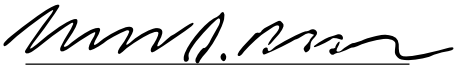
APPROVAL SHEET

Title of Thesis: Network Anomaly Detection via Persistent Homology

Name of Candidate: Joseph Robert Collins
M.S., 2019

Thesis and Abstract Approved: 

Dr. Christopher Marron
Professor of the Practice
Department of Computer Science and
Electrical Engineering

Date Approved: April 26, 2019

ABSTRACT

Title of Thesis: Network Anomaly Detection via Persistent Homology

Joseph Robert Collins, M.S. Computer Science, 2019

Thesis directed by: Dr. Christopher Marron
Professor of the Practice
Department of Computer Science and
Electrical Engineering

Network anomaly detection has wide ranging applications, to include fraud prevention and cybersecurity. This paper introduces several methods of network anomaly detection derived from topological data analysis (TDA). At a high level, TDA captures the qualitative geometric features of data. The primary tool of TDA is persistent homology, which is used to analyze the ”holes” present in data. When applied to networks, the generated features provide insight into global and local trends. Specifically, we employ persistence landscapes generated directly from the weight ranked clique filtration (WRCF) of communication graphs. This obviates the need for graph embedding. The graph construction is application dependent, with communications frequency being the natural choice for edge weight in most cases. Applying persistent homology to this filtration yields a persistence landscape, which is used as a graph invariant. This research aims to show that anomalous behavior corresponds to detectable deviation from previous persistence landscapes. By calculating the persistence landscapes of local neighborhoods around individual vertices over time, suspicious behavior can be detected. The persistence landscapes of the entire network over time are used to detect global changes in behavior corresponding to major events.

Network Anomaly Detection via Persistent Homology

by

Joseph Robert Collins

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
M.S. Computer Science
2019

Dedication: RCB Jr.

ACKNOWLEDGMENTS

I would like thank Dr. Christopher Marron for his advice, guidance, and patience. He has invested substantial time helping to develop the ideas presented in this paper. His mathematical approach has taught me to solve problems from rigorous first principles. I am grateful to have him as a teacher and mentor.

I would also like to thank all of the teachers who have supported me up to this point. I am also grateful to UMBC for their continuing support.

Finally, I would like to thank my family for their unwavering love, patience, and support. My parents, Mike and Lisa, have and continue to be my greatest teachers. My brother, Michael, has been an excellent friend and role model, with a work ethic second to none. You all are the reason I have made it this far.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

For a wide variety of applications it is critical to detect network anomalies. Two domains of particular interest are fraud prevention and cybersecurity. In both cases, network anomalies are associated with negative outcomes. For this reason, it is important to detect anomalies early and accurately. Additionally, any detection method must be robust, and not susceptible to small changes causing false negative results.

Network data is often multimodal, meaning the choice of representation it not obvious. There are many ways to combine the different modalities of graph data given specific application (Goyal & Ferrara 2018). While useful, these methods are often computationally expensive, especially as the size of the dataset grows. Therefore, methods which do not require embedding or significant transformation of the underlying data are desirable. Directly modeling network data as a weighted graph obviates the need for graph embedding. While this method requires calculation of edge weights, this introduces minimal complexity and there is often a natural choice for any given application.

Topological data analysis (TDA) is well suited to this type of problem. Originally developed by Gunnar Carlson, this field is concerned with quantifying the topology of data (Carlsson 2009). The primary tool of TDA is persistent homology. This approach captures the qualitative characteristics of data at different scales, meaning it is robust to

small changes in the data. Most often, it is applied to point cloud data lying in $\mathbb{R}^n$. In order to compute the persistent homology of such data, it must first be converted to a graph based representation called a simplicial complex.

Given that network data already can easily be represented as a weighted graph, it is a natural next step to directly apply persistent homology. From an anomaly detection perspective, this eliminates the need to embed or significantly transform the network data. On the TDA side of the detector, there is no point cloud data which must be converted to a graph; the data is already in the correct form.

Regardless of the domain, network anomalies can be classified as local or global occurrences. Both of these anomaly types are of interest, with the interpretation of each varying based on the application. For instance, in the context of fraud prevention, a local anomaly detection should indicate which nodes are potentially committing fraud. In cybersecurity related applications, a global anomaly might indicate a DDoS or similar widespread attack. With minimal modifications, the same TDA based methods can be used to detect both local and global anomalies, the main difference being the scale of the data.

This paper develops a persistent homology based invariant computed directly from the graph representation of the network. Computed over the neighborhoods surrounding individual vertices, this invariant is used to detect local anomalies. Similarly, applied to the entire network graph, it is used to detect global anomalies.

Chapter 2

BACKGROUND AND RELATED WORK

The techniques presented in this paper are derived from existing tools in TDA and network anomaly detection. The first section of this chapter provides an overview of the definitions and tools which make TDA possible. The second section provides an overview of anomaly detection techniques similar to those developed in Chapter 3.

2.1 Topological Data Analysis

For the purposes of this paper, TDA proceeds as follows. The first step is to construct a filtered simplicial complex, or filtration, from the data. Each step of the filtration is a simplicial complex constructed for some value of a scale parameter. Next, the persistent homology of the filtration is computed. The resulting intervals summarize at which steps in the filtration ”holes” in the data are created and closed. These intervals are most often represented as a barcode graph. A persistence landscape can be derived from the barcode of a filtration. By taking the distance between persistence landscapes, it is possible to compare the persistent homology of different filtrations. The remainder of this section will define the key terms above, and clarify critical steps.

2.1.1 Simplicial Complexes

The simplicial complex is the fundamental building block of persistent homology. This section is mainly distilled from Gunnar Carlson's seminal paper on TDA (Carlsson 2009). The reader is encouraged to refer to that paper for a full treatment of mathematics underlying simplicial homology.

Definition 2.1. *A simplicial complex is a pair $X = (V, \Sigma)$, where V is a finite set, and Σ is a family of non-empty subsets of V such that $\sigma \in \Sigma \wedge \tau \subseteq \sigma \implies \tau \in \Sigma$.*

Each $\sigma \in \Sigma$ is a simplex. Interpreting the simplicial complex as a graph, a simplex corresponds to a vertex, edge, triangle, or similar higher dimensional structure. If $|\sigma| = k$ then σ is a $(k-1)$-simplex. For instance, $\sigma = \{1, 2, 3, 4\}$ is a 3-simplex corresponding to the tetrahedron formed by the clique of the included vertices.

Importantly, it is possible to compute the homology of a simplicial complex. Given $X = (V, \Sigma)$, Σ_k is the set of k-simplices. A k-chain is any formal linear combination of elements of Σ_k. $C_k(X)$ is the free abelian group over Σ_k, or the set of all k-chains in X. The boundary operator, which yields the $k-1$ cofaces of the simplices in Σ_k, is $\partial_k : C_k(X) \to C_{k-1}(X)$, and defined as $\partial_k = \sum_{i=0}^{k} (-1)^i d_i$, where $d_i(\sigma) = \sigma - \{s_i\}$ and s_i is the i^{th} vertex in σ. The k^{th} homology group of a simplicial complex is defined as $H_k(X, \mathbb{Z}) \cong Kernel(\partial_k)/Image(\partial_{k+1})$.

Intuitively, $Kernel(\partial_k)$ corresponds to k-chain boundaries in the complex, and $Image(\partial_{k+1})$ corresponds to the k-chains which bound $(k+1)$-simplices. Therefore, H_k contains the k-chains of the simplicial complex which are boundaries, but do not bound $(k+1)$-simplices. In other words, H_k is the group of holes bound by k-simplices. Note, this means that in order to compute H_k, the simplicial complex must contain $(k+1)$-simplices.

