STATISTICS IN TRANSITION new series, March 2024
Vol. 25, No. 1, pp. 1–22, https://doi.org/10.59170/stattrans-2024-001
Received – 11.09.2023; accepted – 05.12.2023

Statistical risk quantification of two-directional internet traffic flows

Piotr Kokoszka[1], **Mengting Lin**[2], **Haonan Wang**[3], **Stephen Hayne**[4]

Abstract

We develop statistical methodology for the quantification of risk of source-destination pairs in an internet network. The methodology is developed within the framework of functional data analysis and copula modeling. It is summarized in the form of computational algorithms that use bidirectional source-destination packet counts as input. The usefulness of our approach is evaluated by an application to real internet traffic flows and via a simulation study.

Key words: Copula, Functional data, Internet traffic, Principal components, Risk quantification.

1. Introduction

Malicious cyberattacks have emerged as a growing threat to economic performance and national security. They can be launched by criminal organizations or autocratic governments. A significant challenge facing the internet security community is to develop algorithms that can automatically detect abnormal network access patterns. Attackers use many different techniques, such as distributed denial of service attacks (DDoS), intrusions that lead to the installation of malware for exfiltration or ransomware intrusion, misconfigured servers for reflection and amplification attacks. By sending a misconfigured server request using a spoofed IP address, the server will unknowingly bombard the target with a frequency 50 or more times higher than that of the response. Attacks of various types have been subjects of extensive research, with thousands papers on the above topics. Some representative recent contributions are Dong and Sarem (2019), Nishanth and Mujeeb (2020), Sambangi and Gondi (2020) and Awan *et al.* (2021).

In this paper, we propose statistical methodology aimed at detecting attacks manifested as unusual traffic between a source and a destination IP addresses. Our focus is on identifying such pairs and ranking them according to the threat they may pose. Related papers, focusing on outlier detection in multivariate functional data, are Dai and Genton (2018) and

[1]Department of Statistics, Colorado State University, Fort Collins CO 80523, USA.
E-mail: Piotr.Kokoszka@colostate.edu. ORCID: https://orcid.org/0000-0001-9979-6536.
[2]Department of Statistics, Colorado State University, Fort Collins CO 80523, USA.
ORCID: https://orcid.org/0009-0002-7712-9585.
[3]Department of Statistics, Colorado State University, Fort Collins CO 80523, USA.
ORCID: https://orcid.org/0000-0002-8892-6232.
[4]Department of Computer Information Systems, Colorado State University, Fort Collins CO 80523, USA.
ORCID: https://orcid.org/0000-0002-9578-3364.

Amovin-Assagba *et al.* (2022). Dai and Genton (2018) propose graphical tools for identifying the set of potentially outlying curves by taking into account unusually large magnitudes and/or shapes. They do not rank the pairs, even though this might be possible by elaborating on their approach. Amovin-Assagba *et al.* (2022) also focus on identifying the set of outlying pairs, but do not rank them in any way. They postulate a specific model motivated by the industrial application they consider. Such a model, and the clustering technique they use, need not be suitable for the data we consider. Basically, related existing approaches focus on identifying the set of outliers rather than assigning numerical measures of separation from most curves.

Our method is based on multivariate functional principal components and copula modeling. Internet streaming data are recorded at densely spaced time points, so they can be modeled as densely observed functions. This suggests that functional data analysis (FDA) approaches might be suitable. Following the monographs of Bosq (2000), Ramsay and Silverman (2005) and Ferraty and Vieu (2006), FDA has grown into a mature field of statistics. Its advantage over competing approaches is that all information in the time series of traffic traces, e.g. shape, variation, and timing, can be taken into account. Functional principal component analysis (FPCA) is a statistical method used to uncover main patterns in functional data, see e.g. Chapter 11 of Kokoszka and Reimherr (2017). FPCA is a powerful dimension reduction, or feature extraction, tool when a sample of functions from a single population is observed. In our setting, we are dealing with bidirectional traffic flows, so we need an analog of FPCA for samples whose elements are pairs of functions. A suitable tool is therefore Multivariate (bivariate in our case) FPCA. Such methods have recently been studied by Happ and Greven (2018), Górecki *et al.* (2018), Krzyśko and Smaga (2020, 2021), even though earlier related work exists, e.g. Berrendero *et al.* (2011), Jacques and Preda (2014), Chiou *et al.* (2014).

A copula describes the joint distribution of random vectors with standard uniform marginal distributions. Many excellent monographs are available, e.g. Nelsen (2006), Joe (2015), Hofert *et al.* (2018) and Czado (2019). A copula model decomposes a multivariate distribution function into two elements: the marginal distributions and the copula which captures the dependence relationship of the marginals. In recent years, copulas have been used to handle multivariate cybersecurity risks, e.g. Peng *et al.* (2018), and for predicting the effectiveness of cyber defense early-warning, e.g. Xu *et al.* (2017). Both FPCA and copula modeling show flexibility and efficiency that we also demonstrate for our methodology that combines and suitably refines them for our task.

To summarize our contribution, this paper develops statistical methodology to identify IP addresses of source-destination pairs that exhibit unusual and suspicious behavior and quantify their cybersecurity risks. We use the term risk to refer to the level of extreme behavior relative to the bulk of the data. We treat the bi-directional internet flows as bivariate functional data and compute scores using a multivariate FPCA (MFPCA) algorithm. The scores provide low dimensional representations of the traffic between the node IP addresses of each pair. Then, we propose a multivariate copula to compute the cybersecurity risk. The copula model is estimated after outlying scores have been removed because it is used to compute probabilities of extreme observations under the assumption of normal traffic. Even though we deal with a specific application, we propose a general paradigm that can be used

to develop effective screening tools to detect unusual multiple functional data objects.

It is informative to put the approach we propose into the context of previous research. Methods for detecting internet anomalies can be divided into signature-based methods and profile-based methods, Liao *et al.* (2013). Several requirements are necessary for signature-based methods to identify suspects, including the need for labeled data, prior results from anomalies, and an external supervisor. However, using this method, it is not possible to detect new intrusions that are unknown, Modi *et al.* (2013). A number of approaches have been proposed for the detection and prevention of DDoS attacks by using classification algorithms. The majority of such techniques require pre-training on a set of labeled data before they are applied. There are several popular approaches to data analysis, including Support Vector Machines, Bayesian Networks, and Neural Networks, Ahmed *et al.* (2016). Although these algorithms have performed well in certain situations in which "known" anomaly data exist, they can be difficult to incorporate into a larger set of algorithms due to the reliance on labeled data. It is likely that there will be no real knowledge for the classification of network traffic, which means supervised techniques can only be applied when approximated labels are available. It is inevitable that the results of training will be skewed by incorrectly labeled data, Soysal and Schmidt (2010).

Furthermore, an analysis of frequency domains has proven to be effective in detecting DDoS attacks, Fouladi *et al.* (2016). Compared to normal traffic in which energy is distributed among different frequencies, most DDoS attack energy is found at lower frequencies. Such methods have been used to discover abnormalities and analyze traffic patterns, Fouladi *et al.* (2013). Low rate DoS attacks (LDoS) are distinguished from normal traffic using spectrum energy and thresholding methods. Spectrum energy and thresholding are used to separate them, Wu *et al.* (2015). Spectral analysis is one of the methods used by the authors in order to detect DoS attacks, Hussain *et al.* (2003). It should be noted that most studies of frequency domain analysis in identifying DoS and DDoS attacks are carried out in simulation environments.

The remainder of this paper is structured as follows. Section 2 begins with an introduction of the MFPCA followed by algorithms for identification of outliers and copula based risk quantification. In Section 3, we apply our methods to a DDoS data set. The analysis is supplemented by a simulation study in Section 4.

2. Statistical methodology

In Section 2.1, we review the MFPCA and interpret it in context of source-destination traffic flows. Section 2.2 describes strategies used to remove outlying pairs so that a model for normal traffic (for the whole source-destination network) can be constructed. Finally, Section 2.3 explain the estimation of this model.

2.1. Multivariate functional principal components

To make the exposition more relevant, we introduce multivariate functional principal component analysis (MFPCA) in the context of time series of packet counts.

