HYPA: Efficient Detection of Path Anomalies in Time Series Data on Networks

Timothy LaRock* Vahan Nanumyan† Ingo Scholtes‡§ Giona Casiraghi†
Tina Eliassi-Rad*¶ Frank Schweitzer†

Abstract

The unsupervised detection of anomalies in time series data has important applications in user behavioral modeling, fraud detection, and cybersecurity. Anomaly detection has, in fact, been extensively studied in categorical sequences. However, we often have access to time series data that represent *paths* through networks. Examples include transaction sequences in financial networks, click streams of users in networks of cross-referenced documents, or travel itineraries in transportation networks. To reliably detect anomalies, we must account for the fact that such data contain a large number of independent observations of paths constrained by a graph topology. Moreover, the heterogeneity of real systems rules out frequency-based anomaly detection techniques, which do not account for highly skewed edge and degree statistics. To address this problem, we introduce HYPA, a novel framework for the unsupervised detection of anomalies in large corpora of variable-length temporal paths in a graph. HYPA provides an efficient analytical method to detect paths with anomalous frequencies that result from nodes being traversed in unexpected chronological order.

1 Introduction

Anomaly detection refers to the problem of finding "patterns in data that do not conform to a well-defined notion of normal behavior" [14]. The importance of anomaly detection techniques rests on the fact that anomalous patterns may carry valuable meaning. Examples include anomalous usage or traffic patterns used to detect cyberattacks, anomalous sensor readings that may identify imminent faults in technical systems, or anomalous transaction patterns used to detect fraud and compliance violations in financial systems. In order to assess which data represent "anomalies", we must define what we consider "normal" behavior in the particular system under study. Given this baseline of "normal" behavior, we need methods to efficiently assess which patterns in the data exhibit deviations from this baseline. Finally, we need techniques to argue which of those observed deviant patterns are *significant* given the fluctuations and randomness contained in data.

Anomaly detection has been studied extensively for general categorical sequence data. However, we are often confronted with time series data capturing *paths through networks*. Such data have distinctive characteristics. Different from general categorical sequences, an underlying graph topology constrains which paths, i.e., sequences of node traversals, can possibly occur. Moreover, the graphs in which paths are observed often exhibit strong heterogeneities, e.g., heavily skewed node degree distributions or heterogeneous edge statistics.[1] These heterogeneities invalidate frequency-based anomaly detection techniques that do not account for the fact that in real systems, some paths are more likely to be observed at random than others (see Fig. 1).

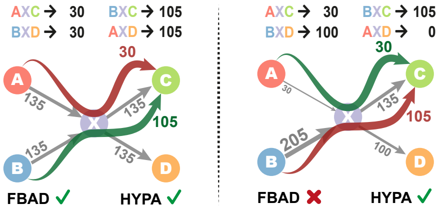

Figure 1: Frequency-based anomaly detection (FBAD) can be used to identify ground truth under- (red) and over-represented (green) paths in graph with homogenous edge statistics (left), but fails to identify anomalies in data with heterogeneous edge statistics (right). Our proposed method HYPA succeeds in both scenarios.

Closing this gap, we consider the problem of detecting *anomalous paths* through graphs based on data capturing sequences of node traversals. Our definition of *anomalous paths* rests on a memoryless baseline model, which assumes that the chronological order of node traversals is determined by the graph topology and the edge traversal statistics. We develop HYPA, an algorithm for detecting paths with unexpected temporal traversal patterns.

This problem is of practical relevance in a number of scenarios. For example, in graphs representing transportation systems, such as passenger flights, we can study trajectories generated by passengers navigating through the network. Here anomalous paths convey

*Network Science Institute, Northeastern University
†Chair of Systems Design, ETH Zürich
‡Chair of Data Analytics, University of Wuppertal
§Data Analytics Group, IfI, University of Zürich
¶Khoury College of Computer Sciences, Northeastern University

[1] In this paper, we use the term *heterogeneous* in reference to statistical distributions of edge and path frequencies in networks. We are *not* working with *heterogeneous graphs*, where nodes and edges in the same graph may have different types (e.g. [30]).

information about the role of airports in routing people through the system.

Our main contributions are:

(i) We introduce *path anomalies*, paths through a graph that are traversed significantly more or less often than expected under a null model. We show that the problem of detecting length k path anomalies can be reduced to detecting *anomalous edges* in a kth-order De Bruijn graph.

(ii) We introduce HYPA, an algorithm for detecting path anomalies. HYPA finds paths that occur significantly more or less often than expected at random, leveraging an analytically tractable statistical model of random weighted De Bruijn graphs to derive closed-form expressions for the cumulative weight distribution of paths of any length k.

(iii) We test HYPA in empirical data representing paths through transportation systems, validating detected anomalies with geographical information.

The remainder of this paper is organized as follows. In the next section, we discuss related work and introduce relevant background that forms the basis of our method. In Section 3, we formally define path anomalies, walk through an illustrative example, and introduce HYPA. In Section 4, we validate HYPA in synthetic data, before applying it to analyze a dataset of passenger trips through an airport network.

2 Related Work and Background

In this section we summarize related work on anomaly detection and sequential pattern mining and provide background on the higher-order graph models and statistical graph ensembles underlying HYPA.

2.1 Related Work Considering the large body of research on anomaly detection in time series data [21], and keeping in mind the focus of this paper, we limit our review to related work on (i) anomaly detection in discrete sequences, (ii) sequential pattern mining, and (iii) graph-based anomaly detection. Since we are concerned with the unsupervised detection of path anomalies, we further exclude (semi-)supervised and reinforcement learning techniques.