2.1.2 Filtration

This section outlines how to create a simplicial complex from data. A filtration of a complex X results in a filtered simplicial complex, or filtered complex.

Definition 2.2. *A filtration of a complex X is a nested subsequence of complexes* $\emptyset = K^0 \subseteq K^1 \subseteq ... \subseteq K^M = K$. *(Zomorodian & Carlsson 2004)*

Again, the goal of persistent homology is to capture how the data behaves across different scales by computing the homology of each step in the filtration. This means that the construction of the filtration should capture the data at different scales in a meaningful way. There are several techniques which have been developed to accomplish this, both for point cloud and graph-based data.

While this paper does not make use of point cloud data, it is an active area of research in TDA and is useful for exploring the basic idea. Given point cloud data in an N-dimensional space, there are three main methods used. In order of increasing efficiency, these are the Čech complex, Vietoris-Rips complex, and the Witness complex (Carlsson 2009). The later two are optimizations of the Čech complex, making it possible to process large amounts of data.

Definition 2.3. *Let* B_ϵ^p *be the* ϵ*-ball around vertex point p, all points within distance* ϵ *of p. Then the Čech Complex of a simplicial complex,* $C(X, \epsilon)$*, where* $\epsilon > 0$*, is the set S such that* $\sigma \in S$ *if the* $\bigcap_{p \in \sigma} B_\epsilon^p \neq \emptyset$

Intuitively, this means that a set of points is added as a simplex in the complex if the ϵ-ball for every point in the set overlaps with the ϵ-ball of every other point in the set. This is illustrated in Figure 2.1. It is clear from the figure that as ϵ is increased, the resulting sequence of simplicial complexes satisfies the definition of a filtration. The Vietoris-Rips complex is similar to the Čech complex, but less expensive to compute. An edge is added

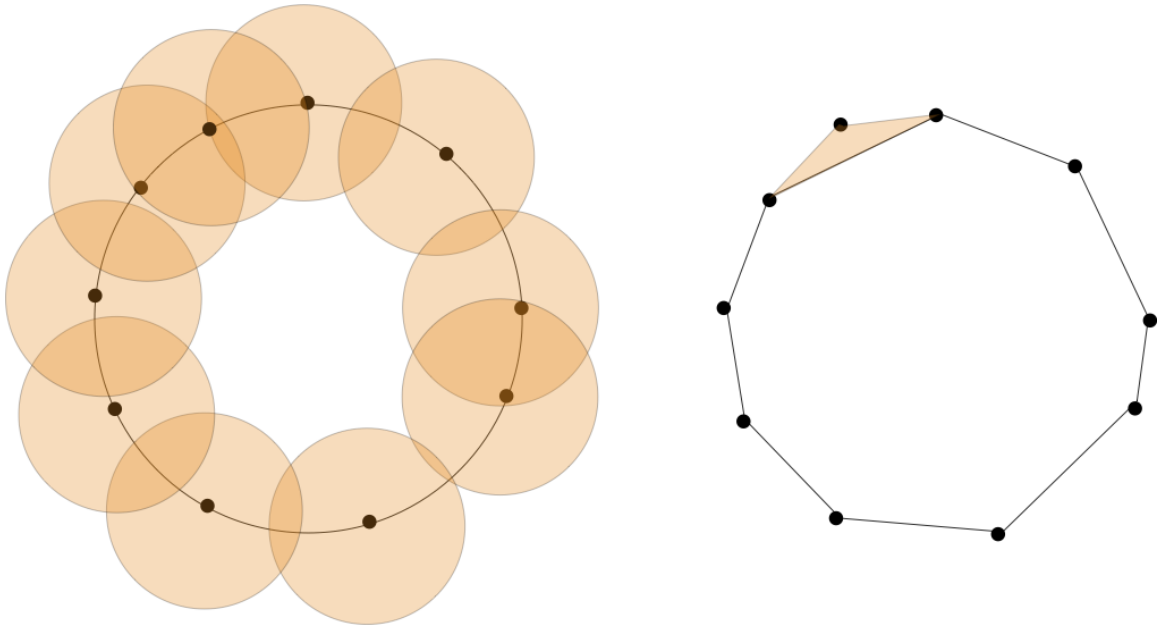


FIG. 2.1. An illustration of how a Čech Complex is formed. Distributed under the CC BY 2.0 license (Foundation 2019).

between two points if the distance between the points is less than ϵ. The clique complex of the resulting graph is computed in order to form a simplicial complex. Again, ϵ is increased to create a filtration. The Witness complex is computed in a similar manner, but using specially selected points to quickly approximate the Vietoris-Rips complex (Carlsson 2009). Note, both of these filtrations rely on finding the clique complex of a graph. Methods for computing clique complexes are outlined in Section 2.1.5.

Definition 2.4. *Given an undirected graph G with edges E and vertices V, the clique complex of G is* $X_G = (V, \Sigma^G)$*, where* $\sigma \in \Sigma_G$ *if the vertices of* σ *form a clique. The elements of* Σ_k^G *correspond to (k+1)-cliques.*

It is also possible to construct a filtered complex directly from graph data instead of point cloud data. There are several approaches to accomplish this, but the most promising is the Weight Rank Clique Filtration (WRCF) over a weighted undirected graph (Petri *et*

al. 2013). For this filtration, each step of the filtration corresponds to a unique edge weight in the graph. Edges can either be added by increasing or decreasing weight, depending on the application and distribution of edge weights.

2.1.3 Persistent Homology

Section 2.1.1 showed how to compute the homology of a simplicial complex. Given a filtered simplicial complex, it is possible to compute persistent homology. At a high level, persistent homology is a record of when "holes" are created and destroyed in the filtered complex. If a homology class α is born at step K_i of a filtration and dies at step K_j, the persistence of α is $j - i$ (Edelsbrunner & Harer 2008). Alternatively, the persistence interval of α is $(i, j]$. The persistence intervals for all homology classes of a dimension can be represented as a barcode as shown in Figure 2.2. While barcodes are a natural way to represent the persistent homology of a filtered complex, it is difficult to compare barcodes across filtrations. This problem is addressed in Section 2.1.4.

Ultimately, computing persistent homology reduces to a series of row and column operations on the boundary matrices of the filtration. The boundary matrices are constructed using the boundary operators from Section 2.1.1. Specifically, the boundary matrix used to compute the p^{th}-persistent homology is defined as follows.

Definition 2.5. $D_p[i, j] = 1$ *if the* i*-th* $p-1$*-simplex is a face of the* j*-th p-simplex and 0 otherwise (Edelsbrunner & Harer 2008).*

The p^{th}-persistent homology of the filtration is computed by converting the D_p to Smith normal form. This process takes $O(m^3)$, where m is the number of simplices in the filtered complex (Zomorodian & Carlsson 2004).

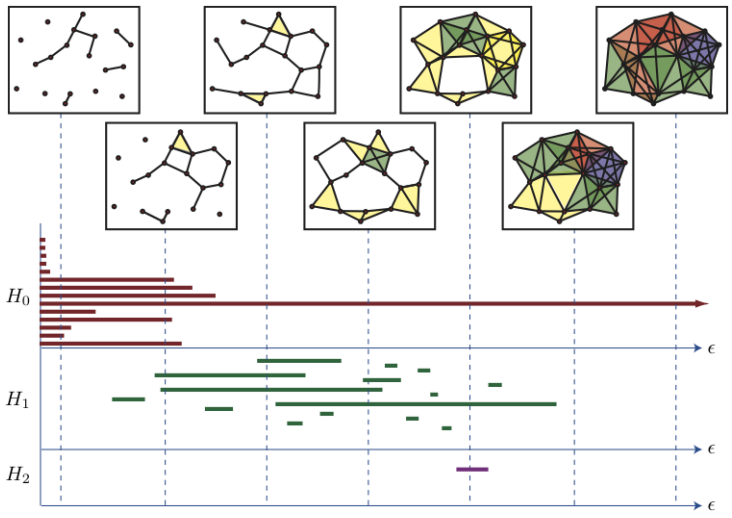


FIG. 2.2. An example of a barcode for 0, 1, and 2 dimensional persistent homology of the filtered complex (Ghrist 2008)

2.1.4 Persistence Landscapes

The persistence landscape is a topological summary of the persistent homology of a space. It is useful because it lies in a vector space, is equipped with an L_p norm, and has a statistical interpretation (Bubenik 2015). This construction allows for easier comparison of the topological characteristics of data. Figures 2.3 and 2.4 show a partial and complete persistence landscape, respectively.

