EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

Characterizing and Leveraging Granger Causality in Cybersecurity: Framework and Case Study

Van Trieu-Do[1], Richard Garcia-Lebron[2], Maochao Xu[3], Shouhuai Xu[4,*], and Yusheng Feng[1]

[1]Department of Mechanical Engineering, University of Texas at San Antonio, USA
[2]Department of Computer Science, University of Texas at San Antonio, USA
[3]Department of Mathematics, Illinois State University, USA
[4]Department of Computer Science, University of Colorado Colorado Springs, USA

Abstract

Causality is an intriguing concept that once tamed, can have many applications. While having been widely investigated in other domains, its relevance and usefulness in the cybersecurity domain has received little attention. In this paper, we present a systematic investigation of a particular approach to causality, known as Granger causality (G-causality), in cybersecurity. We propose a framework, dubbed Cybersecurity Granger Causality (CGC), for characterizing the presence of G-causality in cyber attack rate time series and for leveraging G-causality to predict (i.e., forecast) cyber attack rates. The framework offers a range of research questions, which can be adopted or adapted to study G-causality in other kinds of cybersecurity time series data. In order to demonstrate the usefulness of CGC, we present a case study by applying it to a particular cyber attack dataset collected at a honeypot. From this case study, we draw a number of insights into the usefulness and limitations of G-causality in the cybersecurity domain.

Received on 08 March 2021; accepted on 09 May 2021; published on 11 May 2021

Keywords: Granger Causality, Causality, Cyber Attack Forecasting, Cyber Attack Rate, Time Series

doi:10.4108/eai.11-5-2021.169912

1. Introduction

Cyber attacks have become a big threat against the modern society in many aspects, such as critical infrastructure, economy, and citizen privacy. According to a 2019 report by Symantec [1], a compromised credit card can be sold/purchased for up to US$45 in the underground market, whereas compromised websites can be sold/purchased for up to US$2.2 million each month. According to a 2019 report by ForgeRock [2], 2.8 billion consumer data records are breached in 2018, costing more than US$654 billion to U.S. organizations; the report also states that in the first quarter of 2019, cyber attacks against the U.S. financial services sector cost more than US$6.2 billion. These huge damages call for studies to understand and characterize cyber attacks from various perspectives and at various levels of abstractions.

Most studies on cyber attacks focus on *microscopic* levels of abstractions (e.g., how to defend against a particular attack). These studies are absolutely important because they provide the necessary building-block solutions. However, understanding and characterizing cyber attacks from *macroscopic* levels of abstractions is equally important but much less investigated. Such macroscopic-level studies are important because they would offer insights towards holistic solutions to defending cyber attacks.

One particular kind of macroscopic study is to forecast (i.e., predict) cyber attacks at macroscopic levels, so as to achieve what may be called *predictive* situational awareness. There have been a number of studies in both *univariate* time series analysis in the cybersecurity domain (e.g., [3–14]) and *multivariate* time series analysis in the cybersecurity domain (e.g., [7, 15–17]). The present study belongs to this category, but initiating a new perspective of research. Specifically, we investigate the usefulness of *causality* in cybersecurity. Since the notion of causality is elusive,

*Corresponding author. Email: sxu@uccs.edu

we focus on a particular approach known as Granger Causality (G-Causality) [18], which can be understood as follows: If one time series can be leveraged to help predict another time series more accurately than using the historic data of the latter alone to predict it, then the former is said to G-cause the latter. We call the former time series a "helper" because it can used to help predict the latter more accurately.

Our Contributions. This paper makes two contributions. First, we initiate the investigation on the usefulness and limitations of G-causality in cyber attack rate time series. We propose a framework, dubbed Cybersecurity Granger-Causality (CGC) with a range of intuitive research questions, which can be adopted or adapted to characterize and leverage G-causality in other kinds of cybersecurity time series data. In order to formalize research problems, we propose using a graph-theoretic representation of *G*-causality between time series. In particular, CGC aims to help achieve predictive cybersecurity situational awareness and therefore possibly proactive defense (e.g., allocating more defense resources when predicting that there will be more incoming attacks). To achieve this, we recognize that one key research issue is to select an appropriate number of good *helpers* at a proper network resolution (e.g., /8 vs. /24), where a *helper* is, as mentioned above, a time series that G-causes the time series in question. This research issue goes beyond the original G-causality framework [18]. We systematically address this issue by considering multiple factors and models. To the best of our knowledge, this is the first study in characterizing the usefulness and limitations of G-causality in predictive cybersecurity situational awareness.

Second, in order to demonstrate the usefulness of the framework, we conduct a case study by applying it to a dataset collected by a low-interaction honeypot. This case study enables us to draw a number of insights, such as the following. (i) For measuring cyber attack situational awareness, network resolution matters and using a higher resolution (e.g., /24) would be better than using a lower resolution (e.g., /16 or /8). (ii) Cybersecurity posture at the /16 and /24 network resolutions do change over a period of time, albeit slowly. (iii) G-causality is widely exhibited by cyber attack rate time series at multiple network resolutions, hinting that cyber attacks are not random. (iv) Bidirectional G-causality is widely exhibited at multiple network resolutions, suggesting that G-causality does not really capture the intuitive notion of causality, which should be unidirectional. (v) G-causality is widely exhibited *across* network resolutions; this represents an aspect that also goes beyond the original G-causality framework [18]. (vi) Leveraging bidirectional G-causality leads to higher prediction accuracy than leveraging unidirectional G-causality, especially when the time series in question are dense or correspond to low-resolution networks. This suggests that G-causality is useful despite that it does not really capture the intuitive notion of unidirectional causality. (vii) When leveraging G-causality to predict time series, using an excessive number of helpers can decrease prediction accuracy. This highlights the importance of selecting an appropriate number of *helpers*. (viii) When time series are dense, a smaller *p*-value incurred in the G-causality test, which hints a stronger degree of G-causality, would lead to more accurate predictions.

Related Work. The present study falls into the field of cybersecurity data analytics [19–27], which is a sub-field of the emerging Cybersecurity Dynamics [28–38]. More specifically, the present study falls into the sub-field of *multivariate* time series analysis [7, 15–17] of cybersecurity data analytics. There are studies on *univariate* time series analysis of cybersecurity data analytics, such as [3–14, 39]. However, these studies do not consider causality. In this paper we investigate a new aspect of cybersecurity data analytics, namely G-causality. Although G-causality is widely investigated in many domains (e.g., finance and economics [40], biology [41], social behaviors [42], and wireless communications [43]), its relevance to the cybersecurity domain is little investigated. The only exception we are aware is [44], which applies G-causality to confirm the presence of TCP flooding attacks. By contrast, we initiate the study on the usefulness and limitations of G-causality in predictive cybersecurity situational awareness, which is different from what is studied in [44]. Another related prior study is [45], which uses Bayesian networks to predict next attack steps in the context of intrusion detection. By contrast, we do not consider Bayesian networks.

Paper Outline. Section 2 reviews preliminary knowledge on the Auto-Regressive model and the notion of G-causality. Section 3 presents the CGC framework. Section 4 describes the case study. Section 5 discusses the limitations of the present study. Section 6 concludes the paper. Table 1 summarizes the main notations that are used throughout the paper.

2. Preliminaries

2.1. The Auto-Regressive (AR) Model

AR is a widely used statistical model, which leverages temporal correlations of a time series to predict its future values [46]. AR uses *linear regression* to predict future values as a function of ℓ past observation values (indicating how far one looks back), where ℓ is the order of the AR model or *lag*. Formally, the AR model for time

Table 1. Notations

Notation	Description
n	the number of networks (at a resolution) waging attacks
T	the time horizon at a certain resolution (e.g., days)
T_C	current time $1 \leq T_C \leq T-1$
$X_i(T_C)$	the time series representing the number of attacks waged from network i up to time T_C, with $X_i(T_C) = (x_{i,t})_{1\leq t\leq T_C}$ and $1 \leq i \leq n$
$X(T_C)$	$X(T_C) = \{X_1(T_C), \ldots, X_n(T_C)\}$
$X'(T_C)$	the subset of stationary time series of $X(T_C)$
$X''(T_C)$	the subset of stationary time series $X'(T_C)$ that are also associated with G-causality
$\hat{x}_{i,t}, \hat{y}_{i,t}$	predictions of x_{it} and y_{it}, respectively
ℓ	the lag value (i.e., time steps used in a prediction model)
$A_{X_i,X_{i,t}}$	coefficients of time series X_i w.r.t. itself
$A_{X_i,X_{j,t}}$	coefficients of time series X_i w.r.t. X_j
err, ξ	white noises
$p_{ij}(T_C)$	the p-value in the F-statistic that $X_i(T_C)$ G-causes $X_j(T_C)$ when applicable; $p_{ij} = \perp$ when not applicable
$G(T_C)$	$G(T_C) = (X(T_C), E(T_C), W(T_C))$ is G-causality graph, where $X(T_C)$ is the vertex set (representing time series), $(X_i(T_C), X_j(T_C)) \in E(T_C)$ means $X_i(T_C)$ G-causes $X_j(T_C)$, and $W(T_C) = (p_{ij}(T_C))_{1\leq i,j\leq n}$
$G_{[r]}(T_C)$	G-causality graph $G(T_C)$ at network resolution r; e.g., $r \in \{/8, /16, /24\}$

series $X_i = (x_{i,t})_{t=1,2,\ldots}$ is:

$$x_{i,t} = \beta_0 + \sum_{k=1}^{\ell} \beta_k x_{i,t-k} + \xi_{i,t}, \tag{1}$$

where $\beta_0, \ldots, \beta_\ell$ are coefficients and $\xi_{i,t}$ is a white-noise random variable (i.e., independent and identically distributed normal random variable with mean 0). In this paper, $x_{i,t}$ is the number of attacks that are waged from network i at time t.

2.2. G-causality

The notion of G-causality is named after its inventor Clive Granger and aims to capture causal relations between time series [18]. It is introduced to predict time series in the economics domain and later adapted to other domains [47–50]. It is defined for *stationary* time series, whose statistical properties (e.g., mean, variance, co-variance) do not change with time (cf. e.g., [3, 5, 51]). In practice, stationarity may be tested based on the first and second moments, sometimes known as *wide-sense stationarity*. There are many methods for testing whether a time series is stationary or not (e.g., Phillips-Perron [52] and Augmented Dickey-Fuller [53]).

As mentioned above, in this paper $X_i = (x_{i,t})_{t=1,2,\ldots}$ represents cybersecurity time series, such as the *cyber attack rate* time series [3]. Intuitively, X_i is said to Granger-cause or G-cause X_j, where $i \neq j$, if the past observation values of X_i contain some information that can be leveraged to predict future values of X_j more accurately than predicting X_j by only leveraging its past observation values [18]. Similar to the lag ℓ in the AR model, the number of the past observation values of X_i, which are leveraged to predict future values of X_j, is also called *lag* and denoted by ℓ. Since a large ℓ may cause over-fitting and a small ℓ may cause auto-correlation errors [54], it is important to select an appropriate ℓ via some criterion, such as Bayesian Information Criterion (BIC) [55] or Akaike Information Criterion (AIC) [56], meaning that the optimal ℓ is the one that minimizes the AIC or BIC function.

Formally, G-causality is defined using the linear Vector Auto-Regressive model (VAR) over multivariate time series. In order to highlight the idea, let us consider the example of bi-variate VAR model, while noting that the idea is equally applicable to other multivariate time series. The bi-variate VAR model (or 2VAR) involves two time series X_i and X_j with lag ℓ and is described as:

$$x_{i,t} = \alpha_i + \sum_{k=1}^{\ell} A_{X_i,X_{j,t-k}} x_{j,t-k} + \sum_{k=1}^{\ell} A_{X_i,X_{i,t-k}} x_{i,t-k} + \text{err}_{i,t}, \tag{2}$$

$$x_{j,t} = \alpha_j + \sum_{k=1}^{\ell} A_{X_j,X_{i,t-k}} x_{i,t-k} + \sum_{k=1}^{\ell} A_{X_j,X_{j,t-k}} x_{j,t-k} + \text{err}_{j,t}. \tag{3}$$

where the $A_{*,*}$'s are regression coefficients and $\text{err}_{i,t}$ and $\text{err}_{j,t}$ are white-noise errors (i.e., independent and identically distributed normal random variables with mean 0).