Anomaly Detection in Sequence Data. Following [16, 15], anomaly detection techniques for discrete sequences fall into different categories that address fundamentally different application scenarios. Sequence-based anomaly detection assumes that we are given a set $\mathbf{S} = \{s_1, s_2, \ldots, s_n\}$ of sequences $s_i = (x_j)_{j=1,\ldots,l_i}$ over a discrete alphabet Σ, possibly with variable lengths l_i. Anomalous instances s_i in $\mathbf{S}$ are then detected. For example, each sequence may be assigned an anomaly score, then ranked from most to least anomalous by the magnitude of this score. Different approaches have been used to establish a random baseline against which sequences are defined as "anomalous". Some methods have used (hidden) Markov chain models, e.g., to detect (groups of) sequences which show significant differences in terms of state transition probabilities [40, 26, 29, 4]. Other methods use nearest-neighbours algorithms [31] or distance measures [42] to quantify how any given sequence s_j differs from other instances in $\mathbf{S}$. Adopting a collective definition of anomalies [14], a third class of methods is based on hypothesis testing techniques to detect outliers in the distribution of features of sequences [41, 28, 5]. Our problem setting is different because we are interested in discovering patterns in an underlying network structure, not in marking individual sequences as anomalous.

Sequential pattern mining. A common feature of the methods outlined above is that they focus on anomalies at the level of a whole sequence s_i within $\mathbf{S}$. Addressing a different problem, some methods instead attempt to find anomalous *patterns* or *subsequences* within a long sequence $S = (x_i)_{i=1,\ldots,n}$ [15]. This is called sequential pattern mining, where the goal is to develop algorithms that quickly find the most frequent subsequences in large sequence data [1, 24, 18, 38]. Some work addresses this problem based on statistical methods, e.g., using Markov modeling techniques [22, 36, 45, 10, 44, 34, 23], hypothesis testing [39, 6, 43], or information-theoretic methods to detect "surprising" subsequences [25, 13, 7]. Applications include the detection of common patterns in user trajectories [44, 35], testing hypotheses about generative processes of trajectory data [39], or finding clusters in sequence data [10, 34].

Temporal Anomaly Detection in Graphs. The problem motivating our method is different from those described so far, mainly due to the fact that these methods make no assumptions about the relational structure of the data, while we study sequential data capturing *paths in a (weighted and directed) graph topology.* This aligns our work more closely with anomaly detection techniques for temporal graph data that have been developed in the graph mining community [33, 3]. As summarized in [3], temporal anomaly detection discovers change events [2] or cluster structures in evolving graphs [8, 9, 34].

Different from these problems, our method uses a set $\mathbf{S}$ of sequences to identify paths through a graph that are traversed more or less often than expected. Hence, rather than making statements about anomalous instances in $\mathbf{S}$, we use collective statistical information in $\mathbf{S}$ to identify paths through the graph that are traversed with *anomalous frequencies.*

2.2 Background In this section, we provide definitions necessary to the formulation of path anomaly de-

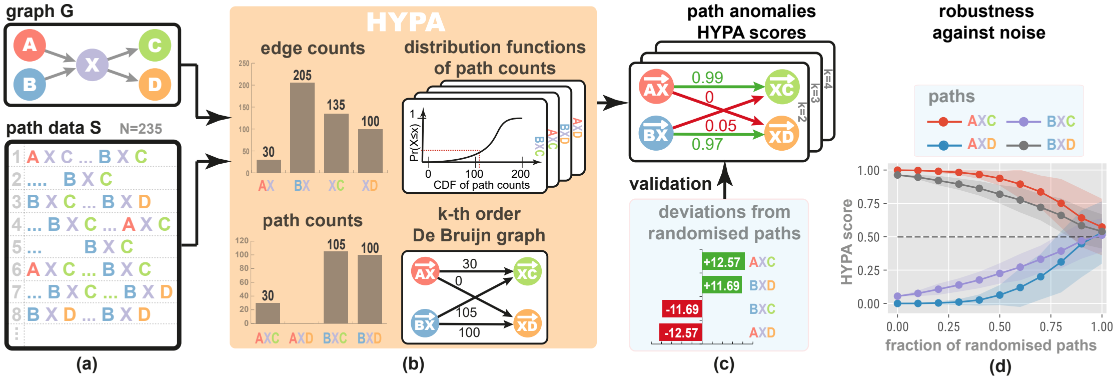

Figure 2: Example of path data **S** observed in a graph G illustrates path anomaly detection with HYPA (focusing on k=2). Given a set of sequences traversing nodes A, B, X, C, and D in a graph (a), HYPA uses higher-order De Bruijn graphs to derive closed-form expressions for the cumulative distribution function of all possible paths in the graph (b). HYPA computes HYPA-scores (c) that allow reliable detection of over- and under-represented paths, even in situations where the least frequent path (AXC) is over-represented, while the most frequent path (BXC) is under-represented. Progressive randomization of the data gradually levels HYPA-scores (d), translating to a decreasing confidence in detected anomalies.

tection and our solution.

