Effect of Data Repair on Mining Network Streams

***Abstract*—Data quality issues have special implications in network data. Data glitches are propagated rapidly along pathways dictated by the hierarchy and topology of the network. In this paper, we use temporal data from a vast data network to study data glitches and their effect on network monitoring tasks such as anomaly detection. We demonstrate the consequences of cleaning the data, and develop targeted and customized cleaning strategies by exploiting the network hierarchy.**

***Keywords*-glitches, missing data, network data, outliers;**

I. Introduction

Corporations, government agencies, social media and search engines record every type of data that can be monitored, generating vast amounts of data each day. With Big Data comes the promise of increased amount of information, leading to better and more timely decisions. However, no matter how large the data, the dependability of decisions depends crucially on the quality and reliability of the data.

In order to realize the full potential of the enormous amounts of data, and its limitations in informing decisions, it is critical to understand the impact of *data glitches* on statistical analyses and stream mining. In this paper, we focus on a classic example of Big Data, network data, and highlight the effect of data problems and their repair on network mining tasks. Quality issues in network data pose special challenges because glitches are propagated along specific temporal and topological pathways, and may have a disproportionate impact on the analyses.

A. *Our Approach*

There is a vast amount of literature on different types of glitches and their treatment. An exhaustive coverage can be found in [3] and [5]. We focus on a single type of glitch, namely missing values, and study their effect on network analysis as exemplified by outlier detection.

At first glance, this might seem counter-intuitive. If a value is missing, how do we know whether it is an outlier? In order to address this concern, we start with a time series of network data that contains a minimal number of data glitches and establish the *ground truth* by running an outlier detection algorithm on it. We then introduce different amounts of missing data into the time series, and examine the impact of the added missing data on the performance of outlier detection. We quantify the impact by comparing the occurrence of outliers before and after missing values are introduced.

Next, we repair the data by imputing missing values using different imputation methods, and run the outlier detection on the repaired data. We compare the performance of outlier detection before and after data repair, for varying proportion and patterns of data glitches. Since the ground truth has been established *a priori* by running the outlier detection before the introduction of glitches, we can benchmark the effect of the newly introduced glitches, as well as their repair, against the ground truth, by using popular classification error metrics on the outliers.

Hence our approach for studying the impact of data repair on network stream mining tasks can be summarized in the following four broad components:

- A network mining task - univariate outlier detection;
- Data glitches and glitch patterns that may affect the mining task - missing values, occurring at random and in blocks;
- Data repair techniques - five imputation methods;
- Performance metrics for analysis, measured for: the ground truth, after the introduction of glitches, and after data repair - *precision*, *recall*, and F*-measure*, and *recoverability*.

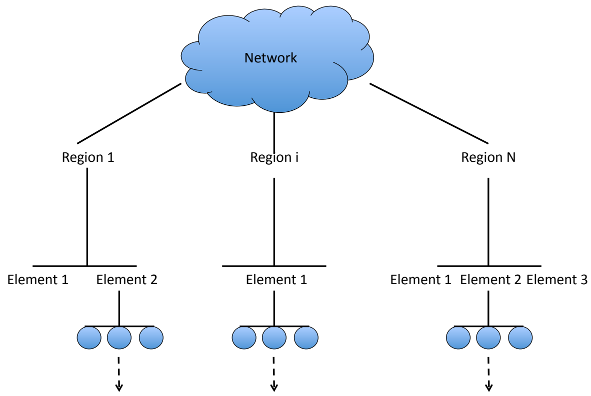
Note that while we use missing values and a simple univariate anomaly detection method, our approach extends trivially to multiple types and patterns of glitches, and to any stream mining or analysis task.

In addition to studying the effect of data repair on network analysis, we also address the use of developing customized data cleaning solutions based on patterns of glitch dependence. In [2], Berti-Equille *et al* suggest the use of patterns of glitches to identify and design customized and effective cleaning strategies. Therefore, we identify existing canonical glitch patterns, and use them to assign *orphaned records* (those that have no unique identifier) to parent regions in the network hierarchy.

