Detecting Network Performance Anomalies with Contextual Anomaly Detection

Giorgos Dimopoulos[*], Pere Barlet-Ros[*], Constantine Dovrolis[†], Ilias Leontiadis[‡]

[*]UPC BarcelonaTech, Barcelona, {gd, pbarlet}@ac.upc.edu
[†]Georgia Institute of Technology, constantine@gatech.edu
[‡]Telefonica Research, Barcelona, ilias.leontiadis@telefonica.com

***Abstract*—Network performance anomalies can be defined as abnormal and significant variations in a network's traffic levels. Being able to detect anomalies is critical for both network operators and end users. However, the accurate detection without raising false alarms can become a challenging task when there is high variance in the traffic. To address this problem, we present in this paper a novel methodology for detecting performance anomalies based on contextual information. The proposed method is compared with the state of the art and is evaluated with high accuracy on both synthetic and real network traffic.**

I. Introduction

The detection of network performance anomalies such as loss, delay and outages has always been of extreme importance for network operators. The capability to accurately identify anomalous behaviours, allows operators to pin-point the offending part of the network and perform root-cause analysis to troubleshoot the underlying problem.

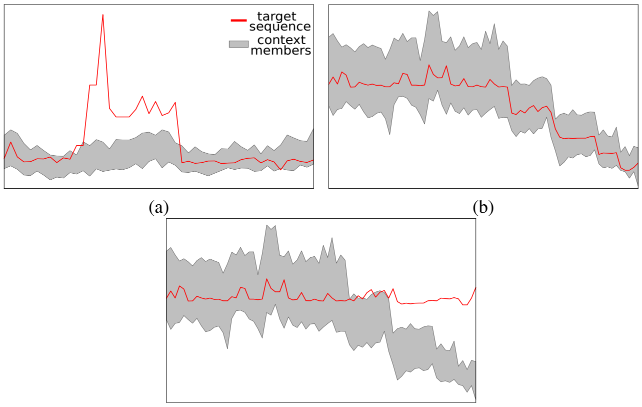
The performance of an entity or a segment in a network can be identified as anomalous, if its behavior deviates significantly from a predefined normal profile. Popular methods in the state of the art, define the normal profile based either on the previous behavior of a target sequence e.g. CUSUM [1], or the variance of the entire dataset e.g. PCA [2].

However, these approaches may fall short when individual paths are characterized by natural high variance or when different normal profiles can be found when examining different regions of the network. Behaviors like these can be observed in real-life scenarios such as the increased latency in a path of an ISP's core network during peak hours or the high packet loss due to an outage over a wide area caused by a natural disaster.

To minimize the probability of undetected anomalies or false alarms in these cases, it is necessary to perform anomaly detection while taking into consideration the behavior of the context that a measured entity or path belongs to. A context corresponds to a peer group where the members have similar behavior with the target in the same time window.

The graphs in Figure 1 show real examples of a performance metric of a target in time series format (red), while the grey area shows the sequences that form the context.

In Figures 1a and 1c the target sequences deviate from their context while in 1b the target follows the context's behavior. Previous detection methods may identify the red sequences in 1a and 1b as anomalous but not 1c. However, if we take the context's behavior into consideration, 1a and 1c should be detected as anomalous but 1b should not.



(a) (b) (c)

Fig. 1: Examples where the target sequence follows its context (b) and deviates from it (a), (c).

To address this problem, we propose the use of the Contextual timeseries Anomaly Detection (CAD) methodology which allows the detection of anomalies based on contextual information with higher accuracy, while at the same time minimizing the false alarm rate.

This solution can directly benefit operators by reducing the number of generated support tickets and calls since individual performance issues can be identified and addressed as soon as they occur and before they affect a larger part of the network.

Specifically, this paper makes the following contributions:

- We introduce a novel methodology for detecting contextual network performance anomalies.
- We present the benefits of detecting anomalies in network measurements using CAD.
- We propose methods to improve the state-of-the art algorithm in terms of accuracy and scalability.
- We evaluate the algorithm with synthetic and real data.

II. Contextual Anomaly Detection

This section presents the two distinct stages of the proposed methodology, i.e. the context construction and the anomaly detection phase.