We reduce the problem of detecting anomalous *paths* in a (first-order) graph to detection of anomalous *edges* in *higher-order* graph models that resemble De Bruijn graphs [17]. Similar to [37], we define a higher-order De Bruijn graph model of paths as:

Definition 2.1. (**k-th order De Bruijn graph model**) *For a given graph $G = (V, E)$ and $k \in \mathbb{N}$ we define a k-th order De Bruijn graph of paths in G as a graph $G^k = (V^k, E^k)$, where (i) each node $\vec{v} := \overrightarrow{v_0 v_1 \ldots v_{k-1}} \in V^k$ is a path of length*[2] *$k-1$ in G, and (ii) $(\vec{v}, \vec{w}) \in E^k$ iff $v_{i+1} = w_i$ for $i = 0, \ldots, k-2$.*

This definition has several implications. First, any two nodes $\vec{v}$ and $\vec{w}$ connected by an edge in a k-th order graph G^k represent 2 paths of length $k-1$ that overlap in exactly $k-1$ out of k nodes. Since paths in a graph are transitive, each edge $(\vec{v}, \vec{w})$ in G^k represents a path of length k in graph G. This implies that the graph G itself is a first-order De Bruijn graph of paths of length one (i.e., edges) in G. We can see De Bruijn graphs as a generalization of standard, first-order graphs to higher-order models of paths of length k, where any path of length q in G^k translates to a path of length $k+q-1$ in G. We iteratively construct De Bruijn graph models of order k by means of a line graph transformation on the $k-1$st order model.

This representation is powerful because it allows us to encode the frequencies of paths of length k through a first-order graph to the weights of edges in a k-th order De Bruijn graph. This can be seen in the illustration of a De Bruijn graph with order $k = 2$ in Fig. 2, where nodes represent paths of length $k-1 = 1$ that overlap in $k-1 = 1$ nodes (i.e., edges in G), while edges represent all paths of length $k = 2$.

This projection of paths allows us to reduce the problem of detecting paths of length k that exhibit anomalous frequencies to the problem of detecting anomalous edge weights in a k-th order De Bruijn graph. To understand which edge weights exhibit "anomalies", we need a null model that provides a baseline against which we compare the observed weights. For this comparison, we need to generate randomized configurations of the path data that selectively destroy only the patterns that we are interested in while preserving all other statistics. Since we can project the path data to the edges of a directed and weighted graph, we can address this problem by employing *statistical graph ensembles*, which randomize certain aspects of a graph (i.e., the weights of edges or the topology itself) while preserving other characteristics. Examples include models that randomize the topology of a graph while preserving the (expected) number of edges [20], as well as combinatorial models that preserve the degrees of nodes [32].

An analytically tractable formulation of such a model for directed and weighted graphs was recently proposed in [11]. It treats the random generation of weighted graphs as an urn problem, where random edges are drawn without replacement from a population of multi-edges connecting different pairs of nodes. Through this formulation, the probability of generating edges with specific weights can be calculated based on the multivariate hypergeometric distribution. This formulation can be used to detect anomalous edges in social networks [12]. However, no analytically tractable null models have been proposed that account for the distinctive characteristics of De Bruijn graphs, i.e., the fact that a directed edge between two nodes in a k-th order De Bruijn graph can only exist if the corresponding path exists in the underlying graph. Closing this gap, we develop a method to detect path anomalies based on statistical ensembles of k-th order De Bruijn graphs.

3 Higher-order Hyper-geometric path anomaly detection

We now define the problem of *path anomaly detection*, illustrate it in an example, and propose our solution.

[2] We assume path length is the number of edges traversed.

Definition 3.1. (**Path Anomaly Detection**) *Let $G = (V, E)$ be a directed graph and $\mathbf{S}$ a set of n sequences s_i, where each sequence $s_i = v_0 v_1 \dots v_{l_i}$ is a* path *of arbitrary length l_i in G, i.e. $v_j \in V$ for $j \in [0, \dots, l_i]$ and $(v_j, v_{j+1}) \in E$ for $j \in [0, \dots, l_i - 1]$. For $k > 1$, identify all paths $\vec{p} = \overrightarrow{v_0 \dots v_k}$ of length k in G whose frequencies (including as subpaths) in $\mathbf{S}$ significantly deviate from the frequencies expected in a* $(k-1)$-order *model of paths in G.*

Unlike sequence-based anomaly detection [15], we are not interested in assigning an *anomaly score* to each sequence in $\mathbf{S}$. Instead we use the instances in $\mathbf{S}$ to identify paths through the graph that exhibit anomalous frequencies compared to a null model, discovering the paths in G that are traversed in unexpected ways given the underlying weighted network structure. To complete our definition of *anomalies*, we define a generative *null model* for paths that builds on definition 2.1, which we use to establish a baseline against which we detect anomalies.

Definition 3.2. (**k-th order model of paths**) *For a graph G let $G^k = (V^k, E^k)$ be a k-th order De Bruijn graph of paths in G (cf. Def. 2.1). For each edge $e := (\overrightarrow{v_0 \dots v_{k-1}}, \overrightarrow{v_1 \dots v_k}) \in E^k$ let the weight $f(e)$ be the frequency of subpath $\overrightarrow{v_0 \dots v_k}$ in $\mathbf{S}$. Let $\mathbf{T}^k$ be the transition matrix of an edge-weighted random walk on G^k, i.e., $\mathbf{T}^k_{\vec{v}\vec{w}} := \frac{f(\vec{v}, \vec{w})}{\sum_{\vec{x} \in V^k} f(\vec{v}, \vec{x})}$. For a path $\vec{p} = \overrightarrow{v_0 v_1 \dots v_l}$ with $l \geq k$ the* k-th order model of paths *generates $\vec{p}$ with probability $\prod_{i=k}^{l} \mathbf{T}^k_{\overrightarrow{v_{i-k} \dots v_{i-1}} \overrightarrow{v_{i-k+1} \dots v_i}}$.*

This model generates paths of length l by performing $l - k + 1$ random walk steps in a k-th order De Bruijn graph. We can use the model to generate random paths of length $l \geq k$ that respect (i) the topology of the underlying graph G, and (ii) the frequencies of paths of length k observed in $\mathbf{S}$.