The VAR model is used to test G-causality between X_i and X_j as follows. The null hypothesis is that X_i does not G-cause X_j, namely that $A_{X_j,X_{i,t-k}} = 0$ for $1 \leq k \leq \ell$ or X_i has no impact on predicting X_j [18]. To test this, one may use the F-statistic hypothesis test [57]. Time series X_i is said to G-cause X_j if the null hypothesis is rejected, meaning that the p-value in the F-statistic is less than 0.05, which is a widely-used significant level. The same method is used to test whether X_j G-causes X_i or not. G-causality is *not* necessarily symmetric, meaning that X_i G-causing X_j does not necessarily mean X_j G-causing X_i.

2.3. Prediction Accuracy Metric

In order to evaluate prediction accuracy, we propose adopting the standard metric known as Symmetric Mean Absolute Percentage Error (SMAPE) [58]. Let $(x_{i,t}, \ldots, x_{i,t+\delta})$ be the observation values of a time series and $(\hat{x}_{i,t}, \ldots, \hat{x}_{i,t+\delta})$ be their respective prediction values, where t is the time at which prediction starts. Then, $\text{SMAPE} = \frac{1}{\delta+1} \sum_{z=t}^{t+\delta} \frac{|x_{i,z} - \hat{x}_{i,z}|}{(|\hat{x}_{i,z}| + |x_{i,z}|)/2}$. This metric is chosen because of its robustness in accommodating $x_{i,t} = 0$, which is often encountered in cybersecurity.

3. The Cybersecurity Granger-Causality (CGC) Framework

The CGC framework aims to characterize the presence and utility of G-causality in the context of cybersecurity time series, as illustrated by cyber attack rate time series. The framework is designed with the mindset that it can be adopted or adapted to study G-causality in other kinds of cybersecurity time series data of a similar nature. As highlighted in Figure 1, CGC has 4 modules. As elaborated below, these modules are associated with a unique set of Research Questions (RQs).

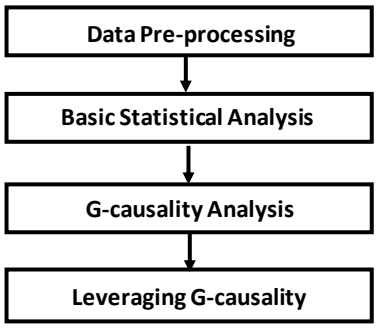


Figure 1. The CGC framework

3.1. Data Pre-processing

The input to the framework is some cybersecurity data, such as the cyber attacks observed by cyber defense instruments (e.g., honeypots [59, 60] or network telescope [4]) over a period of time. In order to represent the data as time series, we propose considering a discrete time horizon $t = 1, 2, \ldots, T$ at some time resolution (e.g., hour or day). The dataset contains the attacking IP addresses that wage attacks against some victims. Depending on the semantic richness of the dataset, the attacks may be further divided into, for example, different types (e.g., denial-of-service or not). The basic CGC framework focuses on coping with cyber attack rates, while leaving the treatment of richer information to its extensions (partly because we have no such semantically rich datasets). We propose grouping the attacking IP addresses into networks at some resolution, such as /8, /16 or /24 networks. Recall that a /8, /16, and /24 network consists of 2^{24}, 2^{16}, and 2^8 IPv4 addresses, respectively. Let n denote the number of networks at a resolution in question (e.g., $n = 2^8$ at the /8 network resolution). For network i ($1 \leq i \leq n$) at a resolution, an appropriate pre-process is often needed to derive a time series $X_i(T_C) = (x_{i,t})_{1 \leq t \leq T_C}$, where T_C ($1 \leq T_C \leq T$) is the current time and $x_{i,t}$ is the number of attacks that are waged from network i at time t.

3.2. Basic Statistical Analysis

Given the pre-processed set of n time series at a network resolution, denoted by $X(T_C) = \{X_1(T_C), \ldots, X_n(T_C)\}$, we propose conducting some basic analyses to deepen the understanding of the dataset. We propose associating this module with the following Research Questions (RQs):

- RQ1: What is the overall cyber attack situational awareness?
- RQ2: What is the evolution of the situational awareness?
- RQ3: What is the tensity of attacks (e.g., the number of attacks per time interval, the sparsity of the time series?
- RQ4: What are the characteristics of the time series?

The preceding basic statistical analysis is both necessary and important. This is because, for model-fitting purposes, when a time series is sparse (i.e., containing many zeros in its observation values), it should be eliminated from further analysis because state-of-the-art statistical techniques cannot cope with such sparse time series data. (Nevertheless, we note that innovative methods are emerging in order to deal with such sparse time series [16], which is orthogonal to the purpose of the present study.)

For the time series that are not sparse, we analyze their basic statistics (e.g., *mean*, *median*, *max*, and *variance*). In order to see how the basic statistical analysis can deepen our understanding of the data in question and guide us in modeling the data, we mention the following. If a time series has a mean value that is much smaller than its variance, then the time series cannot be modelled by the Poisson process but should be fitted with another appropriate model. If several time series exhibit a similar pattern (i.e., all increasing, decreasing, simultaneously changing), then they may be correlated with each other.

3.3. G-causality Analysis

Given a set of pre-processed time series at a network resolution (with sparse ones eliminated), namely

$X(T_C) = \{X_1(T_C), \ldots, X_n(T_C)\}$, this module proceeds as follows. First, test the stationarity of $X_i(T_C) \in X(T_C)$ because G-causality is defined over stationary time series. For this purpose, there are many methods (e.g., Phillips-Perron [52] or Augmented Dickey-Fuller [53]). Second, test the G-causality for every pair of stationary time series $(X_i(T_C), X_j(T_C))$ in $X(T_C)$ with $i \neq j$, while recalling that $X_i(T_C)$ G-causes $X_j(T_C)$ if the null hypothesis that $X_i(T_C)$ does not G-cause $X_j(T_C)$ is rejected in the F-statistic test. Let $p_{i,j}(T_C)$ denote the p-value in the F-statistic. Then, we only consider the time series with associated p-values that are smaller than 0.05, because such p-values indicate that the null hypothesis is not rejected.

Now we propose the notion of *G-causality graph*, which is a simple, directed, weighted graph representation of the G-causality relations between the time series. A G-causality graph is denoted by $G(T_C) = (X(T_C), E(T_C), W(T_C))$, where $X(T_C) = \{X_1(T_C), \ldots, X_n(T_C)\}$ is the vertex or node set that corresponds to the set of networks and represent their respective cyber attack rate time series, an arc $(X_i(T_C), X_j(T_C)) \in E(T_C)$ means a stationary time series $X_i(T_C)$ G-causes time series $X_j(T_C)$, each $(X_i(T_C), X_j(T_C)) \in E(T_C)$ is associated with a weight $p_{i,j}(T_C)$ which is the p-value mentioned above, and the p-values formulates a weight matrix $W(T_C) = (p_{i,j}(T_C))_{1 \leq i,j \leq n}$. With this graph-theoretic representation, $\mathcal{N}(X_j(T_C)) = \{X_i(T_C) : (X_i(T_C), X_j(T_C)) \in E(T_C)\}$ represents the set of neighbor nodes that G-cause $X_j(T_C)$, and the in-degree of a node $X_j(T_C)$ is $\deg(X_j(T_C), G(T_C)) = |\mathcal{N}(X_j(T_C))|$. Note that an isolated vertex or node means (i) the corresponding time series is not stationary or (ii) it has no G-causality relation to any other node. We propose associating this module with the following RQs:

- RQ5: What are the characteristics of G-causality at a single network resolution?
- RQ6: Is G-causality unidirectional or bidirectional?
- RQ7: Is the G-causality relation exhibited between network resolutions?

In order to simplify notations, we may omit the mentioning of T_C when discussing general concepts that are applicable to any T_C, such as the in-degree of node X_j in graph G, or when T_C is clear from the context. This leads to $G = (X, E, W)$ and simplifies notation $\deg(X_j(T_C), G(T_C))$ as $\deg(X_j, G)$. We may further use $X' \subseteq X$ to denote the set of nodes corresponding to stationary time series and use $X'' \subseteq X'$ to denote the set of nodes associated with a G-causality relation.

3.4. Leveraging G-causality

One important utility of G-causality is to leverage it to predict cyber attack rate time series to achieve predictive situational awareness and possibly proactive defense. Therefore, we propose associating this module with the following RQ:

- RQ8: How should one leverage G-causality to predict cyber attack rates?

The key research issue is to select an *appropriate* number of *good* helpers at the *proper* network resolution(s); this research issue goes beyond the notion of G-causality. In order to address this issue, we propose considering 4 factors: direction of G-causality (i.e., unidirectional vs. bidirectional), the number of helpers that are leveraged for prediction, p-value (i.e., small vs. medium vs. large), and layers of network resolutions (e.g., one vs. multiple layers). For either empirical or theoretical comparison purposes, these factors can be tied to any prediction model of interest. As examples, we propose considering 4 classes of models:

- AR: It leverages X_j itself to predict X_j. This model, as reviewed above, does not leverage G-causality (or helpers) at all and serves as the baseline model.
- GC($z+1$)VAR: This a family of models that are inherent to the notion of G-causality, by leveraging z time series (helpers), which G-cause X_j (i.e., neighbor nodes pointing to X_j in the G-causality graph), to predict X_j, where $1 \leq z \leq \deg(X_j, G)$ with $\deg(X_j, G)$ being the in-degree of node X_j in G-causality graph G. These models are elaborated in Algorithm 1 below.
- PenVAR: This model is elaborated below and aims to avoid overfitting and overparameterization of the standard VAR model *without* leveraging G-causality.
- GCPenVAR: This is a hybrid of PenVAR and GC($m+1$)VAR where $m = \deg(X_j, G)$, by leveraging *all* of the times series that G-cause X_j (i.e., all neighbors pointing to X_j in the G-causality graph) to predict X_j.

GC($z+1$)VAR models. Algorithm 1 leverages z helpers, namely z neighbors that G-cause X_j to predict X_j, where $1 \leq z \leq m$ and $m = \deg(X_j, G)$. The heuristic used in Algorithm 1 is to leverage the z helpers with the smallest p-values in the G-causality test, so as to avoid the combinatorial explosion of $\binom{m}{z}$ where m can be large. The heuristic corresponds to the greedy algorithm because $p_{i,j}$ may be interpreted as the degree of G-causality, meaning that the smaller the $p_{i,j}$, the stronger the G-causality. Specifically, suppose we sort the p-values corresponding to X_j's neighbors increasingly as

Algorithm 1 The GC($z+1$)VAR algorithm for predicting x_{j,T_C+1}, $1 \le j \le n$, by leveraging z ($z \ge 1$) of the X_i's in that G-cause X_j as helpers

INPUT: G-causality graph
$G(T_C) = (X(T_C), E(T_C), W(T_C))$ where
$X(T_C) = \{X_1(T_C), \ldots, X_n(T_C)\}$ and
$W(T_C) = (p_{i,j}(T_C))_{1 \le i,j \le n}$
OUTPUT: Predictions $\hat{x}_{j,T_C+1}$ for $1 \le j \le n$

for $j = 1$ to n **do**
  $\mathcal{N}(X_j(T_C)) \leftarrow \{X_i(T_C) : (X_i(T_C), X_j(T_C)) \in E(T_C)\}$ and denote its cardinality by m for ease of reference
  /* $m = \deg(X_j(T_C), G(T_C))$ */
  if $m \ge z$ **then**
    Use the $(z+1)$-variate VAR model to fit $X_j(T_C)$ by leveraging X_j and the z helpers corresponding to the z smallest p-values, according to Eq.(3) and an appropriate model selection criterion
    Use the fitted model to predict $\hat{x}_{j,T_C+1}$
  else
    $\hat{x}_{j,T_C+1} \leftarrow \perp$ /* X_j is not stationary or does not have enough G-causality helpers */
  end if
end for
Return $\hat{x}_{j,T_C+1}$ for $1 \le j \le n$.