Suppose there are N SIP-DIP (source-destination) pairs. Sources are outside, and destinations are inside an organization or a protected network. Let $(X_i(t), Y_i(t))$ be a bivariate

time series associated with the ith SIP-DIP pair. Here, $X_i(t)$ denotes the count of packets in hour t in the SIP $\to$ DIP direction (i.e. inbound), and $Y_i(t)$ denotes the count of packets in hour t in the DIP $\to$ SIP direction (i.e. outbound). These are pairs of noisy functions over the time interval $[0,T]$. We create their smooth versions and set

$$\mathbf{h}_i(t) = [h_i^{(1)}(t), h_i^{(2)}(t)]^\top. \tag{2.1}$$

The smoothing serves two purposes: 1) it is the first step in dimension reduction because it eliminates noise, 2) within an FDA software it converts discrete data to functional objects. The latter can be done in such a way that the functional objects look almost exactly as raw data, but then no noise reduction is achieved. We performed the smoothing using 100 B-spline basis functions. In the context of the data studied in Section 3, it corresponds to approximately using averages over 2.5 h, thus focusing only on persistent anomalies or attacks. Using 250 basis functions would practically correspond to working with raw data and would thus include anomalies lasting an hour or less, which we want to exclude, unless they are so large that their influence spreads over a few hours. Using 50 basis functions would focus on anomalies impacting at least five hours. The latter choice produces basically the same risk rankings as the 100 basis functions we use in the remainder of the paper. The details of smoothing are not essential to understand the remainder of the paper, we refer e.g. to Chapter 1 of Kokoszka and Reimherr (2017). Examples of the raw count data and smooth series are shown in Figure 1.

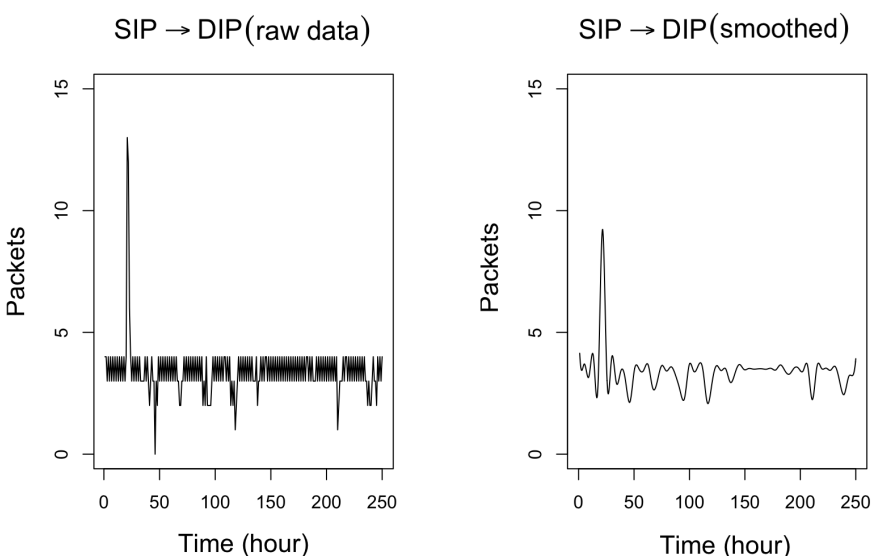

Figure 1: Example of DIP→SIP traffic and its smooth version.

We begin by describing the MFPCA algorithm of Happ and Greven (2018). We initially assume that each pair i comes from the same population, in particular, the functions $h_1^{(k)}, \ldots, h_N^{(k)}$, have the same distributions as a population function $h^{(k)}$. This corresponds to

the absence of any outliers. We set

$$\boldsymbol{\mu}(t) = [\mu^{(1)}(t), \mu^{(2)}(t)]^\top = [E[h_i^{(1)}(t)], E[h_i^{(2)}(t)]]^\top, \qquad 1 \le i \le N, \tag{2.2}$$

and consider the Karhunen–Loève expansions

$$h_i^{(k)}(t) - \mu^{(k)}(t) = \sum_{m=1}^{\infty} \xi_{i,m}^{(k)} \phi_m^{(k)}(t) \approx \sum_{m=1}^{M} \xi_{i,m}^{(k)} \phi_m^{(k)}(t), \quad k = 1,2. \tag{2.3}$$

The functions $\phi_m^{(k)}$ are the functional principal components of the functions $h_i^{(k)}$. Their scores are $\xi_{i,m}^{(k)} = \langle h_i^{(k)} - \mu^{(k)}, \phi_m^{(k)} \rangle$. At this stage, decomposition (2.3) is performed for each k separately. For each $k = 1,2$, the functions $\phi_m^{(k)}$ are orthonormal in the Hilbert space $L^2([0,T])$ and provide optimal data-driven basis systems in the sense that a specified accuracy of approximation that can be achieved with the smallest possible truncation level M. We refer e.g. to Chapter 11 of Kokoszka and Reimherr (2017) for an introductory account of FPCA and to Ramsay and Silverman (2005) and Horváth and Kokoszka (2012) for many examples of applications of FPCA.

Based on the sample $h_i^{(k)}$, $1 \le i \le N$, we can estimate the FPCs $\phi_m^{(k)}$ and the scores $\xi_{i,m}^{(k)}$. We denote the corresponding estimators by $\hat{\phi}_m^{(k)}$ and $\hat{\xi}_{i,m}^{(k)}$. We set

$$\Xi_i = (\hat{\xi}_{i,1}^{(1)}, ..., \hat{\xi}_{i,M}^{(1)}, \hat{\xi}_{i,1}^{(2)}, \ldots, \hat{\xi}_{i,M}^{(2)}) \tag{2.4}$$

and denote by $\boldsymbol{\Xi}$ the $N \times 2M$ matrix whose ith row is Ξ_i. Next, we set

$$\widehat{\mathbf{Z}} = (N-1)^{-1} \boldsymbol{\Xi}^\top \boldsymbol{\Xi} \quad (\dim[\widehat{\mathbf{Z}}] = 2M \times 2M). \tag{2.5}$$

The entries of the matrix $\widehat{\mathbf{Z}}$ are estimators of the covariances $E[\xi_m^{(k)} \xi_{m'}^{(l)}]$, $k,l = 1,2$, $m,m' = 1,\ldots M$.

The eigenvalues of the positive definite matrix $\widehat{\mathbf{Z}}$ are denoted by λ_s and the orthonormal vectors belonging to them by $\hat{\mathbf{c}}_s$, i.e.

$$\widehat{\mathbf{Z}} \hat{\mathbf{c}}_s = \lambda_s \hat{\mathbf{c}}_s, \quad s = 1, \ldots, 2M, \tag{2.6}$$

with the convention that the eigenvalues λ_s are ordered from the largest to the smallest. Each $\hat{\mathbf{c}}_s$ is a column vector of length $2M$. The multivariate eigenfunctions are estimated by $\hat{\psi}_m^{(k)}$ where

$$\hat{\psi}_m^{(k)}(t) = \sum_{j=1}^{M} \hat{c}_{(k-1)M+j,m} \hat{\phi}_j^{(k)}(t), \quad m = 1,2,\ldots,M, \;\; k = 1,2. \tag{2.7}$$

The multivariate scores are calculated as

$$\hat{\rho}_{i,m} = \sum_{k=1}^{2} \sum_{j=1}^{M} \hat{c}_{(k-1)M+j,m} \hat{\xi}_{i,j}^{(k)}, \quad m = 1,2,\ldots,M, \;\; i = 1,...,n. \tag{2.8}$$

There is a correlation between the two sets of scores since the number of packets sent from SIP to DIP is correlated with the number of packets sent from DIP to SIP. The MFPCA algorithm has the advantage of revealing a joint variation in the number of packets sent in both directions that cannot be captured by separate FPCA.

We emphasize that the $\phi_m^{(k)}$ and $\xi_{i,m}^{(k)}$ are the functional principal components and scores from univariate FPCA, while the $\hat{\psi}_m^{(k)}$ are the multivariate functional principal components of the kth variable and $\hat{\rho}_{i,m}$ are the corresponding scores of the ith multivariate functional observation. Thus, in the MFPCA, the functional principal components of both variables share the same score. These scores reflect the variability of pairs rather than their individual components. While the objects at the population level are defined under the assumption of identical distributions, the estimators discussed above can be computed for any sample of SIP-DIP pairs.