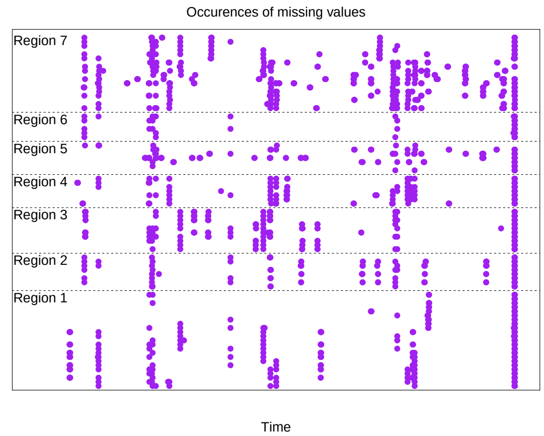
B. *Organization*

The rest of the paper is organized as follows. In Section II we describe the network data stream, the glitches that are embedded in it, and their repair. In Section III, we describe the analysis of interest, a simple univariate outlier detection method. Section IV details our experiments with data repair through imputation of missing values, and how it effects outlier detection in network data. In Section V, we measure the effectiveness of this data mining task by using well known performance metrics.

We address the problem of assigning orphaned data (records that have no unique identifiers) to appropriate



(a)



(b)

Figure 1. **(a) Schematic network hierarchy, and (b) patterns of glitches (missing values in this case) observed in network elements j corresponding to different Regions i organized as depicted in (a).**

regions in the network hierarchy in Section VI. We accomplish this by using glitch pattern similarity as measured by the Earth Mover's Distance, and place the orphaned data within the most "similar" region in the network hierarchy. This enables us to pick an appropriate data repair strategy, dictated by the neighbors in the network. Finally, we present our conclusions in Section VII.

II. Data and Its Quality

In this paper, we use network data that were collected from a mobility network over a period of several weeks. The data consist of measurements on network performance indicators gathered from individual network elements. Network engineers use this data to monitor, in real-time, the performance of the network. In particular, outlying values of performance metrics are used to identify possible performance degradations in the network. However, as with any real-world process, network data could contain data problems. Common types of data glitches include missing values, inconsistent values that violate pre-existing constraints ("network usage cannot be negative"), duplicates caused by multiple polling, and incorrect values. In this paper, we focus on missing data, and study the impact of missing data and their repair on network outlier detection.

A. Data Description

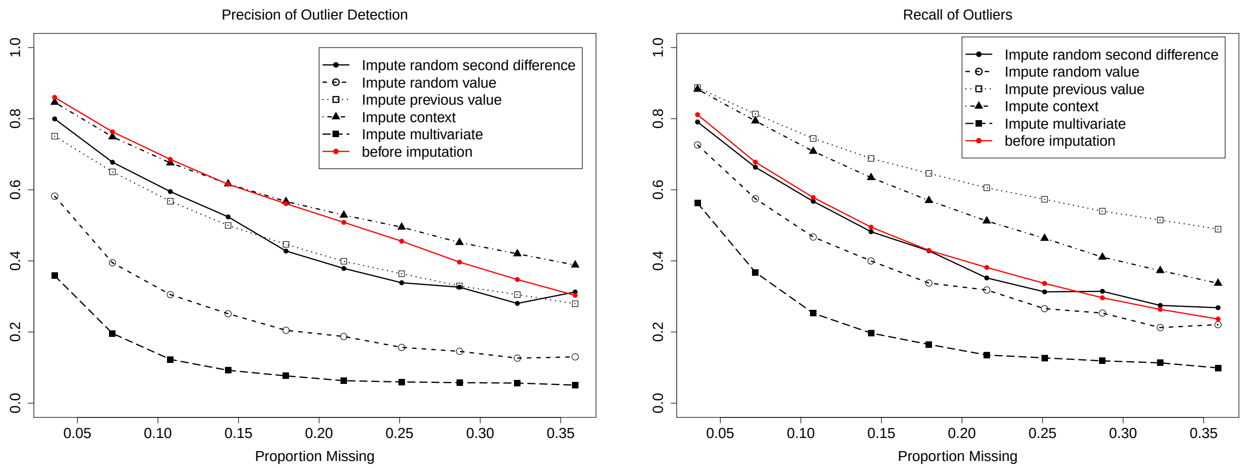
Data are collected from a type of network element, S, at regular short intervals of time. Figure 1(a) depicts a simple network schematic. Individual calls, text messages and data traffic are picked up by antennae on cell towers that transmit them up the network hierarchy via network controllers all the way to the network elements j that are responsible for servicing a geographical area, by performing routing and authentication. We will exploit the geographical characteristics in identifying and customizing data cleaning solutions. A detailed explanation of mobility networking is beyond the scope of this paper. Please see [14] for an introduction to the fundamentals of wireless communication networks.