A. Context Construction

The purpose of this phase, is to cluster together all instances that exhibit similar temporal characteristics across a longer period of time. This results in grouping together into a single context all timeseries that have similar temporal variances within a selected construction time window T_c and makes it easier to later understand deviations from the nominal context behavior.

In order to construct the context, it is necessary to calculate the pair-wise similarity between the instances in the dataset. To accomplish that, we need to employ an accurate timeseries distance measurement method.

However, when dealing with network performance measurements, probe failures, outages or irregular sampling rates may lead to sequences with missing samples, while differences in speed between sequences can occur due to the propagation delay of an anomaly. As a result, traditional methods such as the Euclidean distance cannot be relied upon for accurately calculating the distances between instances.

To address this issue, the pair-wise similarity between instances is calculated using Dynamic Time Warping (DTW) [3]. DTW is used to determine the distance between two sequences that may vary in speed, by warping them in the time dimension in order to properly align them and get a more accurate measure of their distance.

DTW is a very suitable solution for such cases since it can aid in reducing the number of False Positives (FP) and False Negatives (FN) and lead to higher accuracies as we will show in more detail in Section IV-A.

We specifically use the FastDTW [4] implementation of the algorithm which has an $O(N)$ complexity as opposed to the $O(N^2)$ complexity of the original implementation.

After the distances between all the sequences are calculated for a given T_c, the k-Nearest Neighbors (kNN) algorithm is used to classify them into different contexts. kNN has been proven to be a very effective solution when classifying time series sequences based on their distances. Moreover, the value of k is set equal to 1 in order to minimize the model's bias and increase the confidence of the predictions.

B. Anomaly Detection

After clustering a given instance into a context of similar instances based on its behaviour on a large time window, we want to examine whether it deviates from the average context behaviour for shorter periods of time.

Two generic examples of contextual anomalies are shown in Figures 1a and 1c, where although the red sequences were previously identified as members of their context during the context construction phase for a larger time window, smaller parts of the sequences exhibit a very different behavior from the respective context.

We therefore propose the following methodology for identifying any context members that show anomalous behavior within a given scoring time window T_s such that $T_s \leq T_c$.

Initially, we keep only the members of the context that were identified as TP in the context construction stage. Next, DTW is used to calculate the distance matrix of all the members of the context C_{T_s} for the new scoring window T_s. Finally, we define the Context Mean Distance (CMD) as the average value of the C_{T_s}.

Contextual Anomaly Definition: A context member O_{T_s} is identified as anomalous if its average distance from the rest of the context members is larger than the CMD plus one standard deviation.

$$\overline{dist(C_{T_s}, O_{T_s})} \geq \overline{dist(C_{T_s})} + \sigma(dist(C_{T_s})) \qquad (1)$$

One of the most notable advantages of this approach is that it allows the detection of all the anomalies in a context for a specific scoring window in a single step. In other words, it is not required to evaluate each context member separately for anomalies, which can result in great performance benefits, specially when dealing with large contexts.

III. Datasets

A. FCC Data

The algorithm's performance is evaluated with the dataset obtained from the FCC Measuring Broadband America project [5]. The project aims to measure and report the performance of 13 major U.S. fixed and mobile broadband providers.

The measurements are collected with the aid of SamKnows [6] platform, where active measurement probes are installed in the home networks of broadband clients. The probes measure the performance of the last-mile by running tests against servers located in the providers' core network or that are part of the SamKnows infrastructure.

The FCC data contains 13 performance metrics which can be found in the related technical appendix [7]. For the evaluation we obtain a copy of the average RTT measurements which consists of measurements collected over 8 consecutive months in 2015. Apart from the average RTT, the data also includes the whitebox's ID, the server's FQDN, a location ID and the number of successful and failed tests.

Before doing the evaluation with the FCC data in Section V, we run a preliminary analysis of the dataset in order to verify that there is correlation between the instances based on the context they belong to.

To this end, we group all instances in the data based on the server domain that was used to perform the measurements. Hence, each group corresponds to all clients that are connected to the same server. Clients connecting to the same server share the same network provider and are located in the same geographical area. These peer groups are equivalent to contexts since the members in each group are expected to have similar performance characteristics.

Next, for 5 randomly selected servers we calculate the average and the standard deviation of the DTW distances among the members in the same context and the distances of the members with all the instances outside the context.