We conclude this section by introducing the concept of the copula, see Genest and Nešlehová (2012) for a recent review. Consider a random vector $(Z_1,\ldots,Z_d)$ with univariate continuous marginal distribution $F_1,\ldots,F_d$, respectively. Then the random vector $(U_1,\ldots,U_d) = (F_1(Z_1),\ldots,F_d(Z_d))$, where $F_k(z) = P(Z_k \leq z)$ has marginals that are uniformly distributed on the interval $[0,1]$. The copula of $(Z_1,\ldots,Z_d)$ is defined as the joint cumulative distribution functions of $(U_1,\ldots,U_d)$, i.e.

$$C(u_1,\ldots,u_d) = P\left(Z_1 \leq F_1^{-1}(u_1),\ldots,Z_d \leq F_d^{-1}(u_d)\right). \tag{2.9}$$

Equivalently, for any random vector $(Z_1,\ldots,Z_d)$ with distribution function $F(z_1,\ldots,z_d)$ and marginal distributions $F_1,\ldots,F_d$, there is a copula C such that

$$F(z_1,\ldots,z_d) = C(F_1(z_1),\ldots,F_d(z_d)).$$

Therefore, assuming that the margins $F_1,\ldots,F_d$ are continuous and that the unique underlying copula is absolutely continuous, the joint density function can be represented as

$$f(z_1,...,z_d) = c(F_1(z_1),\ldots,F_d(z_d))\prod_{i=1}^{d} f_i(z_i),$$

where $f_i(z_i)$ is the corresponding marginal density function of Z_i and $c(u_1,...,u_d)$ is the $d-$dimensional copula density function. We refer to C or c as a copula model.

In Sections 2.2 and 2.3, we use the letter d in place of M. Our recommendation is to perform the MFPCA for some larger M, and then depending on the variance explained, use $d < M$ initial components.

2.2. Identification of risky source-destination pairs

We will use bivariate FPCA and a probabilistic copula-based method as our anomaly detection and risk quantification techniques. There are three stages. First, we consider the bi-directional streams $[(h_i^{(1)}(t), h_i^{(2)}(t))]$, $i = 1,...,N$, as bivariate functional data and

compute the scores $\hat{\rho}_{i,m}$ defined by (2.8). Then, a copula model is estimated based on the score vectors $\boldsymbol{\rho}_i = (\rho_{i1}, ..., \rho_{id})$, $i = 1, ..., N$, obtained from the bivariate FPCA after outlying scores or outlying functions have been removed. Finally, a copula model is used to compute the risk of each SIP-DIP pair. We propose two strategies to remove outliers. In the first algorithm, we remove extremely large scores before fitting a copula. In the second algorithm, we remove pairs of functions associated with extreme scores, recompute the scores, and then fit a copula. The justification for removing outlying pairs of curves is to ensure that a copula is estimated on data that can be reasonably assumed to come from the same distribution, so a single copula model is appropriate. Outliers come from different distributions than the bulk of the data. These two strategies are summarized in Algorithms 1 and 2 below. In Algorithm 3, we explain how extremely large scores are identified.

ALGORITHM 1

1. For the smooth versions $(h_i^{(1)}(t), h_i^{(2)}(t))$, $i = 1, ..., N$, estimate the multivariate functional principal components $\psi_m^{(k)}$, $k = 1, 2$, and the scores $\hat{\rho}_{i,m}$, $m = 1, ..., d$.
2. If pair i has extremely large $\hat{\boldsymbol{\rho}}_i$, then it is considered as an outlier. Remove $\hat{\boldsymbol{\rho}}_i$ from the estimated scores.
3. Estimate a copula model based on the remaining scores $\hat{\boldsymbol{\rho}}_i = (\hat{\rho}_{i1}, ..., \hat{\rho}_{id})$.

ALGORITHM 2

1. Step 1 is the same as in Algorithm 1.
2. If pair i has extremely large $\hat{\boldsymbol{\rho}}_i$, remove $(h_i^{(1)}(t), h_i^{(2)}(t))$.
3. Estimate the multivariate functional principal components $\psi_m^{(k)}$ and the scores $\hat{\rho}_{i,m}$ again.
4. Iterate Step 2 and Step 3 until there is no more $\hat{\boldsymbol{\rho}}_i$ identified as outlying.
5. Estimate a copula model based on estimated scores $\hat{\boldsymbol{\rho}}_i = (\hat{\rho}_{i1}, ..., \hat{\rho}_{id})$.

Step 2 of both algorithms identifies outlying pairs i using the following Algorithm 3 due to Billor *et al.* (2000). We note that any effective way of identifying pairs of outlying curves could be used; the approaches of Hubert *et al.* (2005), Dai and Genton (2018) or Amovin-Assagba *et al.* (2022) could be effective. As we will see in Section 3, a more significant difference arises depending on whether Algorithms 1 or 2 are used.

ALGORITHM 3

1. Compute the Mahalanobis distance for each $\hat{\boldsymbol{\rho}}_i$:

$$\text{Mahalanobis distance} = (\hat{\boldsymbol{\rho}}_i - \bar{\hat{\boldsymbol{\rho}}}_i)^\top \mathbf{S}^{-1} (\hat{\boldsymbol{\rho}}_i - \bar{\hat{\boldsymbol{\rho}}}_i),\ i = 1, \ldots, N,$$

   where $\bar{\hat{\boldsymbol{\rho}}}_i$ and $\mathbf{S}$ are the mean and the sample covariance matrix of the $\hat{\boldsymbol{\rho}}_1, \ldots, \hat{\boldsymbol{\rho}}_N$. Select a potential basic subset of size k ($k > M$) of smallest Mahalanobis distances that can safely be assumed free of outliers.

2. Compute the discrepancies:

$$d_i = \sqrt{(\hat{\boldsymbol{\rho}}_i - \bar{\boldsymbol{\rho}}_b)^\top \mathbf{S}_b^{-1} (\hat{\boldsymbol{\rho}}_i - \bar{\boldsymbol{\rho}}_b)},\ i = 1, \ldots, N,$$

where $\bar{\boldsymbol{\rho}}_b$ and $\mathbf{S}_b$ are the sample mean and the sample covariance matrix of the observations in the basic subset.

3. Denote by $\chi^2_{d,\alpha/N}$ the $(1-\alpha/N)$th quantile of the chi-square distribution with d degrees of freedom. The level α depends on how many risky pairs we want to identify; see the discussion following (2.13).

Set the new basic subset to all points with discrepancies less than c, where

$$c = \sqrt{\chi^2_{d,\alpha/N}} \left(\max \left\{ 0, \frac{h-k}{h+k} \right\} + 1 + \frac{d+1}{N-d} + \frac{1}{N-h-d} \right)$$

with $h = (N+d+1)/2$.

4. The stopping rule: Iterate Step 2 and 3 until the size of the basic subset no longer changes.

5. Nominate the observations excluded by the final basic subset as outliers.

2.3. Risk quantification using a copula model

Among several copula candidates, we settled on the t-copula that is widely used in finance and risk analysis, see Demarta and McNeil (2005). We also considered the popular normal copula, but it did not lead to a good separation of risks for the most extreme pairs. The R package `copula` contains many other copula models that could be used in various settings, and could be better than the t-copula in different applications.

The d-dimensional t-copula with ν degrees of freedom and association matrix Σ is the probability distribution on $[0,1]^d$ whose distribution function is given by

$$C_{\nu,\Sigma}(\boldsymbol{u}) = \int_{-\infty}^{t_\nu^{-1}(u_1)} \ldots \int_{-\infty}^{t_\nu^{-1}(u_d)} \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})\sqrt{(\pi\nu)^d |\Sigma|}} \left(1 + \frac{\boldsymbol{x}' \Sigma^{-1} \boldsymbol{x}}{\nu} \right)^{-\frac{\nu+d}{2}} d\boldsymbol{x} \tag{2.10}$$

where $t_\nu(\cdot)$ is the distribution function of a univariate t-distribution with ν degrees of freedom. The probability density function corresponding to (2.10) equals to

$$c_{\nu,\Sigma}(\boldsymbol{u}) = \frac{dt_{\nu,\Sigma}(t_\nu^{-1}(u_1), \ldots, t_\nu^{-1}(u_d))}{\prod_{i=1}^d dt(t_\nu^{-1}(u_i), \nu)}, \tag{2.11}$$

where $dt_{\nu,\Sigma}(\cdot)$ and $dt(\cdot,\cdot)$ are the densities of multivariate and univariate t-distribution, respectively. We used the R package **copula** to fit copula (2.10). While the t-copula provides a useful separation of risks for the data we study in Section 3, different copulas could be more appropriate for different data sets. Our criterion is that the highest risks should be clearly separated from each other and the bulk of the data.