B. Data Glitches

Data glitches are problems with the data that could lead to errors in analysis and interpretation. The simplest and perhaps most common type of data glitch is a *missing* value. Individual attributes, or entire records could be missing. Missing values could arise as a result of faulty measuring instruments, connection problems, or even human error.

A data record usually consists of two parts: the first part identifies an entity (a telephone, a person) using a *unique identifier* such as a phone number or name. In the case of network data, every data record has a unique identifier that is associated with a network element and that network element alone. For simplicity, we will assume that these identifiers do not change with time. The rest of the data record consists of attributes of the entity associated with the identifier. In the case of a telephone, it could be the number of calls; in the case of a person, physical attributes such as height and weight; and in the case of our network data, performance indicators. In one scenario of missing data, any and all of the attributes could be missing, while the unique identifier is fully populated. In another scenario, the unique identifier itself could be missing, leaving the data record "orphaned".

In this paper, we address data repair in both these scenarios. For the first scenario, data repair consists of applying numerical imputation techniques. We explore the effectiveness of several such techniques at repairing missing values. In the second scenario, we make use of the similarity of glitch patterns between orphaned records and normal records to impute regional identifiers to the orphaned records. By



(a) (b)

Figure 2. **Performance of 5 data repair techniques (imputation) in assisting analyses (outlier detection): (a) Precision, and (b) Recall, for a single network element S averaged over hundred replications each at varying proportion of missing values (X-axis). Context-based methods imputing previous value and average of the last three values perform well.**

assigning orphaned records to known regions with well-tested cleaning strategies, we can choose the most effective cleaning strategies for them.

C. Patterns of Glitches

Data glitches are often caused by underlying real world phenomena and exhibit patterns. They could appear in clusters, or have complex temporal and spatial correlations. These patterns can be exploited to develop efficient cleaning strategies as explained in [2]. In Figure 1(b), we plot the time series of occurrence of missing values for a single attribute, for several network elements, grouped by the regional hierarchy. It is clear that there are both global and local patterns. In Section VI, we exploit these local patterns to assign orphaned records to regions in order to employ appropriate cleaning strategies.

D. Glitch Detection

Glitch detection is trivial for missing values. We look for the absence of individual attributes in a data record. In addition, since we expect to receive a data record for every known element S at least once during a pre-specified short time interval, the absence of a record indicates that the entire record is missing.

Note that glitch detection is far more complex for other types of glitches, for example when data are incorrect, mangled or censored. If a data value is misrecorded as 91 instead of 19, we would need to compare with an existing baseline, signature or profile to detect the glitch. However, if it falls within the realm of normal behavior, then there is no viable way of detecting it.

E. Imputation Methods

The aim of repairing data is to provide better results and inference by making the data more usable. Missing value data repair methods, known as imputation methods, range from the simple to the sophisticated. Please see [9] for further details. Imputation methods consist of replacing missing values with plausible data values.

For this study we consider five imputation methods:
1. Imputation by randomly drawing a value from the last 30 measurements. This assumes that the data values are independent, and without trend or seasonality.
2. To address seasonality and trend, impute by randomly drawing from the last 30 second differences, defined by $D_i = Y_{i+1} - 2Y_i + Y_{i-1}$. This assumes independence of the second differences, and that second differencing removes trend and seasonality.
3. Impute using the previous non-missing value, where the imputed value $\hat{Y}_i = Y_{i-1}$.
4. A variation of point 3 above, impute the average of the last three (non-missing) values, $\hat{Y}_i = (Y_{i-1}+Y_{i-2}+Y_{-3})/3$.
5. Impute values using the MI procedure in SAS. See the SAS manual for details. It uses the joint distribution of other attributes to impute missing values.

III. Outlier detection methods

There is an enormous amount of literature on outlier detection as surveyed in [7]. In this paper, since our focus is not outlier detection *per se*, we will use a very simple univariate outlier detection method.