Our definition of path anomalies of length k is based on a null model of order $k - 1$. For $k = 2$, the null model of order $(k - 1) = 1$ is simply an edge-weighted random walk on the graph G. In this case, the sequence of nodes traversed by paths is *Markovian*, i.e., the node v_{i+1} on a path only depends on the current node v_i and the graph topology. Apart from the topology, the model accounts for the frequencies at which paths in $\mathbf{S}$ traverse edges in G. That is, if an edge (b, c) is traversed more often than (b, d) we expect path $\overrightarrow{abc}$ to occur more often than $\overrightarrow{abd}$. For $k > 2$, the null model corresponds to an edge-weighted random walk on a De Bruijn graph of order $(k-1) > 1$, where weighted edges capture the frequencies of subpaths of length $k - 1$ in $\mathbf{S}$. This approach to generating a null model is key to disentangling path anomalies that unfold at different lengths: For any given length k, we can exclusively detect those path anomalies that do not trivially result from anomalous path frequencies at shorter lengths. In other words, to answer the question whether a *specific* path $\overrightarrow{abcd}$ of length $k = 3$ is observed more or less often than expected, we discount for any anomalies of shorter paths $\overrightarrow{abc}$ and $\overrightarrow{bcd}$ contained in $\overrightarrow{abcd}$.

3.1 Illustrative Example

A simple example to illustrate the path anomaly detection problem for $k = 2$ is shown in Fig. 2, which gives a high level overview of HYPA. Limiting our focus to paths that traverse nodes A, B, X, C, and D, we consider a set $\mathbf{S}$ that contains 235 (sub)paths of length two. We observe strong heterogeneities in the path frequencies, where the most frequent path $\overrightarrow{BXC}$ occurs 105 times, while the least frequent observed path $\overrightarrow{AXC}$ occurs only 30 times.

Assume we want to detect for which paths of length $k = 2$ the frequencies deviate from the expectation in a first-order null model. If all paths were expected to occur with similar frequency (e.g. if observed frequency was drawn from a normal distribution), we could directly answer this question based on the observed distribution of path frequencies (cf. Fig. 1). Such an approach would trivially detect that path $\overrightarrow{AXC}$ occurs more often than expected while path $\overrightarrow{BXC}$ occurs less often than expected. However, the edge frequencies in our toy example show strong heterogeneities; for example, edge (B, X) is traversed about seven times more often than edge (A, X). If we account for this heterogeneity of edges (i.e., paths of length $k - 1 = 1$), the question of which paths of length $k = 2$ exhibit statistically significant deviations becomes non-trivial. In particular, the same observed frequency could be"normal" (i.e., expected) for one path, a significant over-representation for another, and an under-representation for a third. Whether the frequency of a path is anomalous based on definition 3.1 can not be determined by direct comparison with the overall frequency distribution alone.

We can address this problem by randomizing the data using random walk simulations on the first-order model and preserving the distribution of path lengths. We then count the average frequency of each path of length 2 across many simulations. A comparison of observed vs. average frequencies of paths then indicates which paths exhibit deviations from the random baseline. In Fig. 2c, we report the average of 100 such simulation runs, which indicate that paths $\overrightarrow{AXC}$ and $\overrightarrow{BXD}$ occur *more* often than expected, while paths $\overrightarrow{BXC}$ and $\overrightarrow{AXD}$ occur *less* often than expected. This simple example highlights an important problem: due to the heterogeneous frequency of edges, paths that occur with the smallest frequency ($\overrightarrow{AXC}$) can be over-represented, while paths that occur with the highest frequency ($\overrightarrow{BXD}$) can be under-represented.

This observation rules out collective anomaly detection techniques that assess anomalies based on a *single* frequency distribution. We must instead consider the joint distribution of frequencies under the null model for each possible path and each length k separately. While a simulation-based approach is possible in principle, the combinatorial growth of the required computational effort for large systems is prohibitive. Moreover, such simulations leave open the question of whether the observed deviations in the data indicate a significant pattern or are likely due to chance.

Projecting paths of length k onto edges in a k-dimensional De Bruijn graph, we use closed-form expressions for the cumulative distribution function of path frequencies under the $(k-1)$-order null model for each path individually (see Fig. 2b). This enables us to analytically calculate *HYPA-scores*, which, for each path $\vec{p}$, represent the likelihood that a null model generates realizations where frequencies of $\vec{p}$ are larger (or smaller) than in the data. The calculated scores can then be used to detect path anomalies at various levels of significance without expensive simulations.

3.2 Hypergeometric Ensemble of Higher-Order De Bruijn Graphs

We now introduce the details of *higher-order hypergeometric path anomaly detection* (HYPA), the main contribution of our work.

Mapping of null model to ensemble of k-th order De Bruijn graphs In the illustrative example, we showed that assessing whether a path of length k exhibits anomalous frequencies requires considering the distribution of frequencies under a null model *for each path separately.* The key idea of HYPA is to map the difficult problem of finding the frequency distributions of paths of length k under a null model to the simpler problem of finding the edge weight distribution in a null model for k-th order De Bruijn graphs. For this, we remember that the weights on the edges in a k-th order De Bruijn graph can exactly represent the frequencies of paths of length k observed in a dataset (cf. definition 3.2). We are thus interested in identifying which of these weights are anomalous compared to the baseline given by a $(k-1)$-order null model of paths. In each realization generated by such a $(k-1)$-order model, frequencies of paths of length $k-1$ are fixed, while the frequency of each path of length k follows a different distribution that depends on the null model. We can map each random realization to a different weighted k-th order De Bruin graph, obtaining a *statistical ensemble of k-th order De Bruijn graphs* whose probabilities are given by the null model. Since the frequencies of paths of length $k-1$ are fixed, the *total* out-degree $f^{\text{out}}_{\vec{v}} = \sum_{\vec{x}} f(\vec{v}, \vec{x})$ and the *total* in-degree $f^{\text{in}}_{\vec{v}} = \sum_{\vec{x}} f(\vec{x}, \vec{v})$ for each node $\vec{v}$ is the same across all realizations in this ensemble. However, De Bruijn graphs that correspond to different random realizations differ in terms of the exact edge weights $f(\vec{v}, \vec{w})$, which represent frequencies of paths of length k.