Risk usually refers to the uncertainty of an outcome given a situation. Cybersecurity risk is the potential for a cybersecurity threat to occur. Following an established practice, we use tail probabilities to quantify risk. To explain the idea, we consider the first two scores, i.e., $\boldsymbol{\rho}_i = (\rho_{i1}, \rho_{i2})$. This corresponds to $d = 2$ used in Section 3. In general, the four cases in (2.12) would be replaced by 2^d cases. Define the probability of scores more extreme than those of the observed pair $(\hat{\rho}_{i1}, \hat{\rho}_{i2})$ as

$$p_i = \begin{cases} P(\rho_{i1} \geq \hat{\rho}_1, \rho_{i2} \geq \hat{\rho}_2), & \text{if } \hat{\rho}_{i1} \geq 0 \text{ and } \hat{\rho}_{i2} \geq 0 \\ P(\rho_{i1} \leq \hat{\rho}_1, \rho_{i2} \geq \hat{\rho}_2), & \text{if } \hat{\rho}_{i1} < 0 \text{ and } \hat{\rho}_{i2} \geq 0 \\ P(\rho_{i1} \leq \hat{\rho}_1, \rho_{i2} \leq \hat{\rho}_2), & \text{if } \hat{\rho}_{i1} < 0 \text{ and } \hat{\rho}_{i2} < 0 \\ P(\rho_{i1} \geq \hat{\rho}_1, \rho_{i2} \leq \hat{\rho}_2), & \text{if } \hat{\rho}_{i1} \geq 0 \text{ and } \hat{\rho}_{i2} < 0. \end{cases} \tag{2.12}$$

The extreme (risky) regions may have a different form, and will look differently in higher dimensions, but (2.12) is a commonly used definition on the plane. We require that in every quadrant, both scores are extreme, rather than just one of them. If the ith pair of traffic flows is anomalous, then it should occur infrequently, i.e., the probability of obtaining ρ_i at least as extreme should be small. To associate high risk with large positive values, we work with negative log probabilities. Thus, the cybersecurity risk of pair i is defined as

$$R_i = -\log(\varepsilon + p_i), \tag{2.13}$$

where $\varepsilon > 0$ is a small value, the same in all calculations. (In Section 3, we use $\varepsilon = 0.001$.) The risks R_i can be used to rank the pairs from most risky to least risky. One can also set a probability threshold α, and consider the pairs satisfying $R_i > -\log(\varepsilon + \alpha)$ as exceptionally risky. We emphasize that α has an interpretation as a probability only within the copula model. Alternatively, one can report α corresponding to 10 or 20, or any other number of most risky pairs. In most applications, we are dealing with thousands of pairs.

3. Application to bi-directional packet flows

3.1. Data description and preliminary analysis

The data set we study consists of a collection of time series of bi-directional packet flows, aggregated hourly, between source Internet protocol (SIP) addresses and destination IP (DIP) addresses captured at a large university from October 20th to 30th, 2013. These data are collected 3 months before a major DDoS attack occurred around January 10th, 2014. The data, transformed with Crypto-PAn, as well as the source code, accompany this paper at the journal's website. During the 250-hour time window over which the data were collected, there are 869 unique SIPs connected with 1869 unique DIPs, and a total of approximately 1.2 million data packets were sent. We consider $N = 3049$ unique SIP-DIP pairs, where SIP is an IP outside the university network and DIP is inside. Each pair is associated with two observed time series, an inbound packet flow and an outbound packet flow. The pairs are labeled with integers $1, 2, \ldots, 3049$, the SIPs with $S1, S2, \ldots, S869$ and the DIPs with $D1, D2, \ldots, D1869$. This is needed to anonymize the IP addresses and ease the notation, the real addresses are long string of integers.

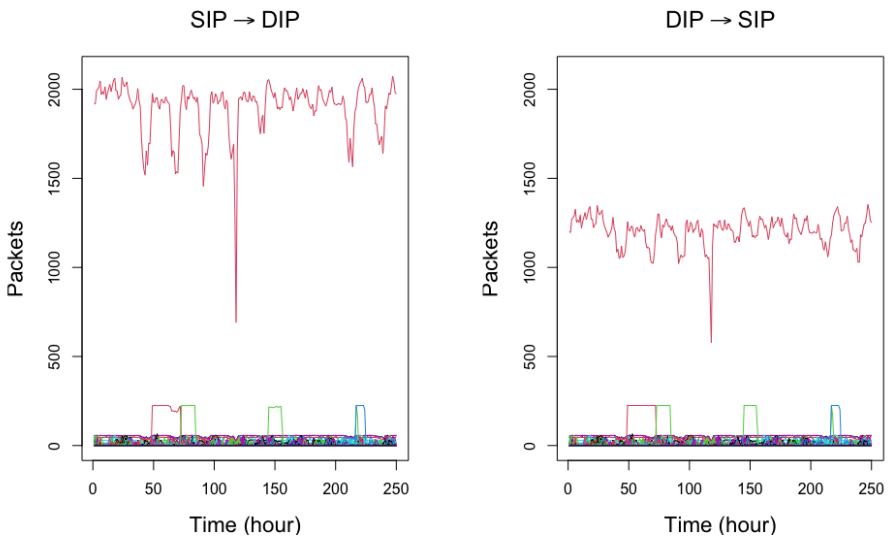

Figure 2: Time series plots of traffic traces. Left: inbound (SIP to DIP); Right: outbound (DIP to SIP). Each time series depicts the hourly count of packets between a SIP-DIP pair.

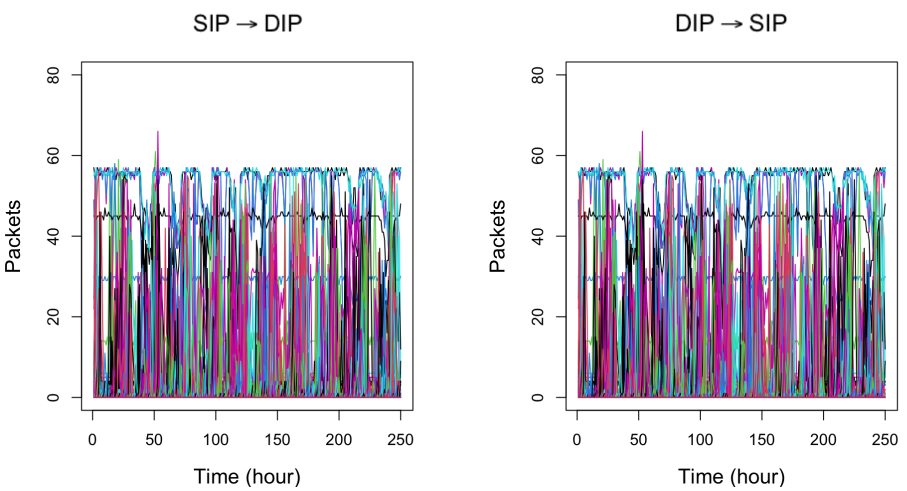

Figure 3: Zoom of Figure 2 with outlying traces removed.

Time series of inbound traffic traces (from SIP to DIP) and outbound traffic traces (from DIP to SIP) are depicted, respectively, in the left and right panels of Figure 2. The hourly count of packets is shown as y-axis, and the time (in hours) is shown as x-axis. It can be seen that there are some clearly, or potentially, outlying packet flows. These are the traces that need to be removed before the computation of the bivariate FPCA is performed. Detection and ordering of risky pairs in the remaining data set shown in Figure 3, cannot be done visually, or by an obvious algorithm. This is why we have developed copula-based algorithms.