We consider second-order differences of the attribute, and use a moving window of size 30 to compute the standard deviation of the second differences,

$$s_i = \sum_{j=1}^{30} (D_{i-j} - m_i)^2/29,$$

where $m_i = \sum_{j=1}^{30} D_i/30$. Any observation whose second difference D_i lies outside 2.5 standard deviations from the

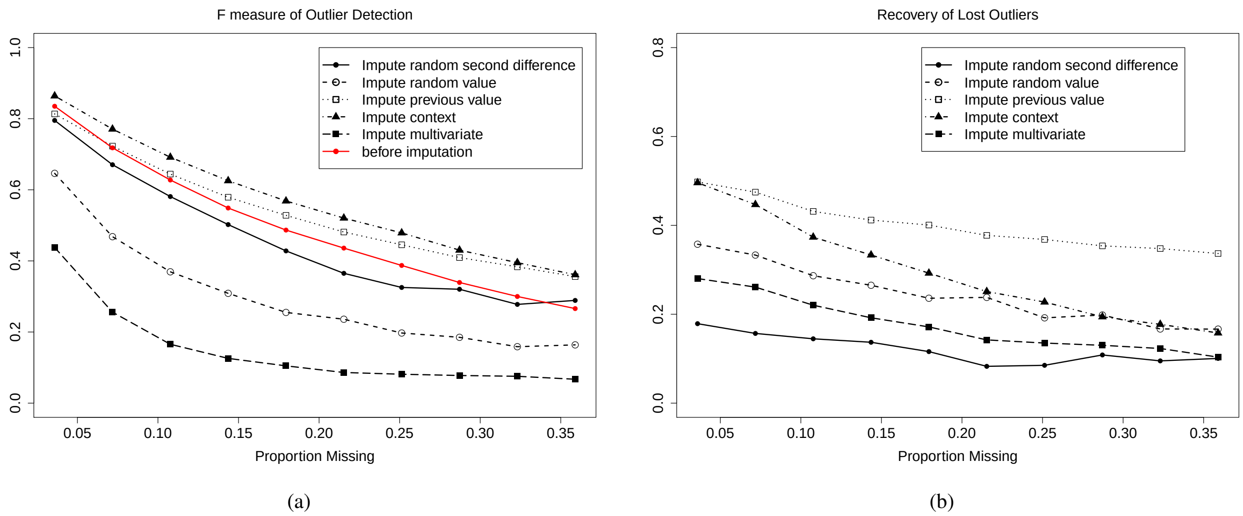


Figure 3. **Performance of 5 data repair techniques (imputation) in assisting analyses (outlier detection): (a) F-Measure, and (b) Recovery, for a single network element S averaged over hundred replications each at varying proportion of missing values (X-axis). Context based methods perform well. However, in terms of recovering lost outliers, most viable methods fall off, as expected with increasing proportion of missing values.**

mean is considered an outlier. This is reasonable since we expect second differences to be independent and drawn from a Gaussian distribution.

IV. Performance

We performed a comprehensive simulation study to examine the effect of data repair – missing value imputation in this case – on data mining tasks as exemplified by network outlier detection.

Formally, let the original (real) time series data be X_{O}. Outlier detection applied to X_{O} yields a set of outliers, denoted by T_{O}, which represents the time points where outliers are detected. Randomly chosen data points are removed, resulting in X_{M}, the simulated time series with artificially induced missing values. Outlier detection, when applied to X_{M}, produces T_{M}. We expect that T_{O} will contain more outliers than T_{M}.

Missing values are then imputed and outlier detection applied to the repaired data, X^i_{M}. This process yields the set of outliers T^i_{M}. We are interested in the comparison of the three sets T_{O}, T_{M}, and T^i_{M} – in particular, the agreement between T^i_{M} (outliers after the artificially introduced missing values have been repaired through imputation), and T_{O} (outlier in the original series), in relation to T_{M} (outlier in the series with artificially introduced missing values). In order to measure the performance of repair techniques in assisting analyses, we use the following metrics.

A. Metrics

As discussed above, T_{O} represents outliers that are detected in the original series, before any missing values were introduced, and represents the ground truth against which we compare the performance of imputation techniques. When we introduce missing values, we expect outlier detection performance to degrade, so that T_{M} will likely contain fewer outliers than T_{O}.