Definition 2.6. *A persistence landscape is a piecewise linear function* $\mathcal{L} : \mathbb{Z}^+ \times \mathbb{R} \to \mathbb{R}$. $\mathcal{L}_\mathcal{P}(k, z) = \underset{p \in \mathcal{P}}{kmax}\ t_p(z)$, *where* $kmax$ *is the* k^{th} *largest value in the set. Given a set of intervals (b,d] denoting the birth and death of a "hole", each persistence point* $p = (x, y) = (\frac{b+d}{2}, \frac{b-d}{2})$ *is replaced with the following triangle function (Chazal* et al. *2013):*

$$t_p(z) = \begin{cases} z - d & z \in [d, \frac{b+d}{2}] \\ b - z & z \in [\frac{b+d}{2}, b] \\ 0 & otherwise \end{cases}$$

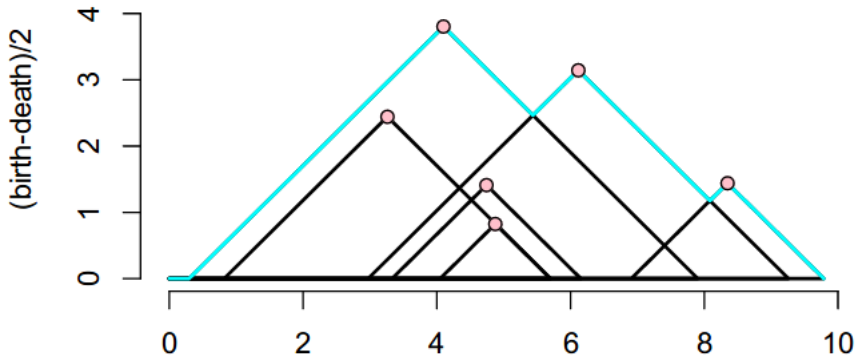


FIG. 2.3. A partial persistence landscape corresponding to $\mathcal{L}(1,\cdot)$ over time (Chazal *et al.* 2013).

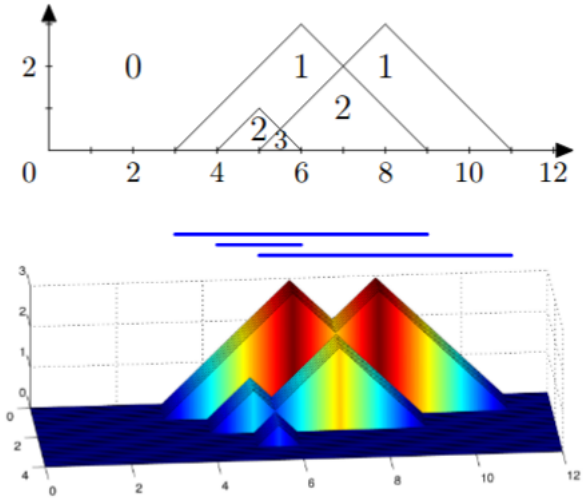


FIG. 2.4. An example of a barcode, its representation using t_p, and the resulting persistence landscape (Bubenik 2015).

2.1.5 Computation of Clique Complexes

As previously mentioned, finding the clique complex of a graph is a critical step in constructing many filtrations. There are three published algorithms to solve this problem (Zomorodian 2010). All three algorithms find cliques in a graph up to size k, corresponding to $(k-1)$-simplices. Finding the cliques of a graph is ultimately a hard combinatorics problem. There is currently no known upper bound on the run time of these algorithms. Empirically, the iterative approach has been shown to be the fastest. This is the algorithm used to construct simplicial complexes in Chapter 4.

2.2 Related Work

The field of anomaly detection is well researched and includes a wide range of techniques. This paper primarily focuses on developing an anomaly score which can be applied locally to detect anomalous nodes and globally to detect events. This section will review related methods used to detect local and global anomalies in graph-based data.

2.2.1 Scan Statistics

Graph based scan statistics are an efficient and effective method of local anomaly detection. The scan statistic around a given node is usually computed over network graphs from a sliding window of time. Changes in the scan statistics can be used to detect anomalous behavior.

Definition 2.7. *The closed k^{th}-order neighborhood of vertex v in a directed graph D is $N_k[v, D] = \{w \in V(D) | d(v, w) \leq k\}$, where $V(D)$ is the vertex set of D (Priebe* et al. *2005).*

Definition 2.8. *Let the scan region $\Omega(N_k[v, D])$ be in the subgraph induced by the vertices*

of $N_k[v, D]$. *A locality statistic is any graph invariant* $\Psi_k(v)$ *over the scan region (Priebe* et al. *2005).*

By computing how the locality statistic of a vertex changes over time, it is possible to detect local anomalous behavior. This technique has successfully been applied to the Enron email dataset (Priebe *et al.* 2005). The invariants developed in Chapter 3, while not actual locality statistics, are closely related.

2.2.2 TDA and Networks

Other anomaly detectors have been developed using persistent homology as a graph invariant (Bruillard, Nowak, & Purvine 2016). These detectors rely on first converting the network data to point cloud data via a vectorization process. By directly comparing barcodes of the 0^{th}-persistent homology (corresponding to connected components), these detectors are able to find anomalies under specific circumstances.

2.2.3 Other Detectors

Global event detection can be accomplished using a wide variety of tools. Neural networks, Bayesian networks, and rule-based systems have all been applied with varying levels of success (Chandola, Banerjee, & Kumar 2009). These techniques, while interesting, are not closely related to the topological methods developed in this paper. Persistent homology is designed to work well across different scales, so this means the techniques developed for local detection also work well for global event detection.

Chapter 3

METHODS

This chapter outlines methods developed to apply persistent homology to network data. These methods are applied to the Enron email dataset (Enron 2015), so this section begins with a description of the data and how it was processed. There are two tasks which must be accomplished in order to successfully apply persistent homology to network data. First, a filtration must be generated from the given data. Second, the resulting barcodes/landscapes must be processed and interpreted. The two methods differ mainly in how the filtration is constructed. The second method, based on a WRCF, best captures the network structure and forms the basis for anomaly detection.

3.1 The Enron Email Corpus

The Enron email dataset is the largest publicly available email dataset. The corpus has been redacted significantly over time. In this context, persistent homology relies on the formation of cliques. It is unlikely that communication with non-Enron email addresses will result in interesting cliques. For this reason, all emails to or from non-Enron users were excluded. The resulting induced sub-graph of Enron users contains 32,062 nodes and 194,306 edges, far more than the 124 users the data was collected from.

The ultimate goal of this research is to develop real-time anomaly detection via persis-

tent homology. For such an application, it is useful to compare past and present behavior. To facilitate this comparison, the data was windowed by week. Significant email traffic first starts to appear in November of 1998 and tapers off in June of 2002. With this in mind, the data was partitioned into one week graphs, starting Sunday 11/1/98, encompassing 189 weeks in total. This is the same division used in similar research (Priebe *et al.* 2005). The edges of these graphs are weighted by total communication between two nodes during a given week. As is shown in Figure 3.2 the distribution of edge weights has a long tail likely given by a power law, with most of the edges having weight one.

