2016.
[9] R. Baillie, "Long memory processes and fractional integration in econometrics," *Journal of Econometrics*, vol. 73, no. 1, pp. 5 – 59, Jul. 1996.
[10] N. Ford, D. Savostyanov, and N. Zamarashkin, "On the decay of elements of inverse triangular toeplitz matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 4, pp. 1288–1302, Oct. 2014.
[11] W. Nicholson, D. Matteson, and J. Bien, "BigVAR: Tools for modeling sparse high-dimensional multivariate time series," Feb. 2017, unpublished.
[12] ——, "Varx-l: Structured regularization for large vector autoregressions with exogenous variables," *International Journal of Forecasting*, vol. 33, no. 3, pp. 627–651, Apr. 2017.
[13] A. Chiuso and G. Pillonetto, "Nonparametric sparse estimators for identification of large scale linear systems," in *Conf. 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, USA, Dec. 2010, pp. 2942–2947.
[14] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, pp. 1–22, Feb. 2010.
[15] M. Babura, D. Giannone, and L. Reichlin, "Large bayesian vector auto regressions," *Journal of Applied Econometrics*, vol. 25, no. 1, pp. 71–92, Jan. 2010.
[16] T. Clark and M. McCracken, "Evaluating the accuracy of forecasts from vector autoregressions," Federal Reserve Bank of St. Louis, Working Papers 2013-010, Feb. 2013. [Online]. Available: https://ideas.repec.org/p/fip/fedlwp/2013-010.html
[17] J. Jacobs and B. Rudis, *Data-driven security: Analysis, Visualization and Dashboards*, 1st ed. Wiley Publishing, Jan. 2014.
[18] ——, "DDS dataset collection," https://datadrivensecurity.info/blog/pages/dds-dataset-collection.html, Jan. 2014, [Online; accessed Feb. 16 2019].
[19] R. Brent, *Algorithms for minimization without derivatives*, 4th ed. Dover Publications, Apr. 2013.