The aim of imputation is to repair the data so that we can recover the ground truth, with the expectation that the sequence of outliers in the imputed series T^i_{M} is closer to T_{O} than T_{M}. In order to measure the performance of the outlier detection method, we use the well-known measures of recall, precision and F-measure, and a measure we call *recoverability*. They are defined as follows:

Recall R measures the effectiveness of the imputation method to enable detection of the original outliers:

$$R = |T^i_{\mathrm{M}} \cap T_{\mathrm{O}}|/|T_{\mathrm{O}}|,$$

where $|A|$ represents the size of set A.

Precision P measures false positives and quantifies incorrectly identified outliers:

$$P = |T^i_{\mathrm{M}} \cap T_{\mathrm{O}}|/|T^i_{\mathrm{M}}|.$$

Precision and recall usually move in opposite directions, and one usually improves at the expense of the other.

The *F-measure*, denoted by F, captures this trade-off and is given by:

$$F = \frac{1}{\alpha(1/P) + (1-\alpha)(1/R)},$$

where α is the user's weighting of precision and recall. See [10] for details. We use $\alpha = 0.5$ in this paper.

Some of the outliers are "lost" as a part of the introduction of missing values. We can measure how well these lost outliers are detected after imputation, by restricting recall to the set of lost outliers instead of all originally detected outliers. We call this

Recoverability R_b, defined as

$$R_b = |T^i_{\mathrm{M}} \cap (T_{\mathrm{O}}) \setminus T_{\mathrm{M}}|/|T_{\mathrm{O}} \setminus T_{\mathrm{M}}|.$$

V. Experiments and Findings

For each of the five imputation methods listed in Section II-E, we conducted the following experiment. We took the time series corresponding to a single attribute, and randomly removed p proportion of the attribute values to simulate missing values. For each p, we repeated this to create 100 such time series (replications) with randomly missing values. For each of the 100 series, we ran the outlier detection as described in Section III, and after imputation, computed the performance metrics described in Section IV. Finally, we averaged the metrics over the 100 replications. We then increased the proportion p of missing values and re-ran the experiments.

Figures 2 and 3 show the results of our experiments, where the X-axis represents the proportion of missing values and the Y-axis represents the performance measure. In Figure 2(b), the red line shows the recall obtained directly from the data with missing values, without doing any imputation. We find that the two methods 1 and 2, based on randomly drawing from the current moving average window, whether it is actual values or the second differences, do not achieve any improvement. In contrast, by using the previous measured value (method 3), the recall is higher. As the proportion of missing values increases, the recall drops as expected, but the advantage of imputation over "no imputation" increases for methods 3 and 4 that use previous values. The method based on the previous 3 values (method 4, labeled "context") achieves the highest precision of outlier detection and the highest F-measure.

This is understandable, since in a time series setting, context is very important and it is reasonable to use the most recent value(s) as the best guess for a missing value, because of strong autocorrelation. Furthermore, with increasing proportion of missing values, these methods lose their effectiveness as more and more of the context is removed. On the other hand, imputing based on information from the joint distribution with other attributes (method 5) provides no advantage, perhaps because the outliers are based on a single attribute.

A. Patterns of Missing Values

Next, we introduced the missing values in blocks, rather than at random, so that the missing values occur sequentially in runs of length 20, at random points in the time series. Therefore, much of the data context is wiped out whenever there is a missing value.

In Figure 4, we plot a single instance of the time series of the original attribute (black), and overlay the corresponding modified time series with artificially introduced missing values (grey). The underlying time series in black is visible in places where the attribute values have been removed and are missing. The dark blue curve corresponds to the second order differences.

Attribute values that were missing in the original time series are shown as purple dots, and constitute a very small fraction of the data. Missing values that were artificially insinuated into the time series are shown as purple tick marks. Notice that they occur randomly in Figure 4(a), while they occur in blocks of 20 in Figure 4(b). This is highlighted by the fact that the original curve (black) is much more visible in (b) where there are runs of missing values resulting in large gaps in the time series data.

Outliers detected in the original time series before imputation, corresponding to T_{O}, are shown as pink circles. Once we perform the data repair and impute the small fraction of missing values in the original time series and re-run the outlier detection, we obtain a different set of outliers shown as red dots.

Outliers detected in the time series injected with missing values but before imputation are represented as lime circles and corresponding to T_{M}. Once we impute the missing values, we re-run the outlier detection and and obtain outliers after imputation shown as green dots that correspond to T^i_{M}.