3.2 k^{th}-order Neighborhood Filtration

This filtration is derived from the concept of a k^{th}-order neighborhood from Definition 2.7. There are two variations of this filtration. The first is defined as the forward scan filtration, F_k. Each step in this filtration is an induced neighborhood $N_i[v, D]$ around a given node for $1 \leq i \leq k$. Step 0 of the filtration for vertex v corresponds to the induced 1^{st}-order neighborhood around v. An example filtration can be seen in Figure 3.1. While an obvious first attempt, this method does not yield any useful persistent homology intervals for the typical neighborhoods around nodes in the Enron data. For $k = 4$, most nodes during the majority of time steps had no dimension 1 persistence intervals which closed. Because the intervals do not close, this method essentially amounts to counting the 3-cliques in the k-neighborhood around v.

As a second attempt, a reverse scan filtration is used. This is similar to the forward filtration, but working inward. For example, for the graph in Figure 3.1, the yellow, orange, and red subgraphs would be added in that order, with the green vertex added last. This method yields more persistence intervals, indicating that it captures more structural information about the neighborhood surrounding a vertex. This is because initial steps are

poorly connected while the final steps are highly connected, making it more likely that holes will form early and be closed out by the end of the filtration. Even with this modified filtration, it is rare to see dimension 1 or higher persistent homology on network graphs.

The reason these two methods fail to provide interesting intervals is that the resulting filtrations are not granular enough. It is only feasible to apply this type of filtration for relatively small k ($\leq$ 5 was tested). The graphs tested have relatively small diameters ($\leq$13), so the local neighborhood graphs start to include most of the global network if k is taken too large. Any interesting structure useful for detecting anomalous activity from a vertex should be present in the neighborhood relatively local to that vertex. Finding the k^{th}-order neighborhood around a vertex is the same as finding the shortest path from the vertex to all other vertices, with a cutoff of k. On an unweighted graph, this is done with a breadth-first search, which has a run-time of $O(E + V)$. While tractable for small k, the expanding/contracting neighborhood approach did not meaningfully capture the structure of the local neighborhood surrounding a given vertex. In order to capture the interesting structure of the local neighborhood, longer, more granular filtrations are needed. This issue is addressed by the Weight Rank Clique Filtration.

3.3 Frequency-Based WRCF

For some applications, the edge ordering is important, but for the networks tested in this paper the resulting persistence landscapes behave similarly. Initially, edge weights were added by increasing order. The issue is that for each step in the filtration, the clique complex must be computed, which has already been shown to be a hard problem.

As shown in Section 3.1, the edge weights follow a distribution with a long tail. Adding low weight edges first means that for every step of the filtration, the clique complex must be computed over almost every edge. If instead, the high weight edges are added first,

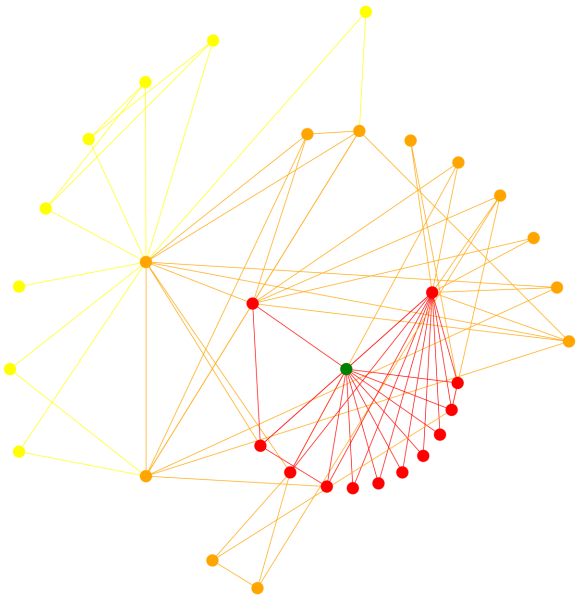


FIG. 3.1. The 1^{st},2^{nd}, and 3^{rd}-order neighborhoods of the green vertex, shown in red, orange, and yellow respectively.

most of the steps of the filtration correspond to relatively small portions of the graph, with only the last few steps including more edges. Changing the order of the WRCF to add high weight edges first provided approximately a 6000% speedup.

When computing persistent homology, the filtration length does not matter, only the number of simplices. However, longer filtrations require more time because more clique complexes must be found. Given a graph with total weight N, the maximum length of the filtration appears to be related to the inverse of the function for the integer sequence of triangle numbers. Specifically, it appears to follow $trinv(N)-1 = \lfloor(1+\sqrt{1+8N})/2\rfloor-1$ (Sloane 2009). This means that even for low weight graphs with many edges, the length of the filtration should remain relatively small, so the problem will remain tractable. In order to exploit the WRCF for anomaly detection, it is applied to the weighted k^{th}-order neighborhoods around each vertex. The resulting landscapes are used to detect anomalies.

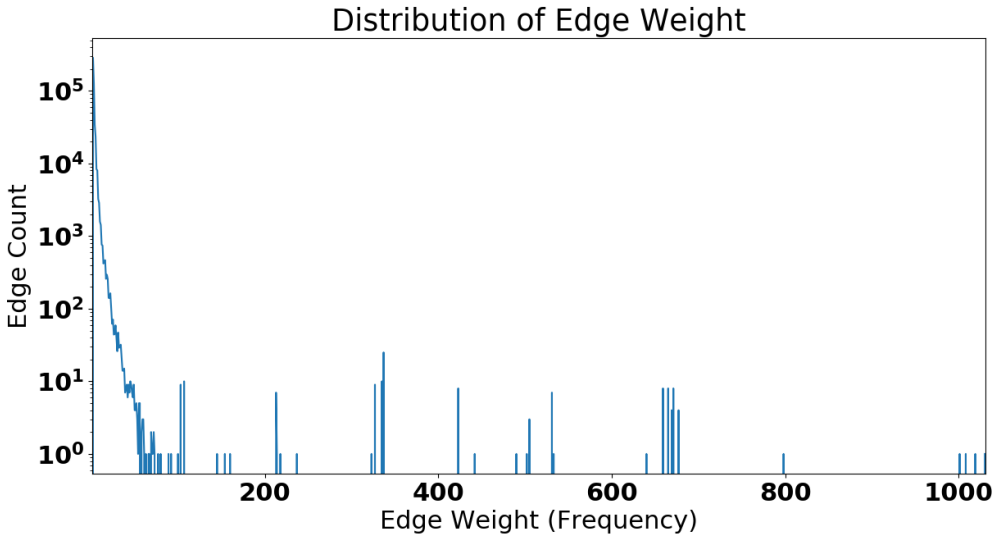


FIG. 3.2. The distribution of edge weight. Note the the y-axis is a log scale.

3.4 Anomaly Scoring

Given the persistence landscapes of network data over time, this section outlines how to generate an anomaly score which reliably detects suspicious local and global events. There are several approaches which may yield similar results. The following three scoring methods are the most promising.

Taking the L_2 distance from the landscape of a given week to the empty landscape provides an approximate measure of absolute activity in the local neighborhood, or an absolute anomaly score (AAS). While a natural first choice, this method tends to result in noisy scores which indicate interesting events with low accuracy.

AAS is a poor choice because it does not take into account past behavior. In order to account for relative changes of persistence landscapes over time, the lagged anomaly score (LAS) at time t is defined to be the distance between the persistence landscapes at t and $t-1$. This method is less noisy and takes past behavior into account. However, the LAS is sensitive to sudden drop offs in communication, which are easily detected by

simpler methods and not generally of interest. This issue can be compensated for via a combination of AAS and LAS.