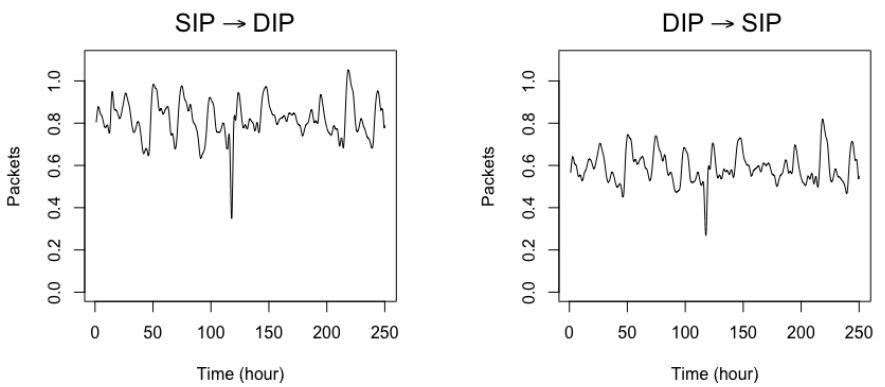

Figure 4: Mean functions of all functions in the sample.

Another justification of the need to develop an algorithm that uses only the pairs that are not obviously outlying comes from the examination of Figures 4 and 5. Figure 4 shows sample mean functions computed from all available data. It is seen that they strongly reflect the extremely outlying curves in Figure 2, one curve in each panel. Similarly, the initial FPCs, shown in Figure 5 reflect the deviations of the mean due to smaller outliers, except the first FPC that reflects the differences in level for most functions. These figures show that the FPCA based on all functions is not suitable for the quantification of risk because it reflects the most risky functions and mostly ignores the bulk of the data. For these reasons, in the following, we first apply the outlier removal algorithms proposed in Section 2.2.

We conclude this section with information about running times. On a 2.2 GHz Intel Core i7 processor, 16GB RAM, the average running times over three repetitions were 27.9 s for Algorithm 1 and 129.9 s for Algorithm 2, for the data set described at the beginning of this section.

3.2. Risk analysis using Algorithm 1

For reasons explained in Section 3.1, before fitting a copula model, we use Algorithm 3 in Section 2.2 to remove outlying curves. It identifies four pairs with abnormal scores (labeled 2, 794, 1077, and 1491). These pairs are excluded from the copula model estimation. Using only the remaining pairs, 95.6% of the variance is explained by the first two MFPCs, with 86.3% of the variance explained by the first MFPC and 9.3% by the second MFPC.

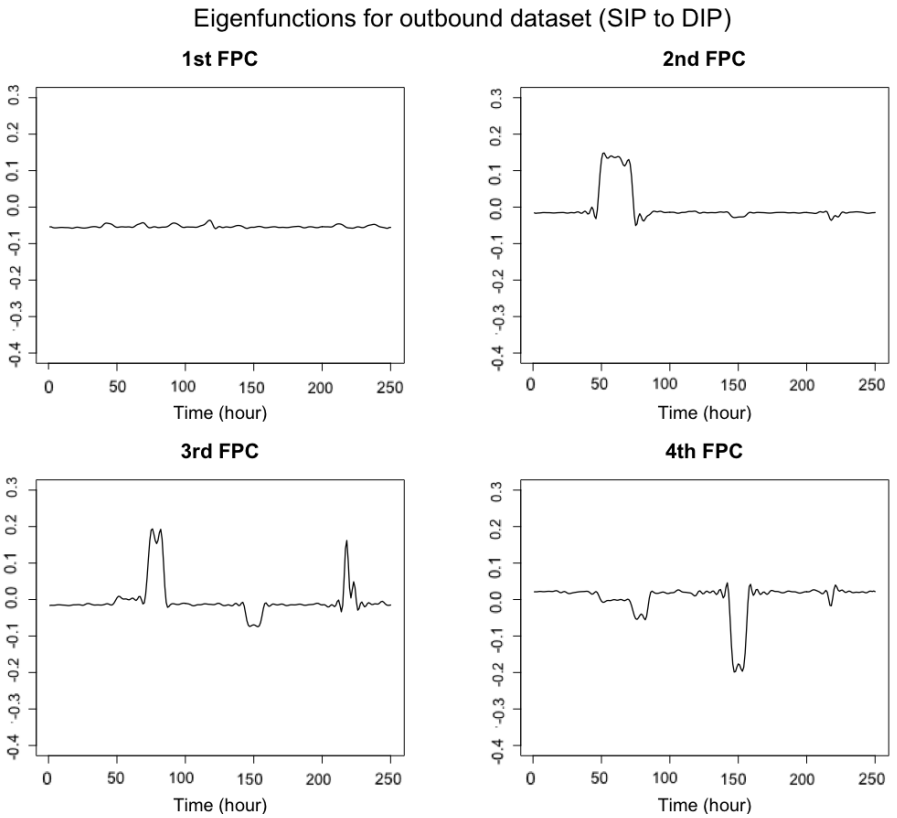

Figure 5: The first four sample FPCs for all functions in the SIP to DIP direction.

Using $d = 2$ is therefore sufficient to capture the main features of the data. After estimating the bivariate t copula (2.10), we compute the probabilities $\hat{p}_i$ using (2.11) for *all* pairs $\hat{\boldsymbol{\rho}}_i = (\hat{\rho}_{i1}, \hat{\rho}_{i2})$, including those that were excluded in the copula estimation. Next, we compute the risks using equation (2.13) with $\varepsilon = 0.001$. The risks are in the range $[0.624, 2.336]$, i.e. $\hat{R}_i \in [0.624, 2.336]$. To give a better idea about the range of risk, we consider, say, 55 pairs with the highest risk. They have risks higher than 1.509. This corresponds to the cut-off level $\alpha = 0.22$, i.e for these 55 pairs, $\hat{R}_i > -\log(\varepsilon + 0.22) = 1.509$. Table 3.2 shows the risks for ten riskiest pairs.

Table 1. The 10 riskiest pairs according to Algorithm 1

Pair	SIP	DIP	Risk
2	S2	D2	2.336
1077	S312	D655	2.0679
1491	S213	D899	2.0404
2260	S312	D1296	1.999
10	S10	D1	1.896
51	S46	D1	1.870
40	S36	D1	1.861
33	S30	D1	1.858
1272	S423	D1	1.854
34	S31	D1	1.820

We note that the risky pairs are found, and their risks computed, using presmoothing with $B = 100$ splines, which is the value used for all analyses presented in this paper. This level of smoothing is suitable to capture the main and relevant features of the data we study. We refer again to Figure 1. We need a level of smoothing that preserves large spikes, but basically ignores typical variability that is not unusual in any way. Larger values of B are not recommended because they would basically reproduce the raw data and distort the MFPCA that requires smooth functions as inputs. Using $B = 50$ produces basically the same risks and identifies almost the same sets of risky pairs. Using fewer that $B = 50$ basis functions is not recommended because the spikes are smoothed out too much.

We examined the patterns of high risk pairs in Table 3.2. The high-risk pairs can, roughly, be classified into three groups, which we denote (a), (b), and (c). Figure 6 shows examples of packet flows in each of the three groups. The curves in group (a) have high levels of packet flows with many rapid drops in the packet counts. Pair 2, the most outstanding outlier, is characterized by exceptionally large traffic. Pair 34 has a similar pattern as pair 2, but the traffic levels are much lower, so it is not displayed in Figure 6. The relatively high levels of activity in group (b) (pairs 1077, 1491, and 2260) last only for a short period of time, and at other times, no activity occurs. The curves in group (c) (pairs 10, 33, 40, 51, and 1272) have generally low levels with many spikes. Only two pairs in groups (b) and (c) are plotted, so as not to obscure the picture.

We also examined other pairs in the group of the 55 riskiest pairs, beyond those in Table 3.2. The general patterns are somewhat different. Basically, the patterns in panels (a) and (b) of Figure 3.2 are exceptional and correspond to outliers. For the majority of high-risk pairs three different groups can be identified. Figure 7 shows examples of packet flows in each of the three groups. The curves in group (a) have moderate levels of packet flows, but exhibit more variability than typical curves. The curves in group (b) have mostly high levels of packet flows with many rapid drops in the packet counts. Group (c) coincides with group (c) in Figure 6. It is basically a mirror image of group (b). The curves in that group have generally low levels with many upward spikes.