Distribution of edge weights in random k-th order graphs This mapping allows us to compute frequency distributions for individual paths of length k, conditional on the frequencies of paths of length $k-1$, based on a random model for k-th order De Bruijn graphs that preserves the total in- and out-degrees of all nodes while randomly shuffling the weights of edges. We can formalize the model as a stochastic process that randomly draws m edges, where m is the sum of all k-th order edge weights (the total number of paths of length k observed in the data). Different from simple random graph models, in this sampling process we must account for the fact that different edges in a k-th order De Bruijn graph have different probabilities to be drawn. Specifically, we are more likely to generate edges between pairs of nodes with a high in- and out-degree. In our null model of paths, this translates to the fact that a path of length k is more likely to occur if it continues a frequently occurring path of length $k-1$. We capture the fact that different edges in a k-th order De Bruijn graph occur with different probabilities using a matrix Ξ, where each entry corresponds to one possible pair of higher-order nodes that can be connected by an edge, and the value of the entry denotes how many times this pair of nodes can possibly be drawn. We thus obtain a sampling procedure that can be described by the multivariate hypergeometric distribution.

Since we consider k-th order De Bruijn graphs we must additionally account for the fact only pairs of higher-order nodes representing paths overlapping in $k-1$ first-order nodes can be connected (cf. Def. 2.1). When sampling from the multivariate hypergeometric distribution, we avoid drawing such pairs by setting their corresponding entry in Ξ to 0. This modification introduces the complication that weighted degrees are no longer guaranteed to be preserved, which violates the constraint that the frequency of paths of length $k-1$ is fixed. We overcome this with an optimization approach (Algorithm 2 in Appendix A.2.1[3]) to redistribute values of the Ξ matrix that were substituted by zeroes across the rest of the matrix, such that the weighted degrees $f^{\text{out}}_{\vec{v}}$ and $f^{\text{in}}_{\vec{v}}$ of the k order nodes $\vec{v}$ are preserved.

HYPA Algorithm The random De Bruijn graph model of order k introduced above is the basis for the HYPA algorithm to detect path anomalies (pseudocode in Appendix A.2, Algorithm 1). In particular, we argued that the distribution of edge weights in the statistical ensemble of random realizations are jointly de-

[3] Appendix A is available in the online version [27].

scribed by a multivariate hypergeometric distribution. We use the marginals of this distribution to calculate the distribution of edge weights for each edge as:

$$\Pr\left(X_{\vec{v}\vec{w}} = f(\vec{v},\vec{w})\right) = \binom{\sum_{ij}\Xi_{ij}}{m}^{-1}\binom{\Xi_{vw}}{f(\vec{v},\vec{w})}\binom{\sum_{ij}\Xi_{ij}-\Xi_{vw}}{m-f(\vec{v},\vec{w})}, \tag{3.1}$$

where $m = \sum_v f_v^{\text{out}} = \sum_v f_v^{\text{in}}$ is the sum of all weights in the graph and $X_{\vec{v}\vec{w}}$ is a random variable assuming the weight of edge $(\vec{v},\vec{w})$ in a random realization of a k-th order De Bruin graph. We use these marginal distributions to define the HYPA$^{(k)}$ score for a path $\overrightarrow{vw}$ of length k with observed frequency $f(\vec{v},\vec{w})$ as the cumulative distribution corresponding to Eq. (3.1):

$$\text{HYPA}^{(k)}(\vec{v},\vec{w}) := \Pr\left(X_{\vec{v}\vec{w}} \leq f(\vec{v},\vec{w})\right) \tag{3.2}$$

Since the HYPA$^{(k)}$ score is a probability, it assumes values in $[0,1]$. Paths whose HYPA$^{(k)}$ scores are close to 0 are likely to be under-represented compared to the random baseline. That is, the probability to obtain at random a frequency for this path that is lower or equal to the frequency in the data is small. On the other hand, a path whose HYPA$^{(k)}$ score is close to 1 is likely to be over-represented, meaning the frequency obtained at random for that path is likely to be smaller than the one observed in the data. A path that has a HYPA$^{(k)}$ score of 0.5 is equally likely to be observed with a higher or lower frequency at random, showing the least indication of an anomaly. Anomalous paths are determined by setting a discrimination threshold $\alpha \in (0,1]$ and classifying as under-represented any path $(\vec{v},\vec{w})$ with HYPA$^{(k)}(\vec{v},\vec{w}) < \alpha$ and as over-represented any path $(\vec{v},\vec{w})$ with HYPA$^{(k)}(\vec{v},\vec{w}) \geq 1-\alpha$.

Computational Complexity The asymptotic runtime of HYPA is $\mathcal{O}\left(N + \Delta^k(G)\right)$, where N is the size of $\mathbf{S}$, $\Delta^k(G)$ is the number of edges in a k-th order De Bruijn graph model G^k of paths in G. An upper bound on $\Delta^k(G)$ is proved in Appendix A.2.2. The implication of this bound is that for sparse real-world graphs, moderate values of k, and above a sufficiently large value of N, *our method scales linearly with the size of the data.*

4 Experiments

In this section we show that we can use the scores calculated by HYPA to detect paths with anomalous frequencies.[4] We note that the datasets we consider do not come with anomaly labels, and that we do not expect our notion of anomalous paths to correspond directly with any existing labeled data, since we have defined the anomalies we are detecting within a specific mathematical frame, rather than as deviation from a domain dependent "normal". With this in mind, we apply HYPA to synthetic data with known anomalies and empirical data representing trajectories through a transportation network. We show that the under- and over-represented paths detected without supervision fall into classes that can be validated using semantic and geographic information.