Given the empty landscape and the landscapes at $t-1$ and t, the scaled lagged anomaly score ($sLAS$) is based on the distance between all three landscapes. More specfically, $sLAS_t = s_t \times LAS_t$, where

$$s_t = \begin{cases} AAS_t / AAS_{t-1} & AAS_t \leq AAS_{t-1} \\ AAS_{t-1} / AAS_t & AAS_t > AAS_{t-1} \end{cases}$$

Note $s_t \leq 1$. This scales the LAS relative to the change in the AAS. Little to no change in the AAS leads to a larger $sLAS$. This makes the $sLAS$ robust to changes in the amount of activity.

It is possible to compute the $sLAS$ for the entire network. The global $sLAS$ can be used for general event detection for the organization modeled by the network. Subtracting the global $sLAS$ from the local $sLAS$ results in an attributed LAS ($aLAS$), which further localizes and attributes anomalous behavior to specific vertices.

3.5 Software Used

Graph processing and filtration construction are done using custom scripts written in Python 3.6. This includes parsing the raw emails to graph format, and an implementation of the iterative clique complex algorithm from Section 2.1.5. Persistent homology calculations are carried out using JavaPlex (Tausz, Vejdemo-Johansson, & Adams 2014). Persistence landscape generation and distance measurement are done using an open-source persistence landscape toolbox (Bubenik & Dlotko 2015)(Dlotko 2015).

Chapter 4

EXPERIMENTS AND RESULTS

This section focuses on the application of AAS, LAS, and $sLAS$ to the Enron dataset. The publicly available records are mostly related to company wide events. For this reason, there are stronger and more verifiable results for the global anomaly detector. However, there are also interesting local anomalies surrounding key figures in the fall of Enron, coinciding with events in which they were involved. Taken together, these results demonstrate the viability of using persistent homology over graph data for anomaly detection.

4.1 Setup and Parameter Choice

All calculations are carried out on an Intel i7-3770 at 3.40 GHz with 16 GB of RAM running Ubuntu 16.04.

For all graphs, the persistence landscapes of the 1^{st}-persistent homology is used. The 1^{st}-persistent homology is the lowest degree which captures the idea of "holes" in the data, with the 0^{th}-persistent homology corresponding to connected components, for which there are much more efficient algorithms. Accordingly, clique complexes are only constructed up to and including 2-simplices (3-cliques). For the local construction around each vertex, the 3^{rd}-order neighborhood is used. These parameters are chosen to be small so that the problem remains tractable given limited resources. By parallelizing the construction of

the clique complexes and the reduction of the boundary matrices to Smith normal form, it should be possible to compute higher degree persistent homology over larger neighborhoods.

4.2 Global Event Detection

We begin by constructing the clique complexes for the weekly global graphs. The distribution of 1-simplices and 2-simplices is shown in Figure 4.1. The 1-simplices are edges in the graph, and approximate the total activity for a given week. The AAS and LAS for the global network are shown in Figures 4.2 and 4.3. To a limited extent, the spikes in both graphs indicate major events in the companies history. However, there are several high-scoring weeks which do not coincide with major events.

Figure 4.4 shows the $sLAS$ of the entire network over time. There are many weeks without any activity at the beginning of the dataset, which is why the score is zero. The graph shows successful anomaly detection, with high scores that coincide with major events leading to the failure of Enron. The records of these events are publicly available (Times 2006) (Marks 2008). The top detections for the $sLAS$ of the network are shown in Table 4.1, listed in order of decreasing score.

As expected, the high global anomaly score indicates major events. All of these events caused substantial and observable changes in communication across the company's email network. This demonstrates that the global $sLAS$ can be used to detect network wide events.

Table 4.1. Highest Scoring $sLAS$ Events

Week #	Date	Event
157	11/4/2001	The week following SEC upgrade of inquiry to formal investigation.
155	10/21/2001	The week following SEC starting an informal probe.
158	11/11/01	The week after admitting $596 million in losses to SEC. Start of bankruptcy.
128	4/15/2001	The week of a highly publicized controversial analyst call with CEO Jeffery Skilling.

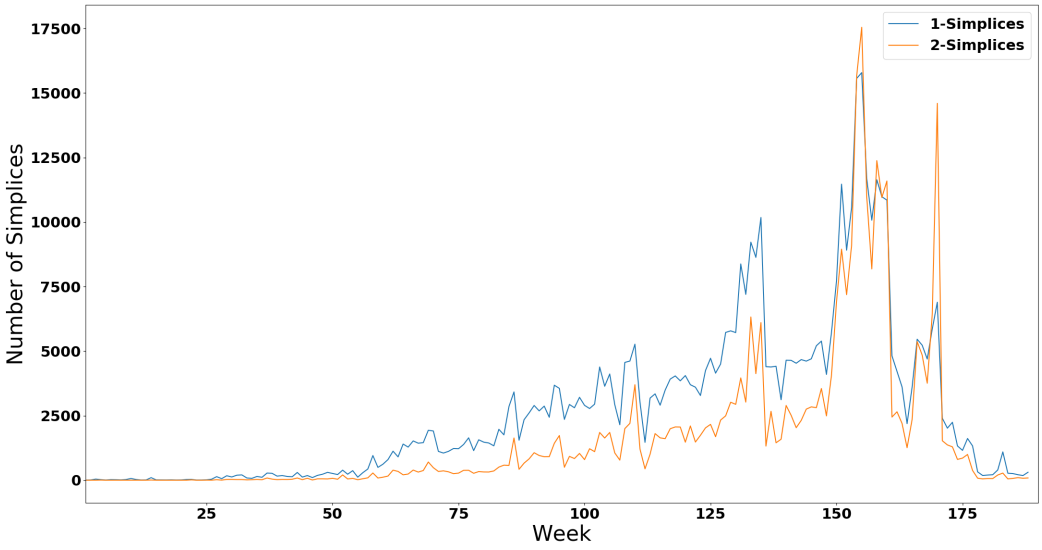


FIG. 4.1. The global weekly distribution of simplices. Recall that calculating persistent homology is $O(m^3)$, where m is the number of simplices.

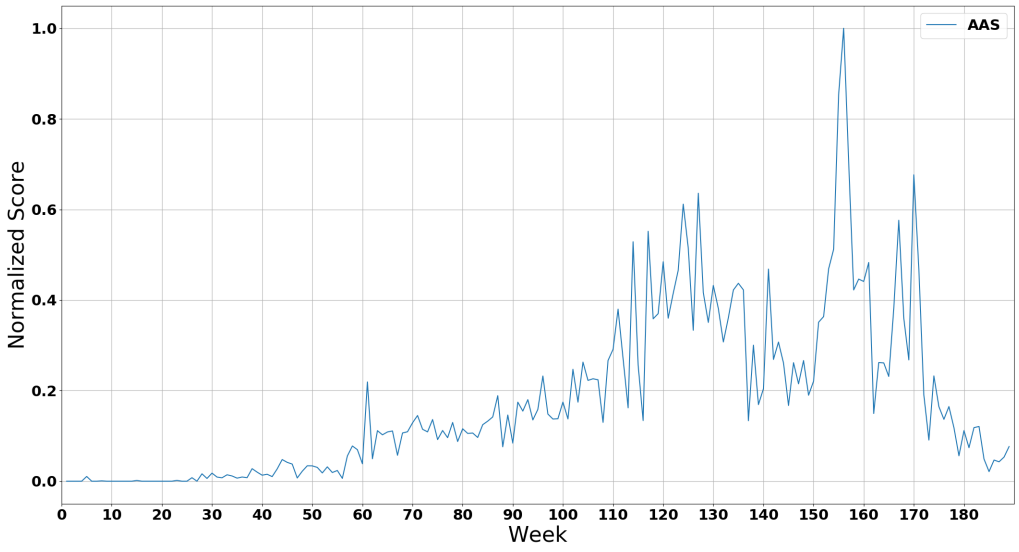


FIG. 4.2. The global AAS over time. Distance from the empty persistence landscape also approximates total activity.