$p_{1,j}, \ldots, p_{m,j}$, where $j \notin \{1, \ldots, m\}$. Figure 2 illustrates 4 scenarios of GC($z+1$)VAR for predicting X_j, where $z \in \{1, 2, 3, m\}$:

- GC2VAR: Leverage X_1 with the smallest p-value as helper to predict X_j via the bivariate VAR model.
- GC3VAR: Leverage X_1 and X_2 as helpers to predict X_j via the 3-variate VAR model.
- GC4VAR: Leverage X_1, X_2 and X_3 as helpers to predict X_j via the 4-variate VAR model.
- GC($m+1$)VAR: Leverage all of the m neighbor $X_{1,j}, \ldots, X_{m,j}$ as helpers to predict X_j via the $(m+1)$-variate VAR model.

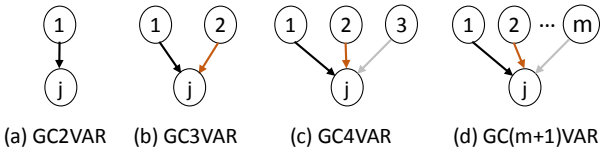


Figure 2. Illustration of GC($z+1$)VAR, where prefix "GC" indicates leveraging G-causality and arrows are colored to indicate their respective p-values with $p_{1,j} \le p_{2,j} \le \ldots \le p_{m,j}$.

PenVAR and GCPenVAR models. The PenVAR model can predict d-variate time series altogether, meaning $1 \le d \le n$ where n is the number of networks at a resolution. Without loss of generality, let $\{\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{d,t}, x_{j,t})^\top\}_{t=1,2,\ldots}$ denote the $(d+1)$-dimensional vector time series. The standard VAR model can be represented as

$$\boldsymbol{x}_t = \boldsymbol{\nu} + \sum_{l=1}^{\ell} \boldsymbol{\Phi}^{(l)} \boldsymbol{x}_{t-l} + \boldsymbol{u}_t \tag{4}$$

where ℓ is the lag, $\boldsymbol{\nu}$ represents a $(d+1) \times 1$ intercept vector, $\boldsymbol{\Phi}^{(l)}$ denotes a $(d+1) \times \ell$ coefficient matrix, and $\boldsymbol{u}_t$ is a $(d+1) \times 1$ white noise vector (i.e., independent and identically distributed normal random vector with mean $\mathbf{0}$ and covariance matrix Σ_μ, namely a diagonal matrix with elements representing variances). The model fitting is to minimize the least square errors

$$\min_{\nu, \boldsymbol{\Phi}^{(l)}} \left\| \boldsymbol{x}_t - \boldsymbol{\nu} - \sum_{l=1}^{\ell} \boldsymbol{\Phi}^{(l)} \boldsymbol{x}_{t-l} \right\|_F^2 \tag{5}$$

which involves $(d+1) + \ell(d+1)^2$ regression parameters, where $\|\cdot\|_F^2$ represents the Frobenius norm. This means that the standard VAR model is likely unstable or infeasible when d is large (i.e., high dimensions), which motivates PenVAR.

Let $\boldsymbol{\Phi} = \left(\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(\ell)}\right)$. The PenVAR model reduces the parameter space and has the following optimization objective [61]:

$$\min_{\nu, \boldsymbol{\Phi}^{(l)}} \left\| \boldsymbol{x}_t - \boldsymbol{\nu} - \sum_{l=1}^{\ell} \boldsymbol{\Phi}^{(l)} \boldsymbol{x}_{t-l} \right\|_F^2 + \lambda \|\boldsymbol{\Phi}\|_1, \tag{6}$$

where $\lambda > 0$ is the penalty parameter and $\|\boldsymbol{\Phi}\|_1$ represents the L_1 norm. The penalty parameter $\hat{\lambda}$ is selected to minimize the one-step ahead mean square prediction error (MSPE) [61]:

$$\text{MSPE}(\lambda) = \frac{1}{b} \sum_{t=k+1}^{k+b} \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2, \tag{7}$$

where $\|\cdot\|_2$ is the L_2 norm.

The GCPenVAR model is obtained by incorporating G-causality into the PenVAR model. Unlike the PenVAR models that predict a vector of d-variate time series simultaneously, the GCPenVAR model, like GC($z+1$)VAR, leverages z helpers only, where $1 \le z \le d$.

4. Case Study

Now we present a case study by applying the framework to analyze a specific dataset collected by a low-interaction honeypot. A honeypot is a cyber

defense instrument that emulates real-world Internet-based vulnerable services at a number of IP addresses. Since these services are exclusively set up for attracting attacks (i.e., no legitimate services are associated with these IP addresses), it is a widely-accepted practice to treat the incoming, unsolicited network traffic as attacks [4, 5, 7, 62–69]. The honeypot in question monitors 1,024 IP addresses and runs a number of low-interaction honeypot programs, including *Honeyd* [70] and *Nepenthes* [71]. The notion of *low-interaction* means that the honeypot only partly emulates the services in question, which explains why the dataset is only used for analyzing the cyber attack rate (rather that cybersecurity semantically richer analyses). The dataset is collected between 2/6/2014 and 5/13/2014 (i.e., $T = 97$ days). Although the dataset is six years old, it is sufficient for demonstrating the usefulness of CGC (while noting that it is often difficult for academic researchers to have access to such data). Researchers with newer datasets can adopt or adapt CGC to analyze their own datasets.

4.1. Data Pre-processing

The raw data collected by the honeypot is in the standard PCAP format, which is converted into the *flow* format to represent attacks. A flow is described by a tuple of five fields: source (i.e., attacker in this paper) IP address, destination (i.e., victim or honeypot in this paper) IP address, source port number, destination port number, and protocol [72]. For converting PCAP data into flow data, a widely-used tool, known as the Yet Another Flowmeter (YAF) with super_mediator [72], is used. This process involves two parameters: the flow idle time and the flow lifetime. We use a widely-used combination of them: 60 seconds for the flow idle time and 300 seconds for the flow lifetime [3–5, 69].

The source (i.e., attacker) IP addresses are grouped into networks at three resolutions: (i) /8 networks, with each consisting of 2^{24} IP addresses; (ii) /16 networks, with each consisting of 2^{16} IP addresses; and (iii) /24 networks, with each consisting of 2^8 IP addresses. Since the framework is equally applicable to any network resolution, we extend the notation $G(T_C) = (X(T_C), E(T_C), W(T_C))$ to accommodate network resolutions as subscripts $_{[r]}$ where $r \in \{/8, /16, /24\}$. For example, $X_{[/8]}(T_C)$ denotes the set of nodes (i.e., the attack rate time series) at the /8 network resolution, $X'_{[/8]} \subseteq X_{[/8]}$ denotes the corresponding subset of stationary time series, and $X''_{[/8]} \subseteq X'_{[/8]}$ denotes the corresponding subset of stationary time series that are associated with at least one G-causality.

In order to indicate cross-network-resolution analyses, we associate lower-resolution networks with their belonging higher-resolution networks. For example, $X'_{[/16 \in /8]}(T_C)$ denotes the stationary time series corresponding to the /16 networks that belong to a /8 network in question; $X''_{[/16 \in /8]}(T_C)$, $X'_{[/24 \in /16]}(T_C)$, and $X''_{[/24 \in /16]}(T_C)$ are similarly defined.

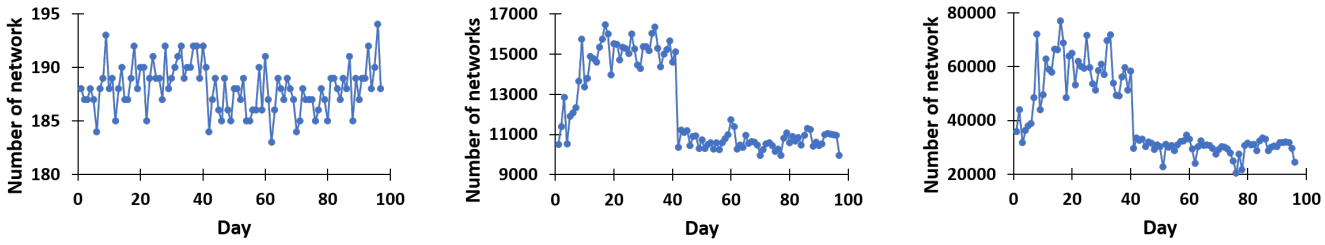
4.2. Basic Statistical Analysis

Characterizing the Overall Cyber Attack Situational Awareness (Answering RQ1). In order to draw insights into the network resolution(s) that would be more appropriate for characterizing the overall cyber threat situational awareness, Figure 3 plots the time series $|X_{[r]}(97)|$, namely the number of attacking networks at a resolution during the time horizon of $T = 97$ days. We make three observations.

First, on average over the $T = 97$ days, the honeypot observed: $|X_{[/8]}(97)| = 188$ or $188/2^8 = 73.44\%$ /8 attacking networks per day; $|X_{[/16]}(97)| = 12,292$ or $12,292/2^{16} = 18.76\%$ /16 attacking networks per day; and $|X_{[/24]}(97)| = 40,840$ or $40,840/(2^{24} - 4) = 0.24\%$ /24 attacking networks per day where "−4" is to exclude the 4 /24 networks corresponding to the honeypot itself. In other words, the attacking networks observed by the honeypot on a daily basis concentrate at a small percentage (0.24%) of /24 networks, which are scattered in a significant number of /16 networks and even more so in a large number of /8 networks. These metrics reflect the average cyber threat landscape situational awareness, especially that some networks are better managed than others and that network resolution matters when measuring the percentage of attacking networks.

Second, cumulatively over the $T = 97$ days, the honeypot observed: 204 or $204/2^8 = 79.69\%$ /8 attacking networks; 27,019 or $27,019/2^{16} = 41.23\%$ /16 attacking networks; and 842,642 or $842,642/(2^{24} - 4) = 5.02\%$ /24 attacking networks. These metrics reflect that as time goes by, more networks get compromised and then wage attacks against others. Nevertheless, within a significant period of time ($T = 97$ days), the attacking computers still concentrate at a relatively small number of /24 networks (5.02%), which are scattered in large numbers of /16 and /8 networks. This further suggests that some networks are much better managed than others because the honeypot does not observe any attack from 94.08% of /24 networks over $T = 97$ days.

Third, by contrasting the network resolutions, we observe that there is a substantial drop at $t = 41$ (or 3/18/2014) in terms of the numbers of /16 and /24 attacking networks. However, this substantial drop is at most slightly reflected in the number of /8 attacking networks. This means that the /16 and /24 networks that stop waging attacks at $t = 41$ belong to a small number of /8 attacking networks. Unfortunately, it is not clear what caused this drop at $t = 41$. One may



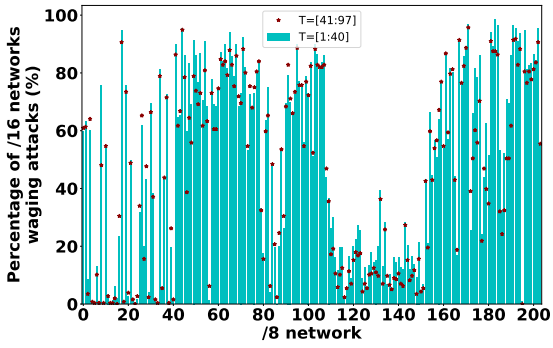
(a) $|X_{[/8]}(97)|$ (b) $|X_{[/16]}(97)|$ (c) $|X_{[/24]}(97)|$

Figure 3. Time series $|X_{[r]}(97)|$, where the x-axis represents time (unit: day) and the y-axis represents $|X_{[r]}(97)|$ (i.e., the number of networks at resolution $r \in \{/8, /16, /24\}$ that wage at least one attack during a day).

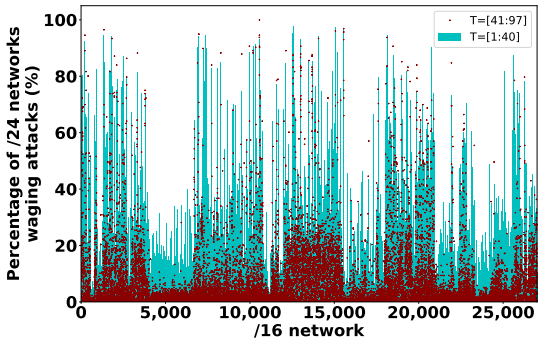
speculate that there is an Internet-wide operation in cleaning up botnets residing in a small number of /8 networks, we cannot find any information on such operations (one source of botnet takedown operations is https://en.wikipedia.org/wiki/Botnet). Another possible, perhaps less likely, scenario is that the honeypot does not capture most attacks in the Internet because it only has 1,024 IP addresses; this is less likely because the drop starting at $t = 41$ seems to be consistent for $t \in [41, 97]$.