Figure 2 illustrates the average in- and out-of-context distances and their standard deviation. In all the 5 cases there is higher similarity among instances of the same context. In contrast, when comparing with out-of-context instances the distances are much higher in comparison. This shows that peer groups where the members have a clear correlation with

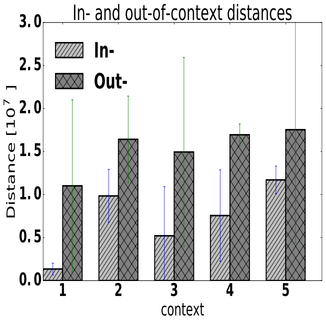


Fig. 2: Average and std. deviation of the in- and out-of-context DTW distances.

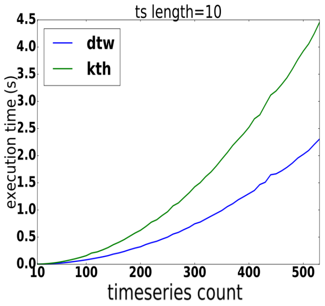


Fig. 3: K_{th} vs. DTW execution t while increasing the sequence number.

each other do exist in the FCC dataset, which makes it very suitable for evaluating our methodology since our algorithm aims to automatically group together such behaviors based on the DTW distance.

B. Synthetic Data

In general, it is typical not to have ground truth in real data and the FCC data is no exception to this rule. A common approach to address this in order to properly evaluate the algorithm's performance, is to use an artificially generated dataset where the ground truth is known. Such a dataset, needs to be created in a way that it accurately represents real network measurements, to ensure that the evaluation results can be generalized to real data.

The synthetic data should consist of multiple contexts for the purpose of evaluating the algorithm's context detection capabilities. To find the most accurate model and the correct configuration parameters for creating each context, we take the average RTT measurements for the month of August and group by the server hostname. In this way, each group represents a different context with RTT measurements from multiple clients against the same server.

To identify the statistical model that best fits the data in each context, we apply the Goodness of Fit (GoF) methodology which allows us to compare the distribution of the data with other well-known distributions. The metric used to determine the GoF is the Sum of Squared Errors (SSE).

The GoF was performed for the contexts that correspond to the ten servers with the largest number of clients and measurements. The ranking of the best fitting distributions based on the SSE score, revealed that the model which best describes the data in each context is the $Johnson's\ S_U$ [8].

Next, each set of parameters that were obtained from fitting each context is used for generating equal number of synthetic contexts. More specifically, every context consists of 100 time series that were created using Johnson's S_U and the corresponding settings. All the time series contain 1 sample per hour and a total length of 1 week.

C. Synthetic Anomalies

In order to properly verify the algorithm's anomaly detection capabilities it is necessary to know which sequences are anomalous. However, the data obtained from FCC do not contain any information about the existence and duration of anomalies. For this reason, we perform the evaluation using synthetic anomalies that we manually inject into the dataset. Specifically, we introduce two types of anomalies that simulate the variations in the RTT measurements of under-performing network links, i.e. level shift and standard deviation shift anomalies.

Level shift anomalies correspond to events where for the duration of the anomaly, the mean of the time series is increased or decreased by a certain fraction of its value before the anomaly occurred. Examples of such events can be routing changes where a longer path is used and as a result the mean of the overall latency is increased by a certain amount.

Shifts in the standard deviation of the RTT measurements of a link can be observed in congested paths. In such scenarios, the minimum delay of the path that is determined by the overall propagation delays remains unchanged. However, the added queuing delays cause increase in the variability of the delay distribution towards the larger values only, resulting in a shift of the standard deviation.

IV. Evaluation with Synthetic Data

A. Context Construction Evaluation

For the purpose of comparing the performance and accuracy of DTW with the state of the art in CAD, the evaluation is performed with two different distance metrics, first using the K_{th} distance and then with DTW.

The K_{th} order statistic distance was presented in the work of Chen et al. [9], as a method to reduce the FP and FN when using the Minkowski distance to construct the context of a target time series.

Before doing the evaluations, each sequence is labeled according to the context it belongs to in order to provide the context construction ground truth. Then, 80% of the data is used for training the classifier and 20% for testing.

The two evaluations are repeated three times, each time modifying one of three variables, i.e. the number of context members, the context count and the construction window T_c. Each time a variable is gradually increased while the other 2 remain fixed.