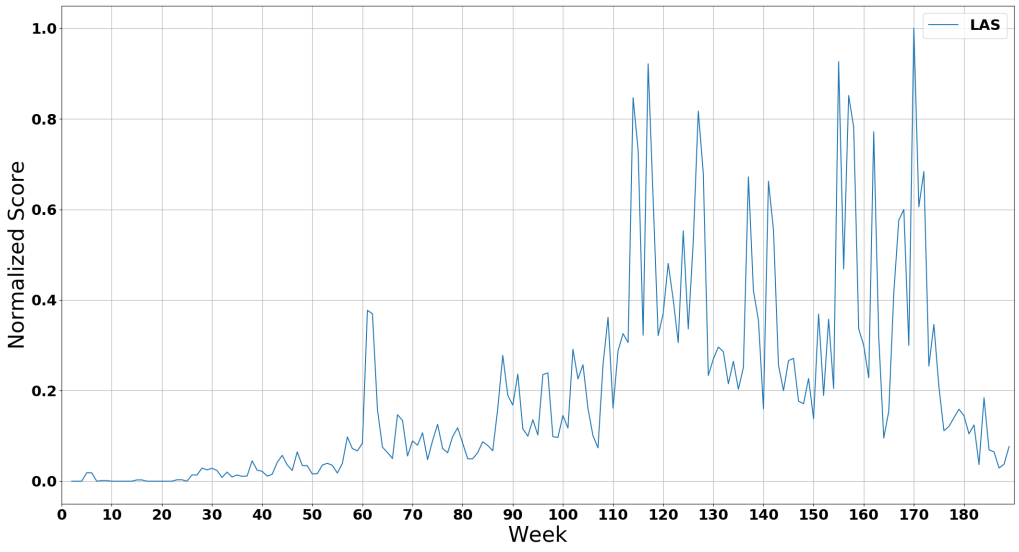


FIG. 4.3. The global LAS over time. Most spikes correspond to major events, but there is still influence from changes in overall amount of activity.

4.3 Local Suspicious Behavior Detection

While interesting and useful for certain applications, global event detection cannot be used to identify suspicious nodes in a network. In order to accomplish this, k^{th}-order neighborhoods are utilized as outlined in Section 3.2.

Several individuals have been convicted of criminal charges in connection with the fall of Enron, one of which is CEO Kenneth Lay. The three scores for Mr. Lay's 3^{rd}-order neighborhood are shown in Figures 4.5 and 4.6. While the local $sLAS$ is less noisy, it tends to move with the global $sLAS$, making it difficult to attribute an anomalous behavior to the vertex.

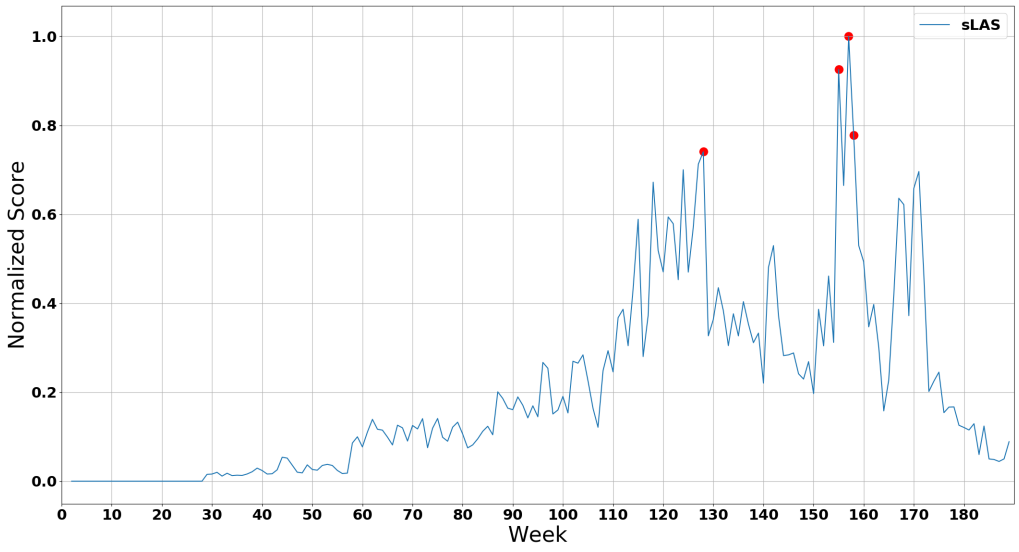


FIG. 4.4. The global $sLAS$ over time. Major spikes correspond to significant events in the Enron timeline.

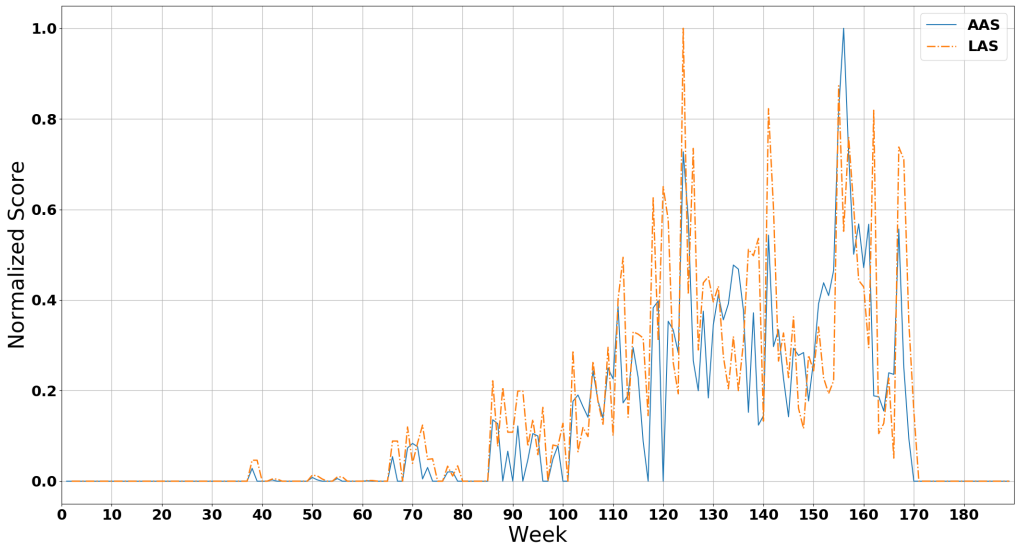


FIG. 4.5. The AAS and LAS for the 3^{rd}-order neighborhood around Kenneth Lay. While the high scores coincide with some events, there are false positives.

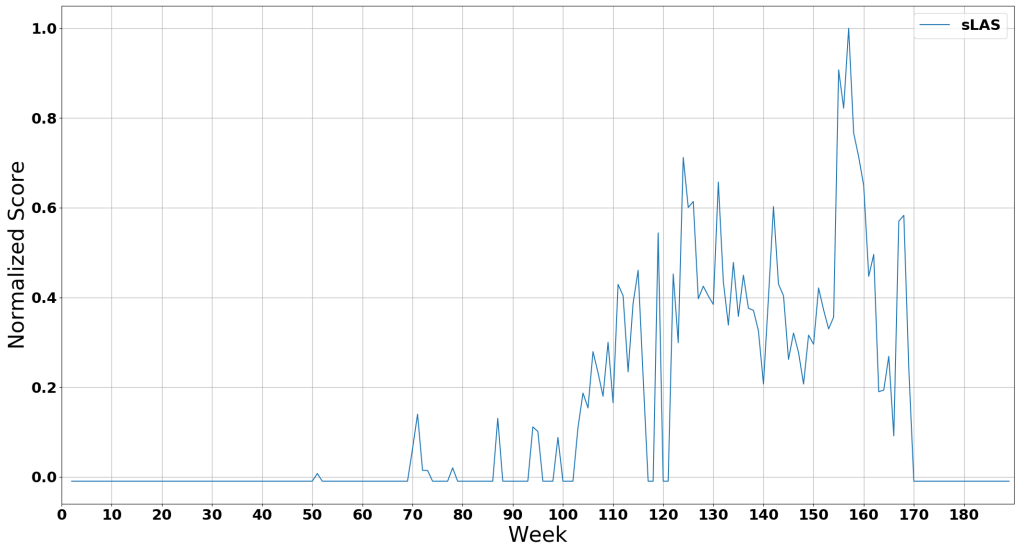


FIG. 4.6. This score alone is not enough to attribute anomalies. There is too much influence from trends in the global network.