Insight 1. For measuring cyber threat situational awareness, network resolution matters and /24 resolution would be more appropriate than /16 and /8, because attacking computers often belong to a small percentage of /24 networks that are scattered in many /8 networks.

Characterizing the Evolution of Cyber Threat Situational Awareness (Answering RQ2). In order to characterize the evolution of situational awareness, Figure 4a plots the percentage of the attacking /16 networks with respect to the /8 networks to which they belong, where time is divided into $t \in [1, 40]$ and $t \in [41, 97]$ because of the significant difference at $t = 41$ (cf. Figures 3b and 3c). We make two observations. First, there are 204 /8 attacking networks for $t \in [1, 40]$. Among them, 70 (or 34.31%) networks have no more than 25% of their belonging /16 networks waging attacks; 111 (or 54.41%) networks have more than 50% of their belonging /16 networks waging attacks; 64 (or 31.37%) networks have more than 75% of their belonging /16 networks waging attacks. On the other hand, there are also 204 /8 attacking networks for $t \in [41, 97]$, and a similar phenomenon is exhibited. This explains why Figure 3a does not show a significant drop at $t = 41$. Second, when compared with the percentage of attacking /16 networks during $t \in [1, 40]$, the percentage during $t \in [41, 97]$ exhibits the following: 45 (or 22.06%) /8 networks have more belonging /16 attacking networks; 20 (or 9.80%) /8 networks have the same number of belonging /16 attacking networks; 139 (or 68.14%) /8 networks have fewer belonging /16 attacking networks. This also explains why Figure 3a does not show a significant drop at $t = 41$.



(a) Percentage of /16 networks waging attacks.



(b) Percentage of /24 networks waging attacks.

Figure 4. Plots of the percentage of attacking networks at higher vs. lower network resolution during $t \in [1, 40]$ vs. $t \in [41, 97]$.

Figure 4b plots the percentage of attacking /24 networks with respect to the /16 networks to which they belong, with time $t \in [1, 40]$ vs. $t \in [41, 97]$. We make two observations. First, among the 27,009 /16 attacking networks during $t \in [1, 40]$, 24,948 (or 92.37%) networks have no more than 25% of their belonging /24 networks waging attacks; 450 (or 1.67%) networks have more than 50% of their belonging /24 networks waging attacks; 112 (or 0.41%) networks have more than 75% of their belonging /24 networks waging attacks. On the other hand, there are also 27,009 /16

attacking networks during $t \in [41, 97]$. Among them, 25,586 (or 94.73%) networks have no more than 25% of their belonging /24 networks waging attacks; 349 (or 1.29%) networks have more than 50% of their belonging /24 networks waging attacks; 79 (or 0.29%) networks have more than 75% of their belonging /16 networks waging attacks. Second, when compared with the percentage of /24 attacking networks during $t \in [1, 40]$, the percentage during $t \in [41, 97]$ exhibits the following: 7,427 (or 27.50%) /16 networks have more /24 attacking networks; 1,641 (or 6.08%) /16 networks have the same number of /24 attacking networks; 17,941 (or 66.48%) /16 networks have fewer /24 attacking networks. This explains why Figure 3b exhibits a significant drop at $t = 41$.

Insight 2. For most /16 and /24 networks, their security posture do change significantly during the $T = 97$ days.

Insight 2 hints that defenders have been active in detecting and cleaning up attacking computers, albeit it may take a long period of time (e.g., at least 40 days as shown by this dataset) in having a globally visible effect.

Characterizing the Tensity of Attacks (Answering RQ3). Figure 5 plots the number of networks waging attacks (y-axis, in log scale) during how many of the $T = 97$ days (x-axis). From Figure 5a, we observe that among the 2^8 possible /8 attacking networks during the $T = 97$ days, 52 (or 20.31%) networks do not wage attacks; 59 (or 23.05%) wage attacks for less than 7 days; 191 (or 74.61%) wage attacks at least for (not necessarily consecutive) 30 days; 181 (or 70.70%) wage attacks for more than 90 days; and 135 (or 52.73%) wage attacks during the entire $T = 97$ days. From Figure 5b, we observe that among the 2^{16} possible /16 attacking networks during the $T = 97$ days, 38,517 (or 58.77%) networks do not wage any attacks; 43,259 (or 66.01%) wage attacks for less than 7 days; 15,577 (or 23.77%) wage attacks at least for 30 days; 4,316 (or 6.59%) wage attacks more than 90 days; and 292 (or 0.45%) wage attacks during the entire $T = 97$ days. From Figure 5c, we observe that among the $2^{24} - 4$ possible /24 attacking networks during the $T = 97$ days, 15,934,570 (or 94.98%) networks do not wage attacks as per the honeypot; 16,647,331 (or 99.23%) wage attacks for less than 7 days; 19,962 (or 0.12%) wage attacks at least for 30 days; 1,280 (or 0.007%) wage attacks more than 90 days; and 51 (or 0.0003%) wage attacks during the entire $T = 97$ days.

If we define intense attacking networks as those which wage attacks more than 90 days of the entire $T = 97$ days, we observe 70.70% /8 attacking networks, 6.59% /16 attacking networks and 0.007% /24 attacking networks. This means that the intensity of attacking networks at different resolutions can be different at orders of magnitudes, which resonates Insight 1, which however copes with *all* attacking networks.

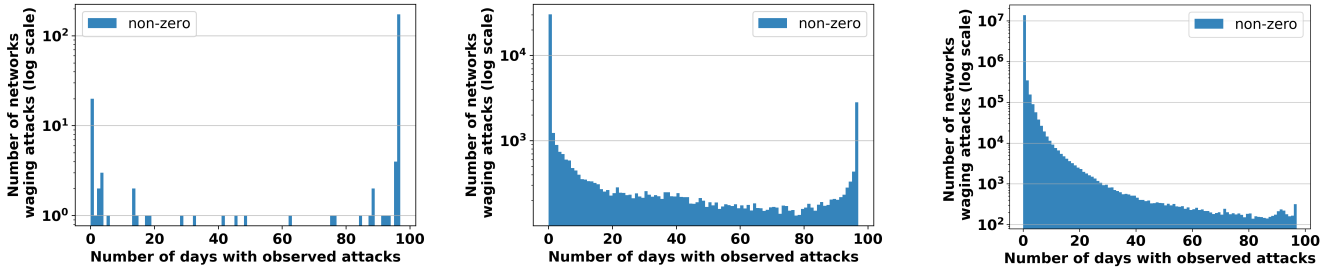
Insight 3. The percentages of *intense attacking networks* at different network resolutions are orders-of-magnitude different, which means that network resolution matters when characterizing the evolution of intense attacking networks.

Characterizing the Time Series (Answering RQ4). The preceding characteristics are made over the entire time horizon of the dataset. Since a core value of studying cyber attack data is to conduct prediction (i.e., forecasting). In the real world, a defender would use the data collected up to current time T_C to predict future cyber attack rates for time $t > T_C$. This means that for prediction purposes, we should characterize the time series in $X(T_C)$. Given that $T = 97$ days is the time horizon of the dataset, we propose starting prediction at $T_C = 90$ (i.e., predicting cyber attack rates for $t \in [91, 97]$). For model-fitting purposes, we propose considering the time series that have at least 30 (not necessarily consecutive) non-zero observations among the first 90 days, where 30 is rule-of-thumb in statistic modeling of time series data. This leads to $|X_{[/8]}(90)| = 190$ (/8 networks), $|X_{[/16]}(90)| = 15,244$ (/16 networks), and $|X_{[/24]}(90)| = 18,565$ (/24 networks).

Table 2. Basic statistics of times series $X_{[r]}(90)$'s, where LB (UB) stands for lower-bound or minimum (upper-bound or maximum) value of a statistic among the networks at resolution $r \in \{/8, /16, /24\}$.

Network resolution	Mean		Median		Variance		MAX	
	LB	UB	LB	UB	LB	UB	LB	UB
$r = /8$	0.689	186.9	0	223.5	0.473	10,608	3	255
$r = /16$	0.333	97.656	0	103.5	0	4,939.17	1	231
$r = /24$	0.333	83.844	0	65.0	0	1,623.504	1	189

Table 2 presents the summary statistics of the time series. By taking /8 networks as an example, we make the following observations. The 190 mean values of the $|X_{[/8]}(90)| = 190$ time series, namely $\sum_{t=1}^{90} x_{i,t}/90$ for $X_i(90)$ where $1 \le i \le 190$, fall into interval $[0.689, 186.900]$ (unit: attacks per day); their medians fall into $[0, 223.500]$ attacks per day (where fraction is caused by that $T_C = 90$ is an even number); their variances fall into $[0.473, 10,608]$; and their maximum numbers of attacks per day fall into $[3, 255]$, perhaps because the honeypot is small (i.e., 1,024 IP addresses). Note that the medians of some /8, /16, and /24 networks are 0, because these networks do not wage attacks for at least 45 days (50% of the 90 days). The variances of some /16 and /24 networks are 0 because a constant number of attacks are waged from these networks. Specifically, one /16 network wages 1 attack per day for the first 90 days; one /16 network wages 2 attacks per day for the first 90 days; and 31



(a) Tensity of /8 attacking networks **(b)** Tensity of /16 attacking networks **(c)** Tensity of /24 attacking networks

Figure 5. Plots of attack tensity: the number of networks (y-axis, in log scale) that waged attacks during x of the $T = 97$ days.

/24 networks wage 1 attack per day for the first 90 days. These time series are not interesting in statistic sense and are eliminated from our study, leading to the updated $|X_{[/8]}(90)| = 190$, $|X_{[/16]}(90)| = 15,242$, and $|X_{[/24]}(90)| = 18,534$ for G-causality analyses.

We further observe that most variances are larger than their respective means. By looking into the individual time series, we observe that among the $|X_{[/8]}(90)| = 190$ time series corresponding to the /8 resolution, 181 time series have a variance that is larger than their respective mean (including 132 with a variance that is at least one order of magnitude larger than the mean); no time series have a variance that equals the respective mean; 9 time series have a variance that is smaller than the respective mean. Among the $|X_{[/16]}(90)| = 15,242$ time series corresponding to the /16 resolution, 9,953 have a variance that is larger than their respective mean (including 233 with a variance that is at least one order of magnitude larger than their respective mean); 14 have a variance that equals their respective mean; and 5,275 have a variance that is smaller than their respective mean. Among the $|X_{[/24]}(90)| = 18,534$ time series corresponding to the /24 resolution, 2,434 have a variance that is larger than their respective mean (including 9 with a variance that is at least one order of magnitude larger than their respective mean); 28 have a variance that equals their respective mean; and 16,099 have a variance that is smaller than their respective mean. The fact that a time series has a variance that is much larger than its mean hints that the time series cannot be modelled by the Poisson process, which means that other kinds of models should be used to fit them.

Insight 4. Most cyber attack rate time series cannot possibly be modeled by the Poisson process because their variance is (much) larger than their respective mean.

4.3. Characterizing G-causality

Guided by the framework, we now analyze G-causality between the time series at a certain resolution and across network resolutions. The first step is to test the stationarity of the cyber attack rate time series. Since we will use the time series in $X(90)$ to start making protection, we test stationarity of these time series rather than those which belong to $X_{[r]}(97)$. Table 3 summarizes the test results on $|X'_{[r]}(90)|$, namely the number of stationary time series at network resolution $r \in \{/8, /16, /24\}$. For example, among the $|X_{[/8]}(90)| = 190$ time series at the /8 network resolution, only $|X'_{[/8]}(90)| = 75$ are stationary. Note that a time series corresponding to a low network resolution (e.g., /8) is stationary does not necessarily mean that the time series corresponding to its sub-networks (i.e., /16) are stationary, and vice versa. In total, 74 of the $X'_{[/8]}(90) = 75$ stationary time series at the /8 network resolution contain at least one stationary time series at the /16 network resolution; 2,360 of the $X'_{[/16]}(90) = 8,007$ stationary time series at the /16 network resolution contain at least one stationary time series at the /24 network resolution.