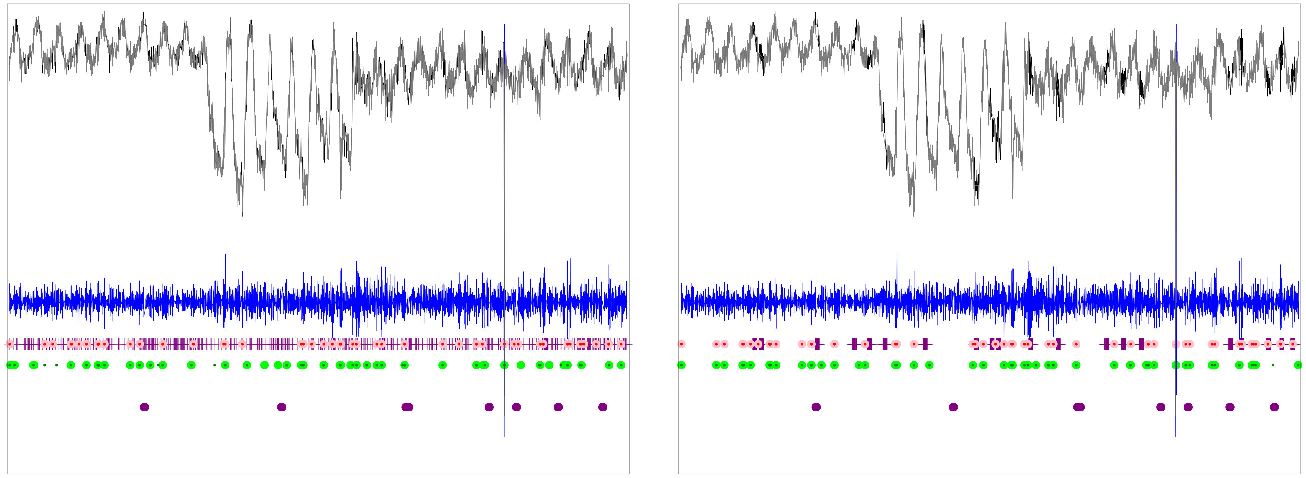
Ideally, we want the green dots, outliers after imputation of the artificially introduced missing values, T^i_{M}, to line up with the outliers in the original series T_{O}, the pink circles.

The performance measures for outlier detection after repairing blocks of missing values are shown in Figure 5. Notice that in Figure 5(a), imputing the previous value (method 3) is not that effective anymore. In fact, most methods do not improve over doing no imputation. The contextual methods do not work well now because much of the context has been removed due to the blocks of missing values. However, when it comes to recovering lost outliers (recoverability), Figure 5(b) shows that imputing a random value (method 1) is the most effective. This is because, in the absence of relevant information, using a random value offers the best approximation to recovering outliers.

VI. Assigning Orphaned Data to the Network

Different proportions and patterns of missing values impact the information contained in the data in different ways. They exert an influence on the effectiveness of methods used to restore or repair the data as well. While it is not surprising that the optimal methods for repairing data depend on the pattern of glitches, given the immensity of network data, it is inefficient to identify optimal methods for each individual data series separately. Since the number of methods for data repair is usually small, either due to cost or by choice, a far more efficient way is to group data series together based on their glitch patterns, and to identify optimal repair strategies for each group. In particular, we can assign orphaned data to regional groups based on glitch pattern similarity. Once the assignment is complete, we can choose the best cleaning strategy based on the group that they are assigned to.

Our method for classification is simple. A glitch vector is constructed for each data series, and a similarity measure for two such glitch vectors is chosen. Given several canonical



(a) (b)

Figure 4. **The success of an imputation mechanism in assisting outlier detection depends on the patterns of glitches: Shown here, a single element S, and a single attribute measured over time: Black=original time series, Grey=simulated time series with artificially introduced missing values, Dark Blue=second order differences, Pink circles=outliers in original $T_{\mathbf{O}}$, Red dots=outliers in original series after imputing, Lime circles=outliers in simulated series $T_{\mathbf{M}}$, Green dots=outliers in simulated after imputation $T^i_{\mathbf{M}}$, Purple ticks=artificially introduced missing values, Purple dots = original missing values. (a) Random missing values (purple ticks) (b) Missing values in blocks of 20.**

patterns of glitch vectors, an orphaned glitch vector is assigned to the group with the closest (or most similar) glitch vector.