The $aLAS$, which is shown in Figure 4.7, helps to address this attribution issue. An $aLAS > 0$ indicates that an individual vertex exhibits anomalous behaviour relative to global trends. Scores < 0 do not have a meaningful interpretation.

There is limited ground truth information about when specific individuals at Enron committed crimes. However, there is a record of Kenneth Lay organizing a secret meeting with Arnold Schwarzenegger and Michael Milken on 5/17/2001. The highest score on week 131 indicates anomalous behavior starting 5/6/2001, the week preceding the secret meeting. The other major spikes appear to only coincide loosely with events involving Kenneth Lay. Examining the local graphs of Kenneth Lay, as shown in Figure 4.9, verifies that his behavior changes significantly from week 130 to week 131. The $aLAS$ for CEO Jeffery Skilling is shown in Figure 4.8. Again, only the largest spike is of interest, coinciding with Mr. Skilling's sale of a large portion of his Enron shares under suspicious

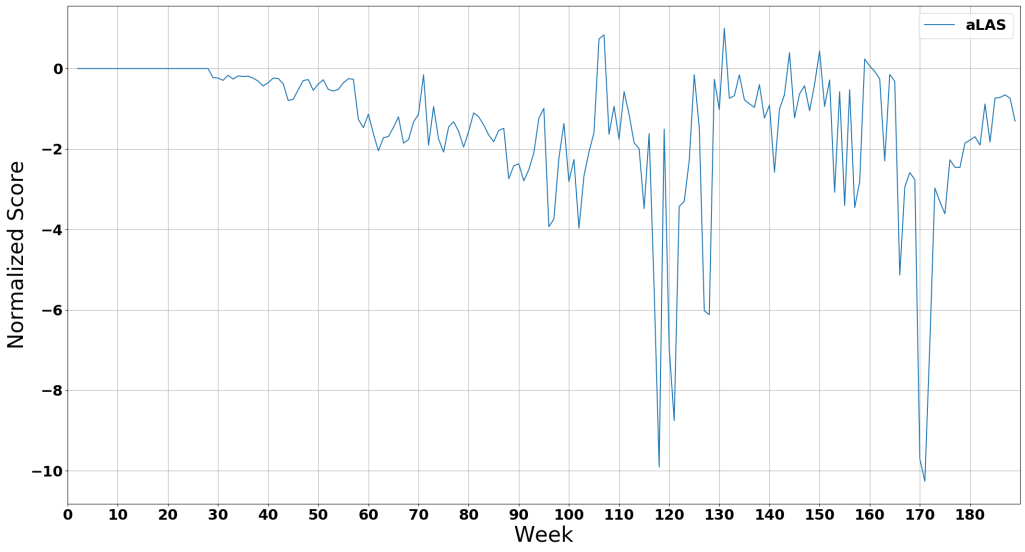


FIG. 4.7. Anomalous behaviour specifically attributed to Kenneth Lay.

circumstances.

The above examples show that the $aLAS$ calculated over local subgraphs can detect and attribute anomalies to a limited extent. With better ground truth data, it should be possible to verify and refine the scoring method.

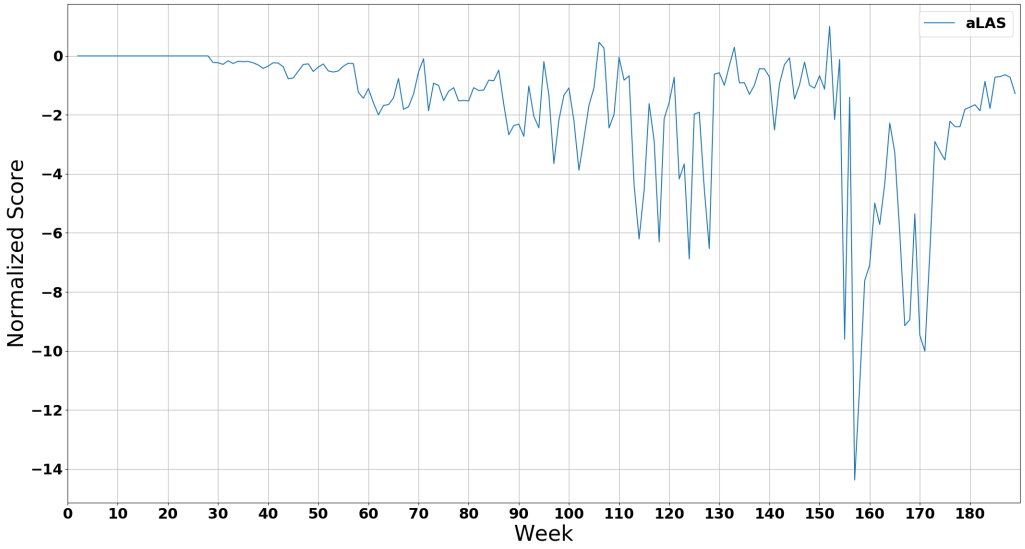


FIG. 4.8. Anomalous behaviour specifically attributed to Jeffery Skilling.

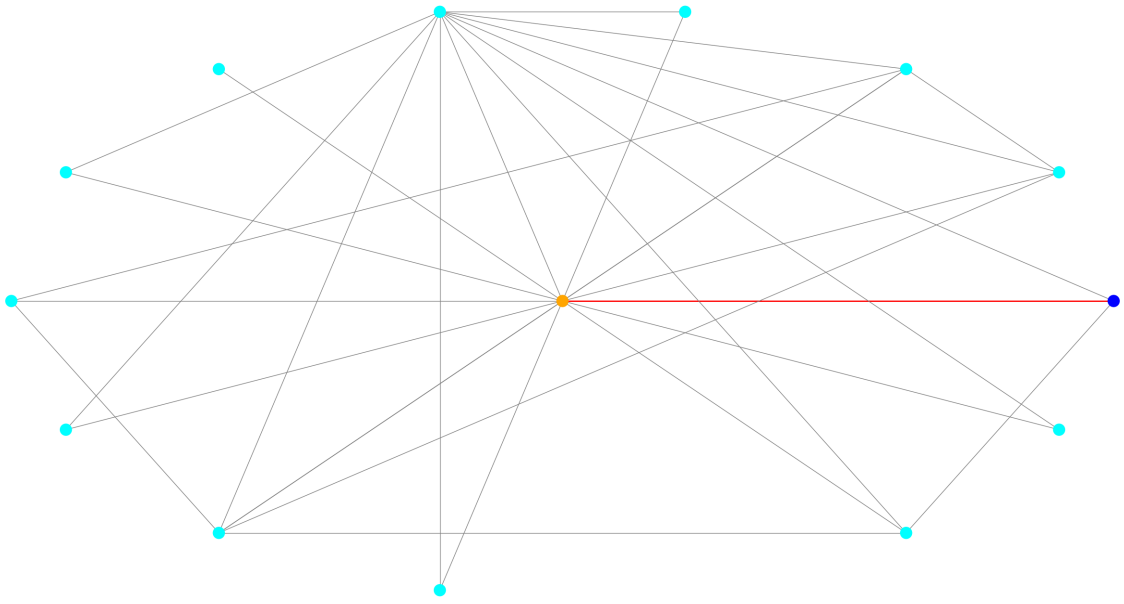


FIG. 4.9. The 1^{st}-order graph of Kenneth Lay (orange vertex) at week 130. Any connections made in week 131 which were also made in week 130 are colored red. Note there is only one red edge.