Table 3. Stationarity and G-causality test results of the time series with at least 30 non-zero observations up to time $T_C = 90$, where $|X_{[r]}(90)|$ is the number of time series at a network resolution $r \in \{/8, /16, /24\}$ that are tested, and $|X'_{[r]}(90)|$ is the number of time series that are stationary, $|X''_{[r]}(90)|$ is the number of time series that have at least one G-causality arc, and $|E_{[r]}(90)|$ is the total number of G-causality arcs.

Net. Res.	$\lvert X_{[/r]}(90)\rvert$	$\lvert X'_{[/r]}(90)\rvert$	$\lvert X''_{[/r]}(90)\rvert$	$\lvert E_{[/r]}(90)\rvert$
$r = /8$	190	75	75	2,452
$r = /16$	15,242	8,007	8,006	27,415,799
$r = /24$	18,534	12,990	12,926	69,832,290

G-causality Analysis at a Single Network Resolution (Answering RQ5). We conduct the G-causality test for all pairs of time series up to time $T_C = 90$. Recall that the G-causality test result for a pair of stationary time series $(X_i(90), X_j(90))$ is a p-value, denoted by $p_{i,j}(90)$. If $p_{i,j}(90) < 0.05$, the null hypothesis that $X_i(90)$ does not G-cause $X_j(90)$ is rejected.

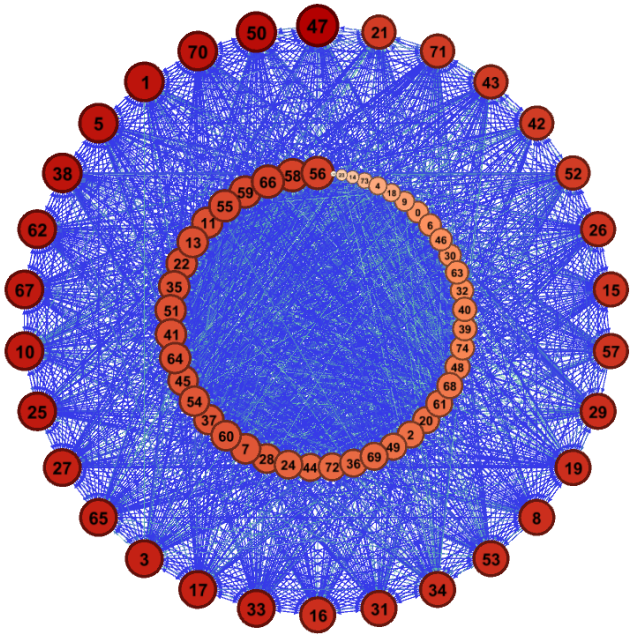


Figure 6. Visualizing G-causality sub-graph $G_{[/8]}(90) = (X_{[/8]}(90), E_{[/8]}(90), W_{[/8]}(90))$ after eliminating the isolated nodes that correspond to non-stationary time series (note: each stationary time series has associated G-causality in this case).

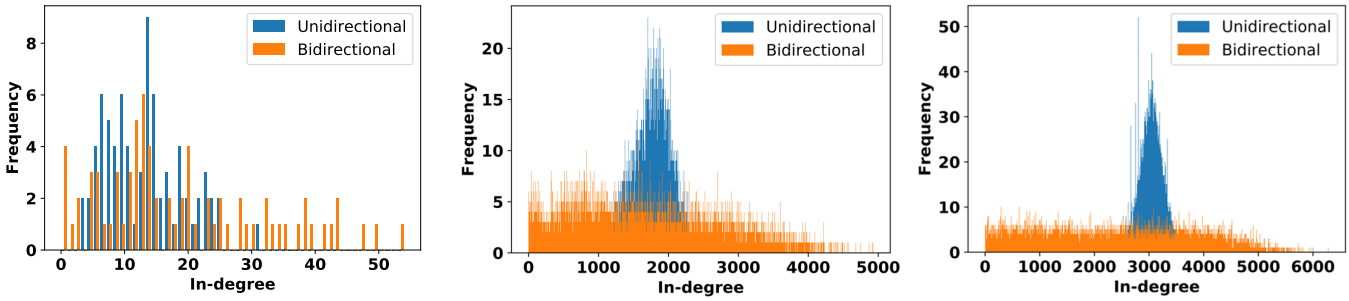
Figure 6 plots the G-causality connections (i.e., arcs) between the stationary time series at the /8 network resolutions for $T_C = 90$, namely the sub-graph of $G_{[/8]}(90) = (X_{[/8]}(90), E_{[/8]}(90), W_{[/8]}(90))$ corresponding to $X''_{[/8]}(90)$, by eliminating the nodes in $X_{[/8]}(90)$ that correspond to non-stationary time series (while noting that each stationary time series has associated G-causality connection in this particular case), where a node's size and color encodes its in-degree (i.e., the larger and darker the node, the larger the in-degree), $E_{[/8]}(90)$ represents the G-causality connections between the nodes, $W_{[/8]}(90)$ is encoded into the color such that a smaller $p_{i,j}(90)$ is indicated as a darker color (albeit the differences are too small to be visually noticeable). The 30 nodes at the outer circle have the highest in-degrees, meaning that they are G-caused by many nodes. Note that we are unable to visualize $G_{[/16]}(90)$ and $G_{[/24]}(90)$ because they have too many nodes.

Insight 5. G-causality is widely exhibited by cyber attack behaviors, hinting that cyber attacks are not random.

Characterizing the Direction of G-causality (Answering RQ6). Recall that unidirectional G-causality means that $(X_i, X_j) \in E$ implies $(X_j, X_i) \notin E$; whereas, bidirectional G-causality relation means that $(X_i, X_j) \in E$ implies $(X_j, X_i) \in E$. Figure 7a plots the density of the nodes' in-degrees at the /8 resolution, by separating unidirectional from bidirectional G-causality. Among the $|X''_{[/8]}| = 75$ nodes associated with G-causality, their in-degrees vary between 1 and 54. This means that some models are not applicable to some nodes (e.g., a node with in-degree 1 can only be applied to GC2VAR, but not GC3VAR because the latter requires 2 helpers). There are $|E_{[/8]}(90)| = 2,452$ pairs of stationary time series at the /8 resolution that have associated G-causality. Among them, 1,011 (41.24%) are unidirectional, and 1,441 (58.76%) are bidirectional. Figure 7b plots the density of the nodes' in-degrees at the /16 resolution. Among the $|X''_{[/16]}| = 8,006$ time series with associated G-causality, their in-degrees vary between 1 and 4,926. There are $|E_{[/16]}(90)| = 27,415,799$ pairs of stationary time series that have associated G-causality. Among them, 14,216,880 (51.86%) are unidirectional and 13,198,919 (48.14%) are bidirectional. Figure 7c plots the density of the in-degrees at the /24 resolution. Among the $|X''_{[/24]}| = 12,926$ time series, their in-degrees vary between 1 and 6,281. There are $|E_{[/24]}(90)| = 69,832,290$ pairs of stationary time series that have associated G-causality. Among them, 39,256,780 (56.22%) are unidirectional and 30,575,510 (43.78%) are bidirectional.

Insight 6. Bidirectional G-causality is widely exhibited at a single network resolution.

G-causality Analysis across Network Resolutions (Answering RQ7). For cross-resolution G-causality analysis, we want to know for example, whether or not a large number of cyber attacks waged from a /8 network are mainly caused by the cyber attacks waged from which of its belonging /16 networks; in order words, we want to know if G-causality can be leveraged to identify the /16 networks that are responsible for the /8 network attack behaviors. As mentioned above, there are 74 /8 networks, 2,360 /16 networks, and 6,017 /24 networks that are applicable for this analysis. To make the discussion succinct, let $|X''_{[/16\in/8]}(90)|$ denote the number of nodes (i.e., time series) at the /8 network resolution that have associated G-causality arc(s) between a /8 network and its belonging /16 networks, and $|E_{[/16\in/8]}(90)|$ denote the total number of such G-causality arcs. We have $|X''_{[/16\in/8]}(90)| = 66$ and $|E_{[/16\in/8]}(90)| = 3,706$. Similarly, we have $|X''_{[/24\in/16]}(90)| = 1,351$ and $|E_{[/24\in/16]}(90)| = 3,783$. For cross-resolution G-causality analysis, we observe that among the $|E_{[/16\in/8]}(90)| = 3,706$ G-causality arcs, 1,011 (27.28%) are unidirectional and 2,695 (72.72%) are bidirectional; among the $|E_{[/24\in/16]}(90)| = 3,783$ G-causality arcs, 1,133 (29.95%) are unidirectional and 2,650 (70.05%) are bidirectional. That is, cyber attack rates at a lower network resolution are strongly indicative of the cyber attack rates corresponding to its belonging higher-resolution sub-network. This



(a) /8 networks **(b)** /16 networks **(c)** /24 networks

Figure 7. Plot of the density (i.e., frequency) of the nodes' in-degrees in the G-causality graph $G_{[r]}(90)$ for $r \in \{/8, /16, /24\}$.

is reasonable because a lower resolution network is composed of a number of higher resolutions networks.

Insight 7. G-causality across network resolution is widely observed, with most being bidirectional.

4.4. Leveraging G-causality (Answering RQ8)

As described in the framework, the key issue is to select an *appropriate* number of *good* helpers at the *proper* network resolution(s). The framework suggests to investigate at least the following 4 factors: direction of G-causality (i.e., unidirectional vs. bidirectional), the number of helpers that are leveraged for prediction, the p-value (i.e., small vs. medium vs. large), and the layers of network resolutions (e.g., one vs. two layers). In order to distinguish predictions leveraging bidirectional vs. unidirectional, G-causality, we use suffix "2" to indicates the former (e.g., GC3VAR2 vs. GC3VAR).

Owing to the limited length of time series, we use rolling forecast, namely using $X_{[r]}(T_C)$ to predict $X_{[r]}(T_C + 1)$, where $90 \leq T_C \leq 96$ and $r \in \{/8, /16, /24\}$. In order to build prediction models for PenVAR and GCPenVAR with $T_C = 90$, we divide the attack data into three parts: 40 observations for model initialization, 50 observations for training, and the rest 7 for forecast evaluation. For fitting the PenVAR model, we need to select the model with the parameter $\hat{\lambda}$ that minimizes the one-step ahead MSPE given by Eq. (7), namely $\text{MSPE}(\lambda) = \frac{1}{50}\sum_{t=41}^{90} \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2$, where $\|\cdot\|_2$ is the L_2 norm. For this purpose, we use the grid search method to identify the optimal $\hat{\lambda}$. At each network resolution r, we set lag $\ell = 2$ to balance model parsimony and fitting performance. This is because a large ℓ will lead to more variables in the model, which will impose a higher computational cost. Moreover, we observe that $\ell = 2$ leads to good prediction accuracy. PenVAR is only conducted at the /8 network resolution because the compute resources available to makes it is infeasible to apply it to the /16 and /24 resolutions, where n is very large or extremely high dimensions (i.e., $|X''_{[/16]}90| = 8,006$ and $|X''_{[/24]}90| = 12,926$). Nevertheless, GCPenVAR does not suffer from this computational problem because it only considers $(m+1)$-variate time series where $m = \deg(X_j, G)$ is the in-degree of node X_j and $m << n$. In what follows we characterize the impact of the four factors.

Factor 1: Direction of G-causality. At each network resolution, we apply the applicable models described in the framework to predict $X_j(T_C + 1)$ for $X_j(T_C) \in X(T_C)$ and $T_C \in [90, 96]$, except that PenVAR is only conducted at the /8 network resolution (owing to the computational feasibility issue mentioned above). In our experiments, we further encounter that many predictions cannot succeed because of the *singularity* problem associated with the GC$(z+1)$VAR models, namely that inverse of a matrix does not exist when its determinant is zero. (It is worth mentioning that it may be technically possible to prune the list of independent variables to alleviate the singularity problem. However, we do not consider this because we aim at fair comparison between the models, which means that the input to these models should be the same. Nevertheless, developing new models to cope with the singularity problem in general is certainly an interesting problem for future research.) As a consequence, predictions are successful only for 31 /8 networks, 1,820 /16 networks, and 2,159 /24 networks. In order to compare the effect of unidirectional vs. bidirectional G-causality, we propose separating the unidirectional G-causality relations from the the bidirectional ones. This also means that we separately sort the p-values associated with the unidirectional G-causality relations and the p-values corresponding to the bidirectional ones.