Formally, let

$$n_i G_i = (0, \ldots, 1, \ldots 0, \ldots 1, \ldots 0)$$

be a series of 0's and 1's where 1's correspond to time points of missing values, and n_i the number of missing values. Hence G_i is a vector of equal weights at the locations of missing values, with the weights summing to one.

We choose the Earth Mover's Distance (EMD) as the similarity measure between two glitch vectors: $D(i, j) = \text{EMD}(G_i, G_j)$. It has an intuitive interpretation and has been found to be useful in a number of applications involving network data [1], [6], [13]. However, other similarity measures can be used as well.

The EMD, also known as Wasserstein's metric, is a well-known algorithm in the computer vision and image retrieval communities, see [11], [12]. The Mallows distance is a special case, see [8]. Given two distributions P and Q, the EMD algorithm computes the amount of work done needed to convert one distribution to the other. Specifically, suppose P and Q are distributions on $i = 1, \ldots, n$. Let $F = \{f_{ij}\}$ represent the exchange of density between i to j to convert P to Q. Define

$$W(F; P, Q) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} |i - j|,$$

and let $F^* = \{f^*_{ij}\} = \text{argmin}_F W(F; P, Q)$. Then the Earth Mover's Distance for P and Q is

$$\text{EMD}(P, Q) = W(F^*; P, Q) / \sum_i \sum_j f^*_{ij}.$$

Note that the above method can be easily extended to multiple glitch types. Suppose G^1_i and G^2_i are the glitch vectors for X_i for two different glitch types, missing values and outliers, say. Then a combined EMD defined by

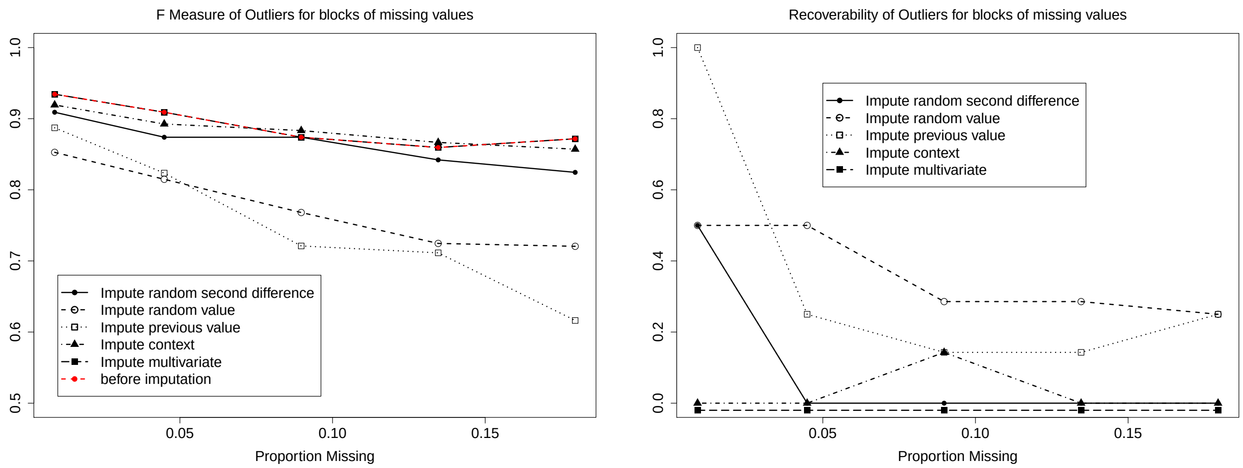
$$D_{12}(i, j) = \beta \text{EMD}(G^1_i, G^1_j) + (1 - \beta) \text{EMD}(G^2_i, G^2_j)$$

can be used as the similarity measure, where β represents the weighting of the two glitch types. The actual choice of β depends on the specific application.

We conducted a study using the network data to illustrate the method. The data are described in Section II. Since the network elements are organized hierarchically, we can group the data into groups that correspond to different sections of the network that the elements belong to. Data glitches tend to propagate through the network, hence we expect the glitch vectors of network elements that are close together in the network hierarchy to be similar. In our case, closeness in the network implies lying within the same "region". This similarity of glitch vectors is evident from Figure 1(b) which shows the presence of missing values (purple dots) for each network element, grouped by their "location" in the network, labeled Regions 1 to 7.