4.1 Baseline Method In the below experiments, we compare HYPA to a simple frequency-based anomaly detection (FBAD) of our own design. We note that despite similar problem settings, the methods for hypothesis testing on human trails presented in [39, 6] are not directly comparable with our work because the output is Bayesian evidence for a hypothesis on an entire dataset (a single number), whereas we are interested in edge-level analysis. Further, we did not compare with a method like [36] because, while based on detecting significant deviations from a Markov chain model, this method assumes that the data is given as one long sequence and detects anomalous subsequences, which does not correspond to any of the datasets we analyze here. Finally, a recently proposed method identifies significant sequential patterns via a permutation strategy with a Monte Carlo estimation procedure [43]. Despite similarity in purpose, the method has limited utility in our setting for two main reasons. First, it relies on the PrefixSpan algorithm [24] to mine the sequences for relatively frequent patterns. As can be noted from Fig. 1, low-frequency patterns can be path anomalies. Second, it does not incorporate constraints on the possible sequences, instead sampling from a uniform space of all possible permutations. For these reasons, we were unable to make a fair comparison and do not report results.

The baseline method FBAD computes the average μ and standard deviation σ of path counts and employs a user-defined threshold α to detect over- and under-represented paths. FBAD implicitly assumes that the distribution of edge weights is normal and thus paths should be considered anomalous if they are outliers with respect to this distribution. In particular, a path is labeled as over-represented if its frequency exceeds $\mu + \sigma\alpha$, and as under-represented if its frequency is smaller than $\mu - \sigma\alpha$. More details on FBAD are available in Appendix A.3 and Algorithm 3.

4.2 Synthetic Data We validate HYPA using a stochastic model that generates synthetic datasets of paths with varying lengths, in which a known set of paths with given length l exhibit anomalous frequencies. Adopting the well-known Erdős-Rényi model [19], our model generates paths in a random directed graph G with n nodes, where pairs of nodes are connected with probability p. Following definition 3.2, the random model generates paths based on an edge-weighted

[4] An implementation is available at `github.com/tlarock/hypa`.

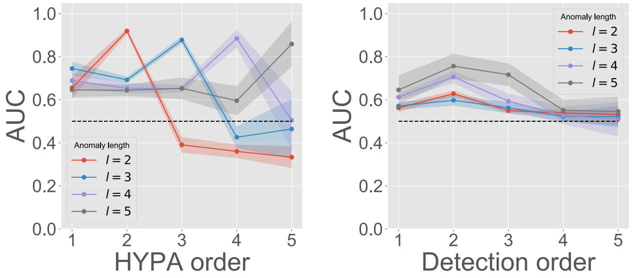

Figure 3: HYPA$^{(k)}$ detects injected path anomalies at the correct length with high accuracy. Each curve corresponds to one length l of generated anomalous paths, and represents the performance of classifying the anomalous paths using HYPA (left) or the naive FBAD method (right) applied at increasing orders k. HYPA detects the exact generated anomalies, i.e., performs highly at $k = l$. FBAD only performs relatively well in detecting short sub-paths (e.g., $k = 2$) of longer anomalies (e.g., $l = 5$). Averages and standard errors are over 10 independent experiments.

random walk in a k-th order De Bruijn graph of paths in the random graph G. By selectively changing transition probabilities in $\mathbf{T}^l$ (cf. definition 3.2), we introduce anomalous frequencies for a known set of paths at length l. Since all paths longer than l are generated by a (Markovian) random walk on a De Bruijn graph with order l, these paths will not exhibit anomalous frequencies beyond those expected from the anomalous frequencies of paths of length l. For details of the random path construction, see Algorithms 4 and 5 in Appendix A.5.1. In the following we report results for graphs with $n = 50$ nodes and an edge probability of $p = 0.05$ (our conclusions do not depend on these parameters).

We test whether HYPA detects anomalous path frequencies (i) with high accuracy, and (ii) at the correct length l introduced by our model. To this end, we calculate the performance of HYPA in a binary classification experiment, categorizing path frequencies as anomalous based on variable discrimination thresholds α for the HYPA$^{(k)}$ scores at different orders k. For each threshold α, we compute the true and false positive rates of detected anomalies w.r.t. the known ground truth and obtain a receiver operating characteristic (ROC) curve for which we can calculate the area under the curve (AUC). We repeat this experiment 10 times for each combination of anomalous path length $l \in [2, 5]$ and detection order $k \in [1, 5]$. Each curve in Fig. 3 presents the mean and the standard deviation of the AUC for anomalies detected at varying orders k, for a given anomaly length l. For $k \neq l$, we use as ground-truth the paths of length k that either include or are included in an anomalous path of length l generated by the synthetic model. For each l we observe that HYPA with the "correct" order $k = l$ is able to identify ground truth anomalies with high accuracy ($AUC \approx 0.9$, left plot), while the baseline FBAD method ($\sigma = 2$) is unable to detect path anomalies with high accuracy at any order, regardless of the order used for detection (max $AUC \approx 0.78$, right).

Efficiency and Balance in Flight Itineraries We analyze an empirical dataset of paths taken through a transportation system using HYPA. *Flights* comprise 5% of all travel itineraries of passengers flying in the US in the first quarter of 2018.[5] Characteristics of the dataset are presented in Table 3 (Appendix A.6).