Table 4 presents the summary statistics of prediction SMAPEs incurred by the same set of models leveraging unidirectional vs. bidirectional G-causality, while recalling that AR is the baseline model and PenVAR is only feasible at the /8 network resolution. Figure 8 further presents the boxplots of the SMAPEs of the models. From Table 4 and Figure 8, we make the following observations.

Table 4. Summary statistics of the SMAPE values of predictions by different models at /8, /16, and /24 resolutions, where bold-font highlights the smallest SMAPE values or the most accurate predictions among all the models at a resolution.

	31 /8 networks with predictions						
Model	Min	Q_1	Q_2	Mean	SD	Q_3	Max
GCPenVAR	0.035	0.176	0.330	0.377	0.284	0.480	1.192
GC($m+1$)VAR	0.065	0.177	0.566	1.042	0.878	2.000	2.000
GC5VAR	0.077	0.142	0.269	0.345	0.284	0.394	1.110
GC4VAR	0.017	0.086	0.230	0.339	0.398	0.422	2.000
GC3VAR	0.070	0.129	0.273	0.329	0.273	0.379	1.111
GC2VAR	0.032	0.126	0.255	0.340	0.286	0.444	1.058
GCPenVAR2	0.056	0.152	0.311	0.358	0.282	0.451	1.117
GC($m+1$)VAR2	1.089	2.000	2.000	1.961	0.170	2.000	2.000
GC5VAR2	**0.012**	**0.056**	**0.083**	**0.165**	**0.210**	**0.157**	**0.924**
GC4VAR2	0.033	0.100	0.192	0.256	0.214	0.336	0.938
GC3VAR2	0.060	0.112	0.262	0.327	0.272	0.420	1.139
GC2VAR2	0.060	0.111	0.219	0.349	0.400	0.359	2.000
PenVAR	0.035	0.177	0.330	0.377	0.284	0.480	1.190
AR	0.029	0.160	0.287	0.340	0.251	0.390	1.030
	1,820 /16 networks with predictions						
Model	Min	Q_1	Q_2	Mean	SD	Q_3	Max
GCPenVAR	0.025	0.410	0.762	0.859	0.512	1.267	2.000
GC($m+1$)VAR	0.045	0.378	0.758	0.860	0.536	1.281	2.000
GC5VAR	0.034	0.381	0.755	0.854	0.533	1.281	2.000
GC4VAR	0.020	0.248	0.680	0.757	0.538	1.200	2.000
GC3VAR	0.034	0.366	0.738	0.845	0.534	1.267	2.000
GC2VAR	0.022	0.373	0.740	0.845	0.533	1.265	2.000
GCPenVAR2	0.006	0.415	0.766	0.858	0.512	1.272	2.000
GC($m+1$)VAR2	0.286	1.143	1.714	1.459	0.566	2.000	2.000
GC5VAR2	0.054	0.857	1.465	1.397	0.584	2.000	2.000
GC4VAR2	0.034	**0.226**	**0.634**	**0.734**	0.539	1.197	2.000
GC3VAR2	0.030	0.344	0.729	0.828	0.537	1.271	2.000
GC2VAR2	**0.005**	0.380	0.769	0.876	0.555	1.339	2.000
AR	0.006	0.345	0.637	0.739	**0.444**	**1.064**	**1.714**
	2,159 /24 networks with predictions						
Model	Min	Q_1	Q_2	Mean	SD	Q_3	Max
GCPenVAR	0.007	0.589	1.104	1.059	0.547	1.530	2.000
GC($m+1$)VAR	0.022	0.571	1.143	1.128	0.581	1.714	2.000
GC5VAR	0.015	0.571	1.121	1.072	0.571	1.558	2.000
GC4VAR	0.006	**0.427**	**0.925**	**0.920**	0.556	1.445	2.000
GC3VAR	0.018	0.602	1.017	1.001	0.468	1.382	2.000
GC2VAR	0.008	0.544	1.068	1.033	0.566	1.527	2.000
GCPenVAR2	0.009	0.586	1.112	1.064	0.552	1.538	2.000
GC($m+1$)VAR2	0.039	0.857	1.143	1.243	0.566	1.714	2.000
GC5VAR2	0.018	0.646	1.143	1.154	0.574	1.714	2.000
GC4VAR2	0.286	2.000	2.000	1.913	**0.262**	2.000	2.000
GC3VAR2	0.286	2.000	2.000	1.906	0.286	2.000	2.000
GC2VAR2	0.004	0.568	1.086	1.048	0.568	1.525	2.000
AR	**0.001**	0.514	0.950	0.923	0.472	**1.313**	**1.714**

First, at the /8 network resolution, all of the bidirectional GC($z+1$)VAR models with $z \in [1, m]$ have at least five statistic values that are lower than their unidirectional counterparts. Consider GC5VAR vs. GC5VAR2 as an example, we observe the following: the minimum SMAPE value of GC5VAR vs. GC5VAR2 G-causality is 0.077 vs. 0.012; the 25% quantile (Q1) of the SMAPE values is 0.142 vs. 0.056; the 50% quantile (Q2 or median) is 0.269 vs. 0.083; the mean is 0.345 vs. 0.165; the standard deviation is 0.284 vs. 0.210; the 75% quantile (Q3) is 0.394 vs. 0.157; and the maximum is 1.110 vs. 0.924. That is, all of the 7 summary statistics show that bidirectional G-causality leads to lower SMAPE values (i.e., higher prediction accuracy) when compared with unidirectional G-causality. This phenomenon is also exhibited by GC3VAR and GC4VAR, but not by GC2VAR, GC($m+1$)VAR, or GCPenVAR. At both /16 and /24 network resolutions, we observe that each GC($z+1$)VAR2 has no or a few statistics that are lower than its GC($z+1$)VAR counterpart and that the GC($z+1$)VAR2 models are not necessarily more accurate than the baseline AR model.

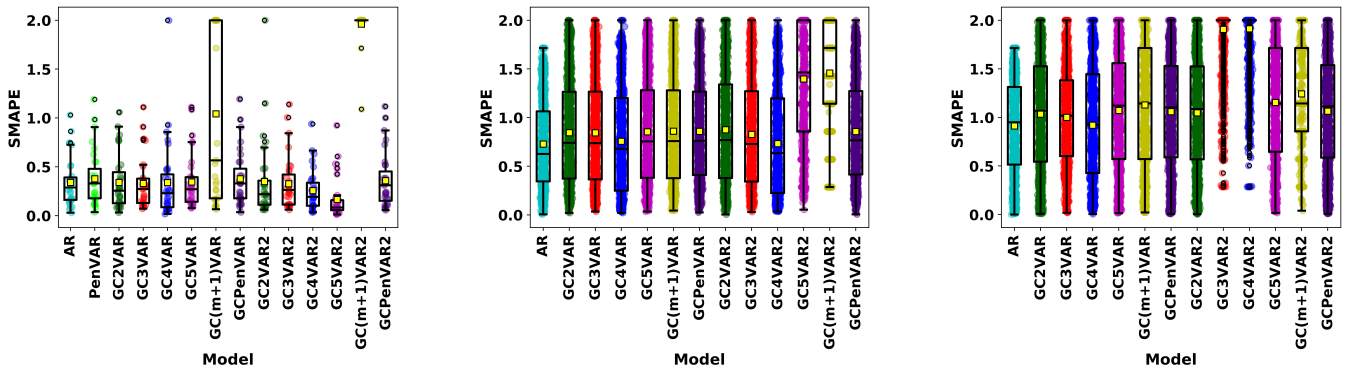
Insight 8. Leveraging bidirectional G-causality at a single network resolution achieves a higher prediction accuracy than leveraging unidirectional G-causality.

Insight 8 may be partly attributed to that the time series corresponding to lower network resolutions are denser (i.e., on average among all of the time series during the $T_C = 90$ days, there are 60.17%, 71.67%, and 97.94% non-zero observations at the /24, /16, and /8 network resolutions, respectively).

Second, AR is indeed less accurate than the models that leverage G-causality at the /8 network resolution, but is surprisingly more accurate for 546 time series at the /16 network resolution and 272 time series at the /24 network resolution. By looking into the data, we notice that these time series have high variations. On the other hand, GCPenVAR is more accurate than AR and the other models that leverage G-causality for 72 time series at the /16 network resolution and 47 time series at the /24 network resolution. By looking into the data, we notice that these time series are low variations and relatively sparse (typically having no more than 50 non-zero observations over the $T_C = 90$ days). This suggests that GCPenVAR may be better at dealing with sparse time series. Nevertheless, GCPenVAR is less accurate than AR and the other models that leverage G-causality for 77 time series at the /16 resolution and 12 time series at the/24 resolution.

Insight 9. G-causality is not always useful in predicting time series with variations, and GCPenVAR may be better at predicting sparse time series, perhaps because it can leverage more information (i.e., all, rather than some, neighbors).

Factor 2: The number of "helpers". From the perspective of whether more "helpers" would lead to more accurate predictions, Table 4 shows that the prediction accuracy increases with the number z of helpers when $z \in [1, 4]$. We observe that all of the 7 summary statistics show that GC5VAR2 has a lower SMAPE value (i.e., higher prediction accuracy) than GC4VAR2, which in turn has a lower SMAPE value than GC3VAR2. While the Min of GC3VAR2 equals that of GC2VAR2, the Mean, SD (Standard Deviation), Max of GC3VAR2 are lower than that of the GC2VAR2's. This suggests that GC3VAR2 offers a higher prediction accuracy than GC2VAR2. That is, GC($z+1$)VAR2 for $z \in [1, 4]$ suggests that more *helpers* would lead to more accurate predictions. However, when $z > 4$ Table 4 shows that the prediction accuracy decreases with z.



(a) /8 networks **(b)** /16 networks **(c)** /24 networks

Figure 8. Box plots of the SMAPE values of different models.

Insight 10. While leveraging G-causality can improve prediction accuracy at a single network resolution, using too many helpers might reduce prediction accuracy because GC5VAR is more accurate than GC($m+1$)VAR when $m > 4$.

Insight 10 suggests that there may be a *threshold*, below which using more *helpers* (i.e., the number of time series that G-causes the target time series) can improve prediction accuracy, but above which the prediction accuracy actually decreases. This may or may not be attributed to over-fitting.

Factor 3: The magnitude of p-values. We propose classifying the $p_{i,j}(T_C)$'s into three categories: small ($p_{i,j}(T_C) < 0.01$); medium ($0.01 \leq p_{i,j}(T_C) < 0.03$); and large ($0.03 \leq p_{i,j}(T_C) < 0.05$). Since leveraging bidirectional G-causality leads to more accurate predictions than leveraging unidirectional G-causality (**Factor 1**), we only consider the former.

Table 5 presents the summary statistics of prediction accuracy of GC($z+1$)VAR2 for $z \in [1, 4]$. Note that we only consider these four models because they contain the three categories of p-values (i.e., small vs. medium vs. large) and therefore permit comparison between them. We observe that all of the 7 statistics show that small p-values correspond to small SMAPE values. However, this phenomenon is not exhibited at the /16 and /24 network resolutions. This discrepancy may be attributed to the matter of data sparsity. To see this, we observe that among the /8 networks, the time series in the small vs. medium vs. large p-value categories on average have 1.19% vs. 1.17% vs. 3.64% observation 0's, respectively. By contrast, among the /16 networks, the time series in small vs. medium vs. large p-value categories on average have 28.35% vs. 27.68% vs. 27.79% observation 0's; among the /24 networks, the time series in the small vs. medium vs. large p-values on average have 40.05% vs. 39.73% vs. 39.71% 0's.

Insight 11. When time series are dense, a smaller p-value in the G-causality test (roughly speaking, stronger G-causality) leads to a higher prediction accuracy.

Factor 4: Layers of network resolutions. In order to see the usefulness of G-causality across network resolutions, we perform predictions by the same set of models, while making a time series corresponding to a low-resolution network the target node and leveraging its sub-networks as *helpers* for prediction purposes. Table 6 presents the summary statistics of the prediction accuracy of leveraging peer networks vs. sub-networks as helpers. At the /8 resolution, we observe that all models, except GCPenVAR, achieve a higher prediction accuracy when leveraging peer networks at the /8 resolution as helpers than leveraging sub-networks at the /16 resolution as helpers. At the /16 resolution, we observe that GCPenVAR, GC3VAR2, GC2VAR, and GC2VAR2 achieve a higher prediction accuracy when leveraging sub-networks at the /24 resolution as helpers than leveraging peer networks at the /16 resolution as helpers.