It is not possible to display risks for all 3049 pairs in our data set. To obtain some additional insights, we proceed as follows. In the 3049 pairs, there are SIPs that appear more often than others. We thus ranked the SIPs by the frequency with which they appear in the pairs. For example, the address S23 appears most frequently, in 241 out of 3049

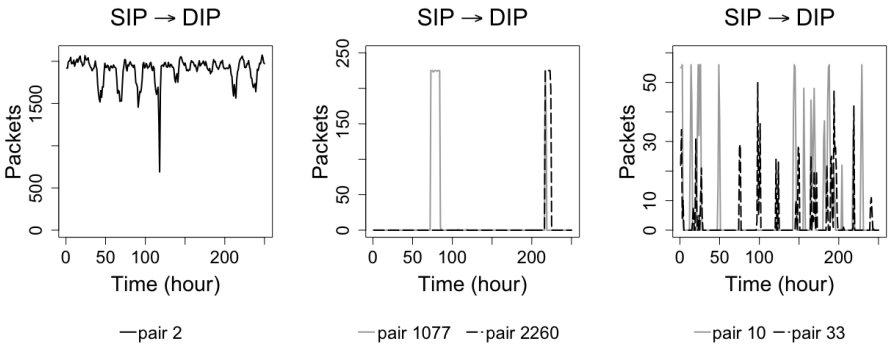

Figure 6: Examples of traffic traces corresponding to the pairs in Table 3.2.

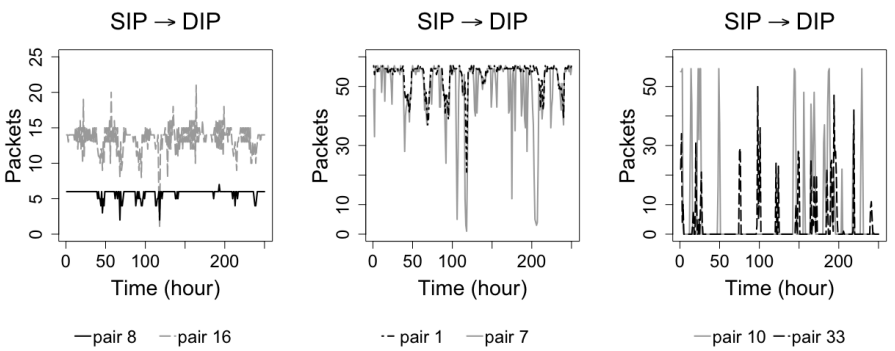

Figure 7: Examples of traffic traces corresponding to pairs identified as high risk under Algorithm 1: Left: pair 8 (S8 → D2), pair 16 (S15 → D2); Middle: pair 1 (S2 → D2), pair 7 (S7 → D2); Right pair 10 (S10 → D1), pair 33 (S30 → D1).

pairs. We performed the same ranking for the DIPs; the address D1 appears most often, in 285 out of 3049 pairs Figure 8 shows risks for the 10 most frequent DIP and SIP addresses. According to Figure 8, the pairs including the 10 most frequent SIPs tend to be less risky, whereas the pairs sent to the 10 most frequent DIPs require more attention, particularly D1 and D2. The DIP D2 was captured by the outlier detection Algorithm 3, but D1 was identified only after computing the risks. A finding of this type may indicate that D1 and D2 (that are within the university) may require special attention.

The results presented in this section illustrate the value of quantitative risk assessment. Certain SIP-DIP pairs are brought to attention by their high risk, even though they are difficult to identify visually due to the fact that we are dealing with thousands of pairs of curves with very complex shapes; in a cloud of thousands of curves it is difficult to see which are more unusual than others, and it is difficult to examine them visually one after another. Our method provides a tool for sorting the SIP-DIP pairs so that attention can be focused only on the riskiest ones.

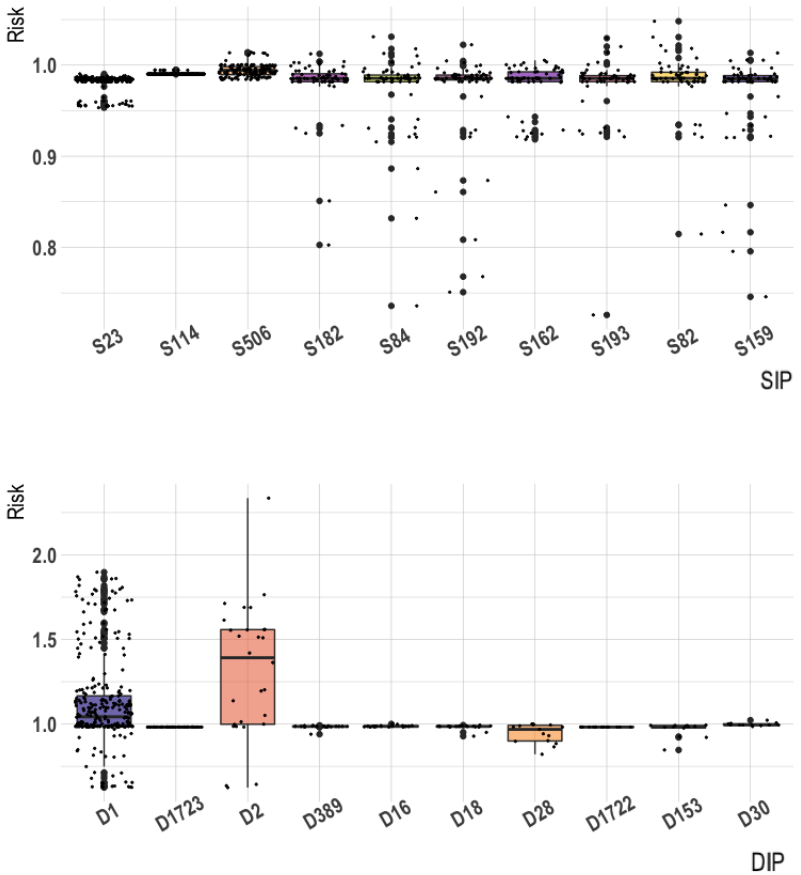

Figure 8: Top: Boxplots of risks for pairs with the 10 most frequent SIPs. Bottom: Boxplots of risks for pairs with the 10 most frequent DIPs. (Algorithm 1)

3.3. Risk analysis using Algorithm 2

We now turn to the application of Algorithm 2 proposed in Section 2.2. The results of copula estimations are shown in Table 3.3. Due to the iterative procedure for the removal of outliers, Algorithm 2 is expected to identify more outliers than Algorithm 1. We emphasize that risks are computed for all pairs, including those identified as outliers. The difference relative to Algorithm 1 is that the copula is estimated on a smaller subset of "typical" pairs. For the data set we study, Algorithm 2 identified 54 pairs as outlying using $\alpha = 0.1$ in Step 3 of Algorithm 3. The first iteration identified, by definition, the same outliers as Algorithm 1: pairs 2, 794, 1077 and 1491. The second iteration identified 42 new outliers, third, 5 outliers and fourth 3. After the fourth iteration no more outliers were identified. The range of $\hat{R}_i$ computed by Algorithm 2 is $[0.870, 2.059]$. The risks are different than those obtained

from Algorithm 1, where the range was $[0.624, 2.336]$. We emphasize that the values of risks are used only for identifying and ranking risky pairs, they do not have an "absolute" interpretation. This point is further highlighted by comparing Figures 8 and 9. Table 3.3 displays the risks of the riskiest pairs identified by Algorithm 2. The pairs in Table 3.3 are different than those in Table 3.2, with some overlap (pairs 2260, 2, 1491, 1272, 40). This is to be expected because different copula models are used to compute them. A change in ranking can also occur if the method is applied to transformed data. We applied Algorithm 2 to $\log(1+\text{count})$ and obtained slightly different, but similar rankings using the level of smoothing similar to that used for the original data. This is understandable, because after any transformation, the curves take on different shapes.