Our first hypothesis is that return flights (ABA) are significantly over-represented, since passengers often leave from and return to the same airport. We first compute HYPA scores for $k = 2$, then separate return from non-return flights and compute the fraction of over-represented paths in each category for varying discrimination thresholds α. The results in Table 1 support the hypothesis that return flights are strongly over-represented compared to the null model.

Table 1: Fractions of over-represented paths of length 2 between airports for return flights (5840 unique paths) and non-return flights (409254 unique paths) at different discrimination thresholds α.

α	0.05	0.01	0.001	0.0001	0.00001
Return	0.915	0.851	0.760	0.688	0.628
Non-return	0.340	0.130	0.023	0.004	0.001

However, we still observe a number of over-represented non-return flight paths. We hypothesize that many of these paths connect small airports to large airports via regional hubs. This means that a relatively short distance trip (e.g. from ORL to ATL) is required before a flight from the regional hub to a relatively distant destination (e.g. ATL to LAX). Rather than classifying airports by their size and role in the network, we test this hypothesis by defining *distance balance*, a measure that captures to what extent one leg of a trip dominates the total trip distance. In a perfectly balanced trip (ABC), the distance of the two legs is equal, e.g. $d(A, B) = d(B, C)$. The most common example of a perfectly balanced trip is the return trip, where $A = C$. In an imbalanced trip, one of the legs of the trip is much larger. We define balance by the ratio $\frac{d(A,B)-d(B,C)}{d(A,B)+d(B,C)}$. It approaches -1 or 1 when the distance of one leg of the trip is much greater than on the other. We expect flights with extreme values to be over-represented as they represent long distance flights that start from small, local airports, fly a short distance to a regional hub, then on to a much further off destination (as well as the reverse). The distribution of balance for over- and under-represented paths of length two ($\alpha = 0.05$) is shown in Fig. 4 (left). We find very few under-represented flights near extreme values of balance, while a larger fraction

[5] Data from US Bureau of Transportation Statistics TransStats `http://www.transtats.bts.gov/Tables.asp?DB_ID=125`.

of over-represented paths are found near -1 and 1. This supports our hypothesis that unbalanced flights tend to be more over-represented than balanced flights.

We now formulate hypotheses based on a notion of *efficiency* for airline trips. We measure efficiency as the ratio of the distance between source and destination, $d(A, C)$, with the actual flight distance, $d(A, B) + d(B, C)$. Using this measure, a straight line between airports A, B and C has maximum efficiency of 1, while a low efficiency trip implies that the actual flight distance is much larger than the straight line distance between the origin and destination. We hypothesize that highly efficient paths are over-represented, while inefficient paths are under-represented in the data. The middle plot of Fig. 4 shows a large peak in the fraction of under-represented paths at very low efficiency, then a steady decrease in under-represented paths as efficiency increases. In the right hand plot we see that after return flights are accounted for (peak at efficiency 0), the fraction of over-represented paths increases monotonically with efficiency. These results indicate that more efficient paths are indeed more likely to be over-represented, and that the more efficient a path is, the less likely it is to be under-represented.

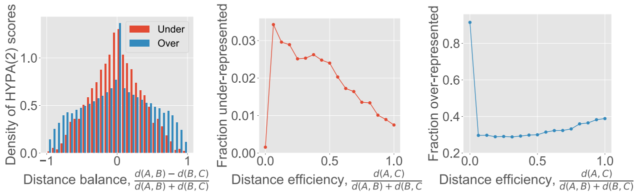

Figure 4: Left: extreme values of *balance* correspond to over-represented paths, confirming that short flights followed by long flights are typical (e.g. flights to a regional hub, then a national hub). Middle/Right: The fraction of over- and under-represented paths varies with the efficiency of the itinerary. After return flights are accounted for, the fraction of under-represented paths decreases with efficiency (middle), and vice versa for over-represented paths (right).

5 Conclusion

We presented HYPA, a novel approach for unsupervised detection of path anomalies in sequential data on networks. By providing a new theoretical basis for anomaly detection in graphs, our work advances the state-of-the-art in multiple directions. We introduced the problem of path anomaly detection and showed that frequency-based anomaly detection techniques cannot address it. Projecting paths through a first-order network onto higher-order De Bruijn graphs, we showed that path anomaly detection can be reduced to the detection of anomalous edge weights in a higher-order graph. Building on an analytically tractable null model of higher-order De Bruijn graphs, we developed a scalable method, HYPA that is able to detect paths that exhibit significant deviations from a random baseline, allowing us to assess statistical deviations in frequencies of paths traversing the nodes of a graph. Some limitations of HYPA could be addressed in future work, including (i) automatically selecting an appropriate discrimination threshold, (ii) combining analysis at different orders, and (iii)incorporating domain specific notions of anomalous paths.

Acknowledgements IS acknowledges support by Swiss National Science Foundation grant 176938. LaRock and Eliassi-Rad were supported by (1) the Combat Capabilities Development Command Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA) and (2) the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the Under Secretary of Defense for Research and Engineering or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation here on.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *IEEE ICDE*, pages 3–14, 1995.

[2] L. Akoglu and C. Faloutsos. Event detection in time series of mobile communication graphs. In *Army Science Conference*, pages 77–79, 2010.

[3] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015.

[4] M. Atzmueller. Detecting community patterns capturing exceptional link trails. In *ASONAM*, pages 757–764, 2016.

[5] M. Atzmueller, A. Schmidt, and D. Arnu. Sequential modeling and structural anomaly analytics in industrial production environments. In *LWDA*, pages 283–290, 2016.