Insight 12. In terms of leveraging information across network resolutions for prediction, GCPenVAR leveraging G-causality across network resolutions is particularly useful.

5. Discussion

Causality vs. G-causality. Intuitively, causality should be asymmetric, meaning that "A causes B" would imply that "B does not cause A". However, our empirical analysis shows that G-causality does not satisfy this property because bidirectional G-causality relation is widely exhibited at the /8, /16, and /24 network resolutions. These bidirectional G-causality relations highlight a gap between the intuitive notion of causality and the G-causality approach.

Limitations. Our CGC framework has four limitations. (i) We assume away the sparse time series because

Table 5. Summary statistics of the SMAPE values for /8, /16 and /24 networks (#net is the # of networks involved), where bold-font highlights the most accurate predictions among the different p-values with respect to a specific model.

Model	p-value	#net	Min	Q_1	Median	Mean	SD	Q_3	Max
/8 networks									
GC5VAR2	small	71	0.012	**0.064**	**0.099**	**0.209**	**0.238**	**0.276**	**1.188**
	medium	7	0.165	0.304	0.857	0.784	0.538	1.214	1.429
	large	3	**0.104**	0.553	1.003	0.845	0.676	1.216	1.429
GC4VAR2	small	71	**0.001**	**0.109**	**0.202**	**0.300**	**0.315**	**0.368**	2.000
	medium	12	0.145	0.269	0.478	0.628	0.451	0.891	**1.429**
	large	3	0.145	0.580	1.015	0.863	0.655	1.222	**1.429**
GC3VAR2	small	71	**0.001**	**0.111**	**0.222**	**0.315**	**0.292**	**0.452**	**1.358**
	medium	21	0.139	0.245	0.452	0.534	0.387	0.717	1.429
	large	9	0.126	0.389	0.474	0.797	0.584	1.429	1.714
GC2VAR2	small	70	**0.060**	**0.134**	**0.265**	**0.458**	**0.507**	**0.590**	2.000
	medium	53	0.067	0.170	0.383	0.719	0.690	0.898	2.000
	large	32	0.080	0.155	0.658	0.881	0.754	1.610	2.000
/16 networks									
GC5VAR2	small	7807	0.054	0.857	1.430	1.394	**0.563**	2.000	2.000
	medium	7808	0.043	**0.759**	**1.310**	**1.262**	0.598	**1.910**	2.000
	large	7765	**0.021**	0.791	1.340	1.272	0.596	2.000	2.000
GC4VAR2	small	7995	0.025	**0.254**	**0.704**	**0.784**	**0.560**	**1.220**	2.000
	medium	7840	**0.009**	0.637	1.140	1.201	0.602	1.710	2.000
	large	**7803**	0.021	0.645	1.140	1.206	0.602	1.710	2.000
GC3VAR2	small	6816	0.030	**0.376**	**0.781**	**0.869**	**0.555**	**1.302**	2.000
	medium	7897	**0.009**	0.571	1.140	1.129	0.601	1.710	2.000
	large	7849	0.011	0.571	1.140	1.136	0.602	1.710	2.000
GC2VAR2	small	7956	0.027	0.774	**1.180**	**1.174**	**0.516**	**1.590**	2.000
	medium	7903	0.022	0.780	1.200	1.196	0.530	1.640	2.000
	large	7887	**0.011**	**0.770**	**1.180**	1.192	0.532	1.640	2.000
/24 networks									
GC5VAR2	small	12408	**0.006**	**0.857**	**1.290**	**1.243**	0.572	**1.710**	2.000
	medium	5642	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
	large	5412	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
GC4VAR2	small	68	**0.069**	**0.160**	**0.299**	**0.641**	0.704	**0.781**	2.000
	medium	6810	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
	large	6619	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
GC3VAR2	small	10249	**0.202**	2.000	2.000	**1.889**	0.306	2.000	2.000
	medium	7735	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
	large	7578	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
GC2VAR2	small	12652	**0.001**	**0.849**	1.270	1.233	**0.535**	1.710	2.000
	medium	12583	0.005	0.857	1.270	**1.232**	0.539	1.710	2.000
	large	12549	**0.001**	0.857	1.270	1.233	0.542	1.710	2.000

Table 6. Summary statistics of SMAPE values of leveraging peer networks vs. sub-networks as helpers, where ***a*** indicates models leveraging sub-networks (i.e., time series data across multiple network resolutions), and bold-font highlights the more accurate prediction when comparing the prediction leveraging time series across network resolutions (i.e., leveraging sub-networks) and the prediction leveraging time series at a single network resolution (leveraging peer networks).

	Leveraging peer /8 networks vs. /16 sub-networks as helpers									Leveraging peer /16 networks vs. /24 sub-networks as helpers								
Model	net	Min	Q_1	Q_2	Mean	SD	Q_3	Max		Model	net	Min	Q_1	Q_2	Mean	SD	Q_3	Max
GCPenVAR	66	0.035	0.155	0.309	0.389	**0.292**	0.540	**1.192**		GCPenVAR	677	0.028	0.311	0.561	0.721	0.510	1.028	2.000
GCPenVAR ***a***	66	**0.026**	**0.154**	**0.273**	**0.388**	0.301	**0.525**	1.209		GCPenVAR ***a***	677	**0.027**	**0.220**	**0.336**	**0.431**	**0.321**	**0.524**	2.000
GCPenVAR2	67	**0.056**	**0.167**	**0.311**	**0.390**	**0.307**	**0.518**	**1.424**		GCPenVAR2	1350	**0.006**	**0.334**	**0.609**	**0.796**	0.555	**1.185**	2.000
GCPenVAR2 ***a***	67	0.069	0.397	0.589	0.748	0.527	1.009	1.980		GCPenVAR2 ***a***	1350	0.073	0.806	1.210	1.181	**0.476**	1.560	2.000
GCnVAR	54	**0.065**	**0.402**	2.000	1.373	0.829	2.000	2.000		GCnVAR	275	**0.027**	0.280	0.485	**0.702**	**0.544**	**1.051**	2.000
GCnVAR ***a***	54	0.364	0.688	**1.164**	**1.284**	**0.669**	2.000	2.000		GCnVAR ***a***	275	0.039	**0.273**	**0.429**	0.827	0.732	1.714	2.000
GCnVAR2	67	**0.198**	2.000	2.000	**1.914**	0.311	2.000	2.000		GCnVAR2	1292	0.167	1.143	**1.714**	1.557	**0.548**	2.000	2.000
GCnVAR2 ***a***	67	0.391	2.000	2.000	1.946	**0.249**	2.000	2.000		GCnVAR2 ***a***	1292	**0.003**	**0.581**	1.772	**1.373**	0.715	2.000	2.000
GC5VAR	39	**0.077**	**0.155**	**0.284**	**0.396**	**0.307**	**0.558**	**1.110**		GC5VAR	474	**0.027**	0.287	0.486	**0.686**	**0.522**	**1.017**	2.000
GC5VAR ***a***	39	0.110	0.262	1.714	1.211	0.839	2.000	2.000		GC5VAR ***a***	474	0.039	**0.254**	**0.439**	0.831	0.732	1.714	2.000
GC5VAR2	68	**0.012**	**0.064**	**0.096**	**0.207**	**0.238**	**0.274**	**1.188**		GC5VAR2	1302	0.088	1.140	1.710	1.501	**0.579**	2.000	2.000
GC5VAR2 ***a***	68	0.068	0.196	0.401	0.791	0.751	1.571	2.000		GC5VAR2 ***a***	1302	**0.003**	**0.287**	**0.545**	**0.875**	0.713	**1.714**	2.000
GC4VAR	54	**0.017**	**0.134**	**0.254**	**0.562**	**0.647**	**0.641**	2.000		GC4VAR	52	0.040	**0.082**	**0.168**	**0.354**	**0.437**	**0.371**	2.000
GC4VAR ***a***	54	0.077	0.280	2.000	1.279	0.836	2.000	2.000		GC4VAR ***a***	52	0.040	0.143	0.185	0.629	0.801	0.748	2.000
GC4VAR2	63	**0.033**	**0.105**	**0.184**	**0.282**	**0.300**	**0.357**	2.000		GC4VAR2	2	**0.001**	**0.009**	**0.018**	**0.018**	0.025	**0.027**	**0.035**
GC4VAR2 ***a***	63	0.069	0.159	0.298	0.632	0.703	0.748	2.000		GC4VAR2 ***a***	2	2.000	2.000	2.000	2.000	**0.000**	2.000	2.000
GC3VAR	45	**0.061**	**0.136**	**0.219**	**0.357**	**0.290**	**0.491**	**1.111**		GC3VAR	112	**0.024**	0.190	0.340	**0.503**	**0.460**	**0.628**	2.000
GC3VAR ***a***	45	0.078	0.142	0.328	0.854	0.821	2.000	2.000		GC3VAR ***a***	112	0.031	**0.170**	**0.249**	0.736	0.815	2.000	2.000
GC3VAR2	58	**0.060**	0.117	0.227	**0.317**	**0.257**	**0.437**	**1.139**		GC3VAR2	192	0.066	0.189	0.376	0.638	**0.580**	0.924	2.000
GC3VAR2 ***a***	58	0.074	**0.116**	**0.221**	0.464	0.547	0.569	2.000		GC3VAR2 ***a***	192	**0.047**	**0.164**	**0.221**	**0.511**	0.663	**0.364**	2.000
GC2VAR	47	**0.032**	**0.132**	**0.209**	**0.318**	**0.265**	**0.392**	**1.058**		GC2VAR	26	0.102	0.260	0.404	0.830	**0.706**	1.490	2.000
GC2VAR ***a***	47	0.074	0.153	0.288	0.650	0.711	0.812	2.000		GC2VAR ***a***	26	**0.072**	**0.195**	**0.240**	**0.589**	0.712	**0.502**	2.000
GC2VAR2	65	**0.060**	0.129	**0.219**	**0.442**	**0.518**	**0.533**	2.000		GC2VAR2	438	0.024	0.281	0.528	0.795	**0.623**	1.247	2.000
GC2VAR2 ***a***	65	0.072	**0.126**	0.276	0.631	0.733	0.746	2.000		GC2VAR2 ***a***	438	**0.017**	**0.190**	**0.304**	**0.594**	0.665	**0.511**	2.000

the state-of-the-art statistical techniques do not permit sound statistical models of sparse time series. (ii) We assume away the non-stationary time series, which is inherited from the notion of G-causality. Addressing these limitations is an important open problem. Our case study has two limitations that are imposed by the particular dataset. (iii) The dataset only lasts for 97 days. Although it is sufficient to demonstrate the usefulness of the framework, it would be better if we have access to a dataset with a longer period of time. (iv) The dataset is collected by low-interaction honeypots and therefore does not present rich semantic information about the attacks. Using datasets collected from high-interaction honeypots or production networks would resolve this issue, for which the framework is equally applicable.

6. Conclusion

We presented the CGC framework for characterizing the usefulness of G-causality in cybersecurity, especially its usefulness in predicting cyber attack rates. We investigated a number of models and drew a number of insights, which can be leveraged as a stepping-stone towards fully understanding the usefulness and limitations of G-causality in cybersecurity. There are many open problems for future research, including: How can we rigorously, rather than empirically, characterize the usefulness and limitations of G-causality? What are the utilities of other kinds of causality (e.g., Pearl-causality) in the cybersecurity domain? How can we model sparse and non-stationary time series in a principled fashion?

Acknowledgement. We thank the anonymous reviewers for their constructive comments. This work was supported in part by NSF Grant #1736209.

References

[1] (2019), Internet security threat report. URL https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf.

[2] (2019), Personally identifiable information targeted in breaches that impact billions of records. URL https://www.forgerock.com/resources/view/92170441/industry-brief/us-consumer-data-breach-report.pdf.

[3] Zhan, Z., Xu, M. and Xu, S. (2013) Characterizing honeypot-captured cyber attacks: Statistical framework and case study. *IEEE Transactions on Information Forensics and Security* **8**(11): 1775–1789.

[4] Zhan, Z., Xu, M. and Xu, S. (2014) A characterization of cybersecurity posture from network telescope data. In *International Conference on Trusted Systems* (Springer): 105–126.