Table 2. Results of the estimation of the t copula based on the two algorithms

	Algorithm 1	Algorithm 2
Degrees of freedom	1.844	3.359
Correlation matrix	$\begin{pmatrix} 1 & 0.344 \\ 0.344 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0.0737 \\ -0.0737 & 1 \end{pmatrix}$
Margin 1	$t_{0.785}(\mu = -0.672, \sigma = 0.0289)$	$t_{1.0769}(\mu = -0.0459, \sigma = 0.0273)$
Margin 2	$t_{0.965}(\mu = 0.0761, \sigma = 0.0295)$	$t_{0.908}(\mu = 0.00433, \sigma = 0.0470)$

Table 3. The 10 riskiest pairs according to Algorithm 2

Pair	SIP	DIP	Risk
2260	S312	D1296	2.0594
794	S312	D13	2.0329
80	S71	D1	2.0294
2	S2	D2	1.995
1491	S213	D899	1.994
79	S70	D1	1.988
43	S39	D2	1.951
1272	S423	D1	1.945
57	S49	D1	1.938
40	S36	D1	1.929

4. Assessment of the methodology on simulated data

A question arises whether Algorithm 1 or Algorithm 2 provides a more useful risk ranking. To address this question, we need an informative simulation study, which is the focus of this section.

The chief difference between Algorithms 1 and 2 of Section 2.2 is as follows. In Algorithm 1, the MFPCs are computed using all available data, even the potential outliers. The largest outliers do not affect the MFPCs because they impact the mean functions that are subtracted before the computation of the MFPCs. In Algorithm 2, the MFPCs are computed after the outliers have been removed. For example, in Section 3.3 they were computed after 54 pairs had been removed. We assess the performance, and relative performance, of the two algorithms using simulated data that has certain features of our real data sets, but also certain characteristics that are known targets. In step 1 of the following data generation algorithm, we have two options, A and B. Option A might seem to, a priori, favor Algorithm

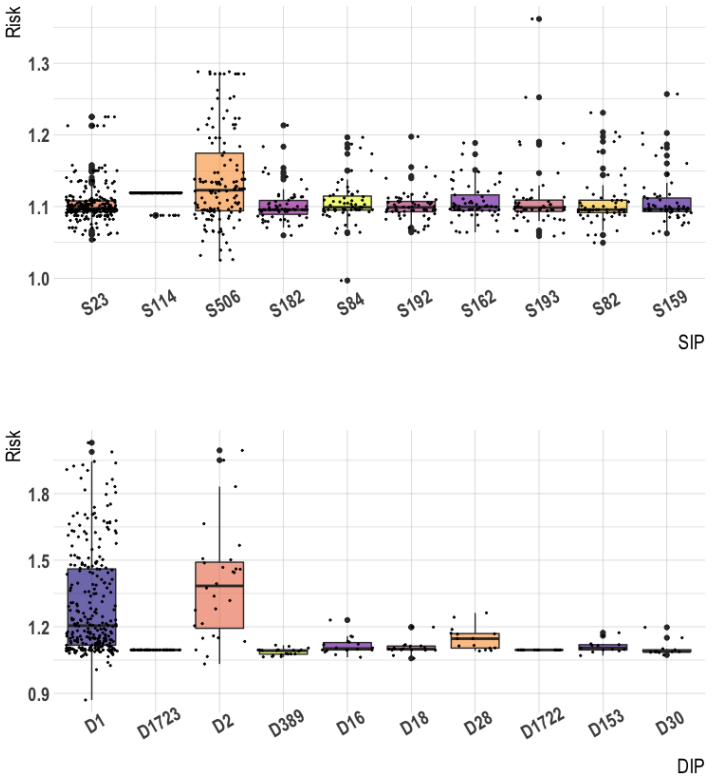

Figure 9: Top: Boxplot of risks for pairs with the 10 most frequent SIPs. Bottom: Boxplot of risks for pairs with the 10 most frequent DIPs. (Algorithm 2)

1 and Option B Algorithm 2.

1. A Estimate $\psi_1^{(1)}, \psi_2^{(1)}, \psi_3^{(1)}$ and $\psi_1^{(2)}, \psi_2^{(2)}, \psi_3^{(2)}$ using all data.

   B Remove 54 pairs identified by Algorithm 2 as outlying and estimate $\psi_1^{(1)}, \psi_2^{(1)}$, $\psi_3^{(1)}$ and $\psi_1^{(2)}, \psi_2^{(2)}, \psi_3^{(2)}$ based on the remaining 3049 - 54 = 2995 pairs.

2. For $1 \leq i \leq 2995$, generate

$$X_i = \sum_{j=1}^{3} \xi_{ij} \psi_j^{(1)}, \quad Y_i = \sum_{j=1}^{3} \eta_{ij} \psi_j^{(2)} \tag{4.14}$$

with iid scores ξ_{ij} and η_{ij} distributed according to

$$\xi_1 \sim t_{10}, \;\; \xi_2 \sim 0.5\, N(0,1), \quad \xi_3 \sim 0.1\, N(0,1),$$

$$\eta_1 \sim t_{11}, \;\; \xi_2 \sim 0.4\, N(0,1), \quad \xi_3 \sim 0.2\, N(0,1),$$

The first 2995 pairs are the typical low risk pairs.

3. For $2996 \le i \le 3041$, generate the pairs (X_i, Y_i) according to (4.14), but ξ_1 and η_1 having different, "larger" distributions, as specified below. The remaining distributions are unchanged. These are the pairs with increasing risks. Pair 2996 has the smallest risk of them, pair 3041 the highest.

4. For $3042 \le i \le 3049$, generate the pairs (X_i, Y_i) according to (4.14), but with ξ_1 and η_1 having "extremely" large distributions. These are the outlying pairs

In steps 3 and 4 above, the distribution of the scores changes, so as reduce the dependence of the the conclusions on a specific distribution of risky and outlying pairs. We repeat steps 1-4 20 times, and use four distributions for each batch of five simulations according to the following specifications:

Simulations 1 to 5: For $2996 \le i \le 3041$, $\xi_1 \sim (i-2995)t_{10}, \eta_1 \sim (i-2995)t_{11}$; for $3042 \le i \le 3049$, $\xi_1 \sim 2(i-3041)t_3, \eta_1 \sim 2(i-3041)t_4$.

Simulations 6 to 10: For $2996 \le i \le 3041$, $\xi_1 \sim (i-2995)\text{Exp}(0.5), \eta_1 \sim (i-2995)\text{Exp}(1)$; for $3042 \le i \le 3049$, $\xi_1 \sim (i-3041)\text{Exp}(0.1), \eta_1 \sim (i-3041)\text{Exp}(0.5)$;

Simulations 11 to 15: For $2996 \le i \le 3041$, $\xi_1 \sim \frac{2i-2995}{10}\text{Exp}(1), \eta_1 \sim \frac{2i-2995}{11}\text{Exp}(2)$; for $3042 \le i \le 3049$, $\xi_1 \sim \frac{2i-3041}{5}\text{Exp}(1), \eta_1 \sim \frac{2i-3041}{6}\text{Exp}(2)$.

Simulations 16 to 20: For $2996 \le i \le 3041$, $\xi_1 \sim (i-2995)\text{Exp}(0.5), \eta_1 \sim (i-2995)t_{11}$; for $3042 \le i \le 3049$, $\xi_1 \sim (i-3041)\text{Exp}(0.1), \eta_1 \sim (i-3041)t_4$.

We apply Algorithms 1 and 2 to the data generated above. Note that each algorithm estimates the MFPCs and the scores. The estimated MFPCs will be different than those used to generated the data in Step 1. We list the pairs identified as outliers. The target list are pairs $3042, 3043, \ldots, 3049$. We find 54 riskiest pairs and order them from the one with the smallest risk to the one with the highest risk (according to each algorithm). We denote the indexes as $i_1, \ldots, i_{54}$. These indexes will be different for the two algorithms. The pair (X_{i_1}, Y_{i_1}) has the the lowest risk out of the 54 pairs. We compute the absolute differences $|i_k - k - 2995|$, $k = 1, \ldots, 54$, and plot them as histograms for both algorithms. If an algorithm performs well, these differences should be small. For an algorithm that detects outliers perfectly and ranks the risks perfectly, they should all be zero. However, due to the random generation of outlying and risky pairs, some of them will not appear to be in these categories because even a t_3 distribution can take a value close to zero. However, our experiment should give a reasonable idea how the algorithms perform, as we now report.