Chapter 5

FUTURE WORK AND CONCLUSION

This section begins by addressing two tangential problems which have arisen while developing the methods outlined above. The first is concerned with the space complexity of computing a WRCF; the second with the time complexity of constructing a clique complex.

The primary goal of this paper is to show that persistent homology can be applied directly to network data with minimal overhead. For this reason, when presented with a design choice, the simplest option is preferred. However, there are several alternative methods which may provide better results. These methods are outlined in Section 5.2.

5.1 Open Problems

5.1.1 WRCF Size

Section 3.3 mentions that the length of a WRCF appears to be $trinv(N) - 1$, where N is the total weight of the graph. Each step in the filtration corresponds to a unique edge weight in the graph. Clearly, the filtration will be longest if every edge has a unique weight. More formally, we are looking for the number of terms in the partition of N with the greatest number of distinct terms. This sequence is thought to be given by OEIS A003056 = $trinv(N) - 1$, though no proof is available (Sloane 2019). A formal proof would help place bounds on the complexity of computing a WRCF for a given graph.

5.1.2 Clique Construction Bounds

There are currently no known bounds on the algorithms used to compute clique complexes. This is a difficult problem mainly due to the sparsity of cliques in most graphs (Zomorodian 2010). However, this computation is a critical step in the construction of most filtrations. While it may be difficult to provide bounds for arbitrary graphs, it may be possible given prior knowledge about the graph structure. Most graphs of interest correspond to scale-free networks. Therefore, finding the bounds of clique construction over scale-free graphs is of particular interest.

5.2 Additional Methods

This section covers additional methods and techniques which appear promising, but were not pursued due to time constraints. These techniques may provide improved anomaly detection in general and for certain applications.

5.2.1 Filtrations

The WRCF used in Section 4 groups edges by unique weight. As shown in Figure 3.2, edge weight follows a very long tail distribution. This means that edges are added unevenly to the filtration. In order to create more consistent filtrations, it may be beneficial to bin the edge weights non-linearly when constructing the filtration.

In the experiments run, edge weight is the total communication between a pair of vertices. While informative, this method may ignore important information about the network. Instead, using the net frequency of communication between nodes should provide a more complete picture. This could be done with minimal overhead by assigning an arbitrary direction to each edge. Messages which follow the edge direction would increment the weight, while messages going in the opposite direction would decrement it. This is a more

application-dependent approach and may provide improved detection.

It may be possible to create a true locality statistic using a variation of the WRCF. Given a subgraph $S \subseteq G$ which is the k^{th}-order neighborhood around a vertex v, add edges and vertices to the filtration in order of increasing distance from v, where $d(v, u)$ is the shortest path from v to u. The persistence landscapes generated will likely satisfy the conditions of a locality statistic, which means well-developed scan statistic techniques can be applied.

5.2.2 Supervised Methods

The methods in this paper are unsupervised, so no training examples are needed. Given an application where supervised learning is possible, persistence landscapes can be used as features. Clustering methods have shown particular promise in other applications (Bubenik & Dlotko 2015). Supervised methods are likely a good fit for security applications, where there are training examples of known attacks. Given enough information, it may be possible to train a classifier on a sample network and apply a modified version to a target network.

5.3 Conclusion

This paper has developed a graph invariant based on persistent homology derived directly from network data. Applied globally, this invariant can be used to detect network wide events. It can also be applied locally to detect nodes which are behaving suspiciously.

With further refinement and additional data, this invariant can be used to develop a real-time anomaly detection system. Specifically, there are two areas which would benefit from further research. Parallelizing clique complex and homology calculations will help the approach scale. Adjusting the methods of filtration should improve local performance.

Topological data analysis is a relatively young field with many applications still being developed. This research provides a proof of concept, and a starting point for future investigation.

REFERENCES

[Bruillard, Nowak, & Purvine 2016] Bruillard, P.; Nowak, K.; and Purvine, E. 2016. Anomaly detection using persistent homology. In *2016 Cybersecurity Symposium (CYBERSEC)*, 7–12. IEEE.

[Bubenik & Dlotko 2015] Bubenik, P., and Dlotko, P. 2015. A persistence landscapes toolbox for topological statistics. *CoRR* abs/1501.00179.

[Bubenik 2015] Bubenik, P. 2015. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research* 16(1):77–102.

[Carlsson 2009] Carlsson, G. 2009. Topology and data. *Bulletin of the American Mathematical Society* 46(2):255–308.

[Chandola, Banerjee, & Kumar 2009] Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41(3):15.

[Chazal *et al.* 2013] Chazal, F.; Fasy, B. T.; Lecci, F.; Rinaldo, A.; Singh, A.; and Wasserman, L. 2013. On the bootstrap for persistence diagrams and landscapes. *arXiv preprint arXiv:1311.0376*.

[Dlotko 2015] Dlotko, P. 2015. The persistence landscape toolbox. `https://www.math.upenn.edu/~dlotko/persistenceLandscape.html`.

[Edelsbrunner & Harer 2008] Edelsbrunner, H., and Harer, J. 2008. Persistent homology-a survey. *Contemporary mathematics* 453:257–282.

[Enron 2015] Enron. 2015. Enron email dataset. `http://www.cs.cmu.edu/~enron/enron_mail_20150507.tar.gz`.

[Foundation 2019] Foundation, W. 2019. Cech complex. https://en.wikipedia.org/wiki/Œech_complex#/media/File:Cech-example.png.

[Ghrist 2008] Ghrist, R. 2008. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society* 45(1):61–75.

[Goyal & Ferrara 2018] Goyal, P., and Ferrara, E. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151:78–94.

[Marks 2008] Marks, R. 2008. Enron timeline. http://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html.

[Petri *et al.* 2013] Petri, G.; Scolamiero, M.; Donato, I.; and Vaccarino, F. 2013. Topological strata of weighted complex networks. *PloS one* 8(6):e66506.

[Priebe *et al.* 2005] Priebe, C. E.; Conroy, J. M.; Marchette, D. J.; and Park, Y. 2005. Scan statistics on enron graphs. *Computational and Mathematical Organization Theory* 11(3):229247.

[Sloane 2009] Sloane, N. J. A. 2009. https://oeis.org/A002024.

[Sloane 2019] Sloane, N. J. A. 2019. https://oeis.org/A003056.

[Tausz, Vejdemo-Johansson, & Adams 2014] Tausz, A.; Vejdemo-Johansson, M.; and Adams, H. 2014. JavaPlex: A research software package for persistent (co)homology. In Hong, H., and Yap, C., eds., *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, 129–136.

[Times 2006] Times, N. Y. 2006. Timeline: A chronology of enron corp. https://www.nytimes.com/2006/01/18/business/worldbusiness/timeline-a-chronology-of-enron-corp.html.

[Zomorodian & Carlsson 2004] Zomorodian, A., and Carlsson, G. 2004. Computing persistent homology. *Proceedings of the twentieth annual symposium on Computational geometry - SCG 04*.

[Zomorodian 2010] Zomorodian, A. 2010. Fast construction of the vietoris-rips complex. *Computers & Graphics* 34(3):263–271.