[5] Zhan, Z., Xu, M. and Xu, S. (2015) Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security* **10**(8): 1666–1677.

[6] Chen, Y.Z., Huang, Z.G., Xu, S. and Lai, Y.C. (2015) Spatiotemporal patterns and predictability of cyberattacks. *PloS one* **10**(5): e0124472.

[7] Peng, C., Xu, M., Xu, S. and Hu, T. (2018) Modeling multivariate cybersecurity risks. *Journal of Applied Statistics* **0**(0): 1–23.

[8] Bakdash, J.Z., Hutchinson, S., Zaroukian, E.G., Marusich, L.R., Thirumuruganathan, S., Sample, C., Hoffman, B. *et al.* (2018) Malware in the future? Forecasting of analyst detection of cyber events. *Journal of Cybersecurity* **4**(1): tyy007.

[9] Werner, G., Yang, S. and McConky, K. (2017) Time series forecasting of cyber attack intensity. In *Proceedings of the 12th Annual Conference on cyber and information security research* (ACM): 18.

[10] Fang, Z., Zhao, P., Xu, M., Xu, S., Hu, T. and Fang, X. Statistical modeling of computer malware propagation dynamics in cyberspace. *Journal of Applied Statistics* .

[11] Pritom, M., Schweitzer, K., Bateman, R., Xu, M. and Xu, S. (2020) Data-driven characterization and detection of covid-19 themed malicious websites. In *IEEE ISI'2020*.

[12] Pritom, M., Schweitzer, K., Bateman, R., Xu, M. and Xu, S. (2020) Characterizing the landscape of covid-19 themed cyberattacks and defenses. In *IEEE ISI'2020*.

[13] Ficke, E. and Xu, S. (2020) Apin: Automatic attack path identification in computer networks. In *IEEE ISI'2020*.

[14] Li, D., Li, Q., Ye, Y. and Xu, S. (2020) Sok: Arms race in adversarial malware detection. *CoRR* **abs/2005.11671**.

[15] Xu, M., Hua, L. and Xu, S. (2017) A vine copula model for predicting the effectiveness of cyber defense early-warning. *Technometrics* **59**(4): 508–520.

[16] Fang, Z., Xu, M., Xu, S. and Hu, T. (2021) A framework for predicting data breach risk: Leveraging dependence to cope with sparsity. *IEEE Trans. Inf. Forensics Secur.* **16**: 2186–2201.

[17] Liu, Z., Zheng, R., Lu, W. and Xu, S. (2021) Using event-based method to estimate cybersecurity equilibrium. *IEEE CAA J. Autom. Sinica* **8**(2): 455–467.

[18] Granger, C.W. (1969) Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society* : 424–438.

[19] Xu, L., Zhan, Z., Xu, S. and Ye, K. (2014) An evasion and counter-evasion study in malicious websites detection. In *IEEE CNS*: 265–273.

[20] Li, X., Parker, P. and Xu, S. (2011) A stochastic model for quantitative security analyses of networked systems. *IEEE Transactions on Dependable and Secure Computing* **8**(1): 28–43.

[21] Xu, S., Li, X., Parker, T. and Wang, X. (2011) Exploiting trust-based social networks for distributed protection of sensitive data. *IEEE T-IFS* **6**(1): 39–52.

[22] Xu, L., Zhan, Z., Xu, S. and Ye, K. (2013) Cross-layer detection of malicious websites. In *Third ACM Conference on Data and Application Security and Privacy (CODASPY'13)*: 141–152.

[23] Li, Z., Zou, D., Xu, S., Ou, X., Jin, H., Wang, S., Deng, Z. *et al.* (2018) Vuldeepecker: A deep learning-based system for vulnerability detection. In *Proc. NDSS'18*.

[24] Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., Zhang, Z., Chen, Z. *et al.* (2021), Vuldeelocator: A deep learning-based system for detecting and locating software

vulnerabilities, IEEE TDSC (accepted for publication).
[25] Zou, D., Wang, S., Xu, S., Li, Z. and Jin, H. (2019) μvuldeepecker: A deep learning-based system for multiclass vulnerability detection. *IEEE Transactions on Dependable and Secure Computing* : 1–1doi:10.1109/TDSC.2019.2942930.
[26] Li, Z., Zou, D., Xu, S., Jin, H., Qi, H. and Hu, J. (2016) Vulpecker: an automated vulnerability detection system based on code similarity analysis. In *Pro. ACSAC'16*: 201–213.
[27] Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., Chen, Z., Wang, S. *et al.* (2021) Sysevr: A framework for using deep learning to detect software vulnerabilities. *IEEE Transactions on Dependable and Secure Computing (accepted for publication)* .
[28] Xu, S., Yung, M. and Wang, J. (2021/04/28) Seeking foundations for the science of cyber security. *Information Systems Frontiers* doi:10.1007/s10796-021-10134-8.
[29] Xu, S. (2014) Cybersecurity dynamics. In *Proc. HotSoS'14*: 14:1–14:2.
[30] Xu, S. (2019) Cybersecurity dynamics: A foundation for the science of cybersecurity. In *Proactive and Dynamic Network Defense*, 1–31.
[31] Xu, S. (2020) The cybersecurity dynamics way of thinking and landscape (invited paper). In *ACM Workshop on Moving Target Defense*.
[32] Zheng, R., Lu, W. and Xu, S. (2018) Preventive and reactive cyber defense dynamics is globally stable. *IEEE TNSE* **5**(2): 156–170.
[33] Lin, Z., Lu, W. and Xu, S. (2019) Unified preventive and reactive cyber defense dynamics is still globally convergent. *IEEE/ACM Trans. Netw.* **27**(3): 1098–1111.
[34] Han, Y., Lu, W. and Xu, S. (2020) Preventive and reactive cyber defense dynamics with ergodic time-dependent parameters is globally attractive. *CoRR* **abs/2001.07958**.
[35] Xu, S., Lu, W. and Zhan, Z. (2012) A stochastic model of multivirus dynamics. *IEEE Transactions on Dependable and Secure Computing* **9**(1): 30–45.
[36] Xu, S., Lu, W. and Xu, L. (2012) Push- and pull-based epidemic spreading in networks: Thresholds and deeper insights. *ACM TAAS* **7**(3).
[37] Xu, S., Lu, W., Xu, L. and Zhan, Z. (2014) Adaptive epidemic dynamics in networks: Thresholds and control. *ACM TAAS* **8**(4).
[38] Zheng, R., Lu, W. and Xu, S. (2015) Active cyber defense dynamics exhibiting rich phenomena. In *Proc. HotSoS*.
[39] Xu, M., Schweitzer, K.M., Bateman, R.M. and Xu, S. (2018) Modeling and predicting cyber hacking breaches. *IEEE T-IFS* **13**(11): 2856–2871.
[40] Kar, M., Nazlioğlu, Ş. and Ağir, H. (2011) Financial development and economic growth nexus in the mena countries: Bootstrap panel granger causality analysis. *Economic modelling* **28**(1-2): 685–693.
[41] Shojaie, A. and Michailidis, G. (2010) Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics* **26**(18): i517–i523.
[42] Dewan, S. and Ramaprasad, J. (2014) Social media, traditional media, and music sales. *Mis Quarterly* **38**(1).
[43] Tilghman, P. and Rosenbluth, D. (2013) Inferring wireless communications links and network topology from externals using granger causality. In *MILCOM 2013-2013 IEEE Military Communications Conference* (IEEE): 1284–1289.
[44] Deka, R.K., Bhattacharyya, D.K. and Kalita, J.K. (2019) Granger causality in tcp flooding attack. *IJ Network Security* **21**(1): 30–39.
[45] Qin, X. and Lee, W. (2004) Attack plan recognition and prediction using causal networks. In *20th Annual Computer Security Applications Conference* (IEEE): 370–379.
[46] Zaiontz, C. (2013) Real statistics using excel. cronbach's alpha. *Retrieved January* **21**.
[47] Granger, C.W. (1988) Some recent development in a concept of causality. *Journal of econometrics* **39**(1-2): 199–211.
[48] Hiemstra, C. and Jones, J.D. (1994) Testing for linear and nonlinear granger causality in the stock price-volume relation. *The Journal of Finance* **49**(5): 1639–1664.
[49] Asimakopoulos, I., Ayling, D. and Mahmood, W.M. (2000) Non-linear granger causality in the currency futures returns. *Economics Letters* **68**(1): 25–30.
[50] Brovelli, A., Ding, M., Ledberg, A., Chen, Y., Nakamura, R. and Bressler, S.L. (2004) Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by granger causality. *Proceedings of the National Academy of Sciences* **101**(26): 9849–9854.
[51] Yamanishi, K. and Takeuchi, J.i. (2002) A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*: 676–681.
[52] Rothe, C. and Sibbertsen, P. (2006) Phillips-perron-type unit root tests in the nonlinear estar framework. *Allgemeines Statistisches Archiv* **90**(3): 439–456.
[53] Cheung, Y.W. and Lai, K.S. (1998) Power of the augmented dickey-fuller test with information-based lag selection. *Journal of Statistical Computation and Simulation* **60**(1): 57–65.
[54] Lütkepohl, H. (2005) *New introduction to multiple time series analysis* (Springer Science & Business Media).
[55] Liddle, A.R. (2007) Information criteria for astrophysical model selection. *Monthly Notices of the Royal Astronomical Society: Letters* **377**(1): L74–L78.
[56] Akaike, H. (1974) A new look at the statistical model identification. *IEEE transactions on automatic control* **19**(6): 716–723.
[57] Lomax, R.G. (2007) *Statistical concepts: A second course* (Lawrence Erlbaum Associates Publishers).
[58] Chen, Z. and Yang, Y. (2004) Assessing forecast accuracy measures. *Preprint Series* **2010**: 2004–10.
[59] Alata, E., Dacier, M., Deswarte, Y., Kaaâniche, M., Kortchinsky, K., Nicomette, V., Pham, V.H. *et al.* (2006) Collection and analysis of attack data based on honeypots deployed on the internet. In *Quality of Protection* (Springer), 79–91.
[60] Almotairi, S., Clark, A., Mohay, G. and Zimmermann, J. (2008) Characterization of attackers' activities in honeypot traffic using principal component analysis. In *2008 IFIP International Conference on Network and Parallel Computing* (IEEE): 147–154.

[61] Nicholson, W.B., Matteson, D.S. and Bien, J. (2017) Varx-l: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting* **33**(3): 627–651.

[62] Gao, Y., Li, Z. and Chen, Y. (2006) A dos resilient flow-level intrusion detection approach for high-speed networks. In *ICDCS*, **6**: 39.

[63] Dagon, D., Qin, X., Gu, G., Lee, W., Grizzard, J., Levine, J. and Owen, H. (2004) Honeystat: Local worm detection using honeypots. In *International Workshop on Recent Advances in Intrusion Detection* (Springer): 39–58.

[64] Pham, V.H. and Dacier, M. (2011) Honeypot trace forensics: The observation viewpoint matters. *Future Generation Computer Systems* **27**(5): 539–546.

[65] Antonatos, S., Polakis, I., Petsas, T. and Markatos, E.P. (2010) A systematic characterization of im threats using honeypots. In *ISOC Network and Distributed System Security Symposium (NDSS)*.

[66] Kreibich, C. and Crowcroft, J. (2004) Honeycomb: creating intrusion detection signatures using honeypots. *ACM SIGCOMM computer communication review* **34**(1): 51–56.

[67] Portokalidis, G. and Bos, H. (2007) Sweetbait: Zero-hour worm detection and containment using low-and high-interaction honeypots. *Computer Networks* **51**(5): 1256–1274.

[68] Anagnostakis, K.G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E. and Keromytis, A.D. (2005) Detecting targeted attacks using shadow honeypots .

[69] Peng, C., Xu, M., Xu, S. and Hu, T. (2017) Modeling and predicting extreme cyber attack rates via marked point processes. *Journal of Applied Statistics* **44**(14): 2534–2563.

[70] Provos, N. (2004) A virtual honeypot framework. In *Proc. USENIX Security Symposium*.

[71] Balas, E. and Viecco, C.H. (2005) Towards a third generation data capture architecture for honeynets. *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop* : 21–28.

[72] Inacio, C. and Trammell, B. (2010) Yaf: Yet another flowmeter. In *LISA*.