In both scenarios A and B, Algorithm 1 identifies five to nine pairs as outlying and Algorithm 2 eight to seventeen pairs. In this sense, Algorithm 1 is closer to our target of seven outlying pairs. However, as shown in Figure 10, Algorithm 2 has an advantage in ranking the risky and outlying pairs, but is more prone to make serious mistakes more often that Algorithm 1. The reader can certainly draw conclusions from the above analysis, but it appears that the additional outliers identification step in Algorithm 2 does not provide a decisive improvement. One might conclude that both algorithms identify outliers and risky pairs in a satisfactory manner, but may result in somewhat different risk rankings, as we have seen in Section 3.

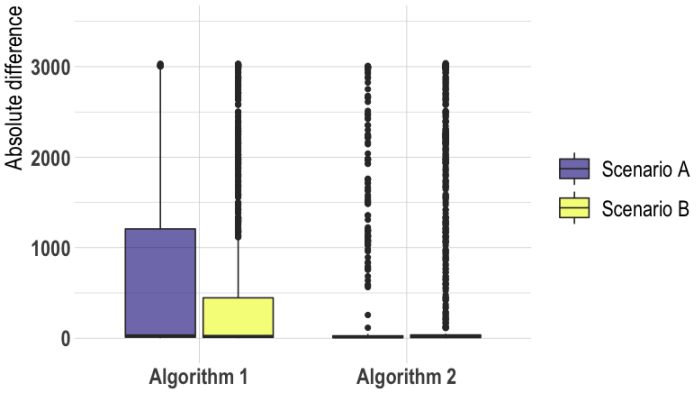

Figure 10: Boxplots of absolute difference for the top 54 risky pairs for both algorithms in two scenarios.

Acknowledgements

This research was partially supported by the United States National Science Foundation grant DMS–2123761.

References

Ahmed, Mohiuddin, Mahmood, Abdun Naser and Hu, Jiankun, (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60**,** pp. 19–31.

Amovin-Assagba, Martial, Gannaz, Irène and Jacques, Julien, (2022). Outlier detection in multivariate functional data through a contaminated mixture model. *Computational Statistics & Data Analysis*, 174**,** 107496.

Awan, Mazhar Javed, Farooq, Umar, Babar, Hafiz Muhammad Aqeel, Yasin, Awais, Nobanee, Haitham, Hussain, Muzammil, Hakeem, Owais and Zain, Azlan Mohd, (2021). Real-time DDoS attack detection system using big data approach. *Sustainability*, 13**,** no. 19, 10743.

Berrendero, José R, Justel, Ana and Svarc, Marcela, (2011). Principal components for multivariate functional data. *Computational Statistics & Data Analysis*, 55**,** no. 9, pp. 2619–2634.

Billor, Nedret, Hada, Ali and Velleman, Paul, (2000). BACON: blocked adaptive computationally efficient outlier nominators. *Computational Statistics and Data Analysis*, 34**,** pp. 272–298.

Bosq, Dennis, (2000). *Linear Processes in Function Spaces*. Springer.

Chiou, Jeng-Min, Chen, Yu-Ting and Yang, Ya-Fang, (2014). Multivariate functional principal component analysis: A normalization approach. *Statistica Sinica*, pp. 1571–1596.

Czado, Claudia, (2019). *Analyzing Dependent Data with Vine Copulas: A Practical Guide with R*. Springer.

Dai, Wenlin and Genton, Marc G., (2018). Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics*, 27**,** no. 4, pp. 923–934.

Demarta, Stefano and McNeil, Alexander, (2005). The *t* copula and related copulas. *International Statistical Review*, 73**,** pp. 111–129.

Dong, Shi and Sarem, Mudar, (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access*, 8**,** pp. 5039–5048.

Ferraty, Frédérick and Vieu, Philippe, (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer.

Fouladi, Ramin Fadaei, Kayatas, Cemil Eren and Anarim, Emin, (2016). Frequency based DDoS attack detection approach using naive Bayes classification. In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 104–107. IEEE.

Fouladi, Ramin Fadaei, Seifpoor, Tina and Anarim, Emin, (2013). Frequency characteristics of DoS and DDoS attacks. In *2013 21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4. IEEE.

Genest, Christian and Nešlehová, Johanna, (2012). Copulas and copula models. In *Encyclopedia of Environmetrics* (eds El-Shaarawi A.H. and Piegorsch W.W.), 2 edn, volume 2, pp. 541–553. Wiley, Chichester.

Górecki, Tomasz, Krzyśko, Mirosław, Waszak, Łukasz and Wołyński, Waldemar, (2018). Selected statistical methods of data analysis for multivariate functional data. *Statistical Papers*, 59**,** no. 1, pp. 153–182.

Happ, Clara and Greven, Sonja, (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113**,** number 522, pp. 649–659.

Hofert, Marius, Kojadinovic, Ivan, Mächler, Martin and Yan, Jun, (2018). *Elements of Copula Modeling with R*. Springer.

Horváth, Lajos and Kokoszka, Piotr, (2012). *Inference for Functional Data with Applications*, volume 200. Springer Science & Business Media.

Hubert, Mia, Rousseeuw, Peter J and Vanden Branden, Karlien, (2005). ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47**,** no. 1, pp. 64–79.

Hussain, Alefiya, Heidemann, John and Papadopoulos, Christos, (2003). A framework for classifying denial of service attacks. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 99–110.

Jacques, Julien and Preda, Cristian, (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71**,** pp. 92–106.

Joe, Harry, (2015). *Dependence Modeling with Copulas*. Chapman & Hall.

Kokoszka, Piotr and Reimherr, Matthew, (2017). *Introduction to Functional Data Analysis*. Chapman and Hall/CRC.

Krzyśko, Mirosław and Smaga, Łukasz, (2020). Measuring and testing mutual dependence of multivariate functional data. *Statistics in Transition*, 21**,** no. 3, pp. 21–37.

Krzyśko, Mirosław and Smaga, Łukasz, (2021). Two-sample tests for functional data using characteristic functions. *Austrian Journal of Statistics*, 50**,** no. 4, pp. 53–64.

Liao, Hung-Jen, Lin, Chun-Hung Richard, Lin, Ying-Chih and Tung, Kuang-Yuan, (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36**,** no. 1, pp. 16–24.

Modi, Chirag, Patel, Dhiren, Borisaniya, Bhavesh, Patel, Hiren, Patel, Avi and Rajarajan, Muttukrishnan, (2013). A survey of intrusion detection techniques in Cloud. *Journal of Network and Computer Applications*, 36**,** no. 1, pp. 42–57.

Nelsen, Roger, (2006). *An Introduction to Copulas*. Springer.

Nishanth, N. and Mujeeb, A., (2020). Modeling and detection of flooding-based denial-of-service attack in wireless ad hoc network using Bayesian inference. *IEEE Systems Journal*, 15**,** no. 1, pp. 17–26.

Peng, Chen, Xu, Maochao, Xu, Shouhuai and Hu, Taizhong, (2018). Modeling multivariate cybersecurity risks. *Journal of Applied Statistics*, 45**,** no. 15, pp. 2718–2740.

Ramsay, James and Silverman, Bernard, (2005). *Functional Data Analysis*. Springer.

Sambangi, Swathi and Gondi, Lakshmeeswari, (2020). A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regression. In *Proceedings*, volume 63, p. 51. MDPI.

Soysal, Murat and Schmidt, Ece Guran, (2010). Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation*, 67**,** no. 6, pp. 451–467.

Wu, Zhijun, Yue, Meng, Li, Douzhe and Xie, Ke, (2015). SEDP-based detection of low-rate DoS attacks. *International Journal of Communication Systems*, 28**,** no. 11, pp. 1772–1788.

Xu, Maochao, Hua, Lei and Xu, Shouhuai, (2017). A vine copula model for predicting the effectiveness of cyber defense early-warning. *Technometrics*, 59**,** no. 4, pp. 508–520.