[6] M. Becker, F. Lemmerich, P. Singer, M. Strohmaier, and A. Hotho. Mixedtrails: Bayesian hypothesis comparison on heterogeneous sequential data. *Data Mining and Knowledge Discovery*, 31(5):1359–1390, 2017.

[7] R. Bertens, J. Vreeken, and A. Siebes. Keeping it short and simple: Summarising complex event sequences with multivariate patterns. In *KDD*, page 735–744, 2016.

[8] B. Boden, S. Günnemann, and T. Seidl. Tracing clusters in evolving graphs with node attributes. In *CIKM*, pages 2331–2334, 2012.

[9] P. Bogdanov, M. Mongiovì, and A. K. Singh. Mining Heavy Subgraphs in Time-Evolving Networks. In *IEEE ICDM*, pages 81–90, 2011.

[10] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *KDD*, pages 280–284, 2000.

[11] Casiraghi and Nanumyan. Generalised hypergeometric ensembles of random graphs: The configuration model as an urn problem. *arXiv:1810.06495 [physics]*, 2018.

[12] G. Casiraghi, V. Nanumyan, I. Scholtes, and F. Schweitzer. From relational data to graphs: Inferring significant links using generalized hypergeometric ensembles. In *Social Informatics*, pages 111–120, 2017.

[13] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In *VLDB*, volume 98, pages 606–617, 1998.

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

[15] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE TKDE*, 24(5):823–839, 2012.

[16] V. Chandola, V. Mithal, and V. Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *IEEE ICDM*, pages 743–748, 2008.

[17] N. G. de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie v. Wetenschappen*, 49:758–764, 1946.

[18] M. El-Sayed, C. Ruiz, and E. A. Rundensteiner. Fs-miner: efficient and incremental mining of frequent sequence patterns in web logs. In *WIDM*, pages 128–135, 2004.

[19] P. Erdos and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[20] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[21] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *TKDE*, 26(9):2250–2267, 2014.

[22] R. Gwadera, M. Atallah, and W. Szpankowski. Markov models for identification of significant episodes. In *SDM*, pages 404–414, 2005.

[23] R. Gwadera and F. Crestani. Ranking Sequential Patterns with Respect to Significance. In *Advances in Knowledge Discovery and Data Mining*, pages 286–299. 2010.

[24] Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. PrefixSpan,: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE*, pages 215–224, 2001.

[25] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14(1):99–129, 2007.

[26] T. Lane and C. E. Brodley. An empirical study of two approaches to sequence learning for anomaly detection. *MLJ*, 51(1):73–107, 2003.

[27] T. LaRock, V. Nanumyan, I. Scholtes, G. Casiraghi, T. Eliassi-Rad, and F. Schweitzer. Hypa: Efficient detection of path anomalies in time series data on networks. *arXiv preprint arXiv:1905.10580 [physics]*, 2020.

[28] R. Laxhammar and G. Falkman. Online learning and sequential anomaly detection in trajectories. *TPAMI*, 36(6):1158–1173, 2014.

[29] F. Lemmerich, M. Becker, P. Singer, D. Helic, A. Hotho, and M. Strohmaier. Mining subgroups with exceptional transition behavior. In *KDD*, pages 965–974, 2016.

[30] Z. Liu, V. W. Zheng, Z. Zhao, H. Yang, K. C.-C. Chang, M. Wu, and J. Ying. Subgraph-augmented path embedding for semantic user search on heterogeneous social network. In *WWW*, pages 1613–1622, 2018.

[31] I. Melnyk, B. Matthews, H. Valizadegan, A. Banerjee, and N. Oza. Vector autoregressive model-based anomaly detection in aviation systems. *J. of Aerospace Information Systems*, 13:161–173, 2016.

[32] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 1995.

[33] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[34] T. P. Peixoto and M. Rosvall. Modelling sequences and temporal networks with dynamic community structures. *Nature communications*, 8(1):582, 2017.

[35] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5, 2014.

[36] R. Sadoddin, J. Sander, and D. Rafiei. Finding Surprisingly Frequent Patterns of Variable Lengths in Sequence Data. In *SDM*, pages 27–35, 2016.

[37] I. Scholtes. When is a network a network?: Multi-order graphical model selection in pathways and temporal networks. In *KDD*, pages 1037–1046, 2017.

[38] S. Servan-Schreiber, M. Riondato, and E. Zgraggen. Prosecco: Progressive sequence mining with convergence guarantees. In *IEEE ICDM*, pages 417–426, 2018.

[39] P. Singer, D. Helic, A. Hotho, and M. Strohmaier. Hyptrails: A bayesian approach for comparing hypotheses about human trails on the web. In *WWW*, pages 1003–1013, 2015.

[40] P. Smyth. Clustering sequences with hidden markov models. In *NIPS*, pages 648–654, 1997.

[41] A. Tajer, V. V. Veeravalli, and H. V. Poor. Outlying sequence detection in large data sets: A data-driven approach. *IEEE Signal Processing Magazine*, 31(5):44–56, 2014.

[42] E. Tonnelier, N. Baskiotis, V. Guigue, and P. Gallinari. Anomaly detection in smart card logs and distant evaluation with twitter: a robust framework. *Neurocomputing*, 298:109–121, 2018.

[43] A. Tonon and F. Vandin. Permutation Strategies for Mining Significant Sequential Patterns. In *IEEE ICDM*, 2019.

[44] S. Walk, P. Singer, and M. Strohmaier. Sequential action patterns in collaborative ontology-engineering projects: A case-study in the biomedical domain. In *CIKM*, pages 1349–1358, 2014.

[45] D. Zhou, J. He, Y. Cao, and J.-S. Seo. Bi-level rare temporal pattern detection. In *IEEE ICDM*, 2016.