The actual experiment consists of repeatedly drawing 5 glitch vectors at random from a set of a 100 network element vectors to form an *orphaned*, unlabeled set. For each of these 5 glitch vectors, we compute the EMD to the remaining 95 labeled glitch vectors. The unlabeled vector is assigned to the group that has the smallest EMD from it.

In other words, if L represents the set of labeled glitch vectors, with elements $L_1 \ldots L_{95}$, and $g_i, i = 1, \ldots 95$ are the group labels, then the label assigned to the unlabeled



(a) (b)

Figure 5. Performance when attribute values are missing in chunks. (a) F-measure; note that imputing previous value is no longer as effective as when attribute values are missing at random. Imputation is almost not worth the effort. (b) Ability to recover outliers "lost" or removed during the simulation, after the missing value imputation process.

vector U_1 is g_j, where $j = \text{argmin}_i \text{EMD}(U_1, L_i)$. The assigned label g_j is then compared to its true label.

Using missing value glitch vectors, we obtained an accuracy rate of 0.68. This compares favorably to a rate of 0.14 if the regions were to be assigned randomly. The accuracy rate did not change when we used the combined EMD measure using both missing values and outliers in the glitch vectors. This lack of improvement with the additional use of outlier glitch vectors is due to the fact that the occurrences of outliers tend to be rather evenly spread out, so that the glitch vectors are similar across groups as well as within groups.

VII. Conclusion

In this paper, we studied and quantified the impact of data glitches and their repair on network analyses. We used missing values as exemplars of data glitches, and studied their effect on outlier and anomaly detection in network data. The analysis is affected by the proportion of missing as well as the pattern of missing values. For example, missing at random can influence analysis differently than missing in chunks.

We studied five different data repair techniques, missing value imputation in our case. Here too, the pattern of glitches influences the efficacy of imputation techniques. But beyond a certain point, no method of imputation can alleviate the problem when a large proportion of data is missing.

Finally, we used the pattern of glitches to assign "orphaned" records whose unique identifier is missing. Assignment is made by choosing the region with the most similar glitch pattern based on the Earth Mover's Distance (EMD). The assignment can be refined and made more specific in terms of the position in the network hierarchy by using similarity based on other attribute patterns.

A. Future Work

In the future, we will focus on simultaneous occurrence of multiple types of glitches, and on multivariate outliers based on multiple attributes. With multivariate outlier detection, we expect multivariate imputation methods to perform better. It will be interesting to study the performance of univariate imputation methods and how their performance varies with the strength of correlation between attributes.

In addition, we will quantify the effect of data repair in terms of the notion of *statistical distortion*, see [4]. Statistical distortion measures the bias introduced by data repair, and measures the change in the statistical distribution of the attributes caused by the repair. It is more general than studying the effect on a specific analyses such as outlier or anomaly detection. Statistical distortion has been proposed as a data quality metric.

References

[1] D. Applegate, T. Dasu, S. Krishnan, and S. Urbanek. Unsupervised clustering of multidimensional distributions using earth mover distance. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 636–644, New York, NY, USA, 2011. ACM.

[2] L. Berti-Equille, T. Dasu, and D. Srivastava. Discovery of complex glitch patterns: A novel approach to quantitative data cleaning. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 733–744, Washington, DC, USA, 2011. IEEE Computer Society.

[3] T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley, New York, 2003.

[4] T. Dasu and J. M. Loh. Statistical distortion: Consequences of data cleaning. *PVLDB*, 5(11):1674–1683, 2012.

[5] J. M. Hellerstein. Quantitative data cleaning for large databases, 2008.

[6] K. Kleisouris, B. Firner, R. Howard, Y. Zhang, and R. P. Martin. Detecting intra-room mobility with signal strength descriptors. In *Proc. of the 11th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '10, pages 71–80, 2010.

[7] H.-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. In *PAKDD*, 2009.

[8] E. Levina and P. Bickel. The earth movers distance is the mallows distance: some insights from statistics. *Proceedings Eighth IEEE International Conference on Computer Vision ICCV 2001*, 2:251–256, 2001.

[9] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, New York, 1987.

[10] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[11] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 59–, Washington, DC, USA, 1998. IEEE Computer Society.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40:99–121, November 2000.

[13] S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. *Computer Vision and Pattern Recognition, IEEE Computer Society Confere nce on*, 0:1–8, 2008.

[14] D. Tse and P. Viswanath. Fundamentals of wireless communications, 2004.