Moreover, the data for each context is generated using 20 different settings in order to eliminate the possibility to get results that are specific to a particular configuration.

K_{th} **Distance vs. Dynamic Time Warping:** In this part and throughout the rest of the paper, the accuracy is expressed using the f1-score, which corresponds to the harmonic mean of the Precision and Recall and is calculated as shown in (2).

$$f1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2}$$

Figure 4 shows a comparison of the accuracy achieved using the two methods. Both graphs show the evolution of the f1-score when using the K_{th} distance and when using DTW, as the number of contexts increases for T_c = 7 days (left) and for 30 days (right).

In both cases the accuracy improvement when using DTW is evident. When the context count is equal or greater than 3,

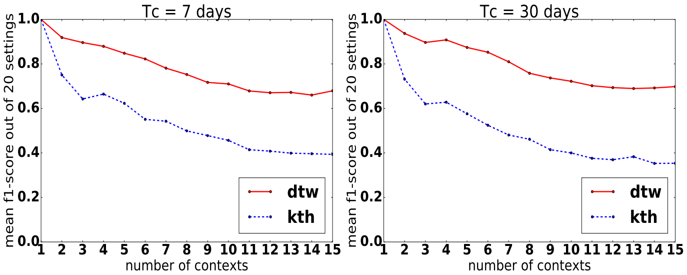


Fig. 4: mean f1-score comparison for context construction with k-NN using K_{th} distance and DTW as distance metrics. T_c = 1 week (left) and 1 month (right)

the accuracy gain is constantly above 20% and can reach up to 35% for $T_c = 7$ days.

The reason why the K_{th} distance is outperformed by DTW, is because the alignment of the two compared time series is done by rearranging their samples based on their point-wise distance. This can change the shape and the sample order of the sequences and return a less accurate distance value.

In contrast, DTW performs time warping to find the optimal alignment between the sequences without sample rearrangement. Hence, when compared to K_{th}, DTW is a more reliable distance calculation method.

The next part of the comparison is done with regards to the execution time of the two algorithms. Both algorithms were implemented in Python and all the tests were executed on a Linux PC with an Intel Core i7 @ 3.4GHz.

Figure 3 shows the execution time comparison when increasing the number of instances in the dataset while keeping a fixed time series length. Here, we observe that DTW constantly outperforms the K_{th} distance and when the time series count reaches 500, DTW is already twice as fast.

Context Construction with 1NN-DTW: The first phase of the context construction evaluation is performed with an increasing context size, while the number of contexts are fixed to 5 and the T_c window is set to seven days. The number of members in each context is increased from 10 to 100 in steps of 10. In Figure 5 (left), the thick black line shows the mean f1-score obtained from the 20 different settings that were used in each of the steps. The grey dashed lines represent the mean ± one standard deviation.

These results show that the overall accuracy is improved as the context size is increasing. Specifically, we find that a 10% improvement is achieved when comparing the accuracy between the min and the max context size.

Contexts with larger number of members are identified more accurately, due to the addition of more observations to the training set without increasing the complexity of the data since each new observation has common characteristics with the other members.

Next, the evaluation is repeated with an increasing number of contexts, while the context size is fixed to 100 members. The results from this evaluation that can be found in Figure 5 (right), which shows that there is a negative correlation between the accuracy and the number of contexts.

More specifically, we find that by increasing the context number, the complexity of the dataset increases as well. This has a negative impact on the overall context detection accuracy since the algorithm has a more difficult task to classify larger number of observations with different characteristics.

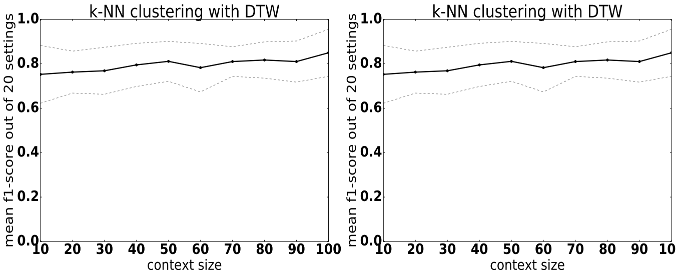


Fig. 5: Accuracy of the context construction with increasing context size (left) and increasing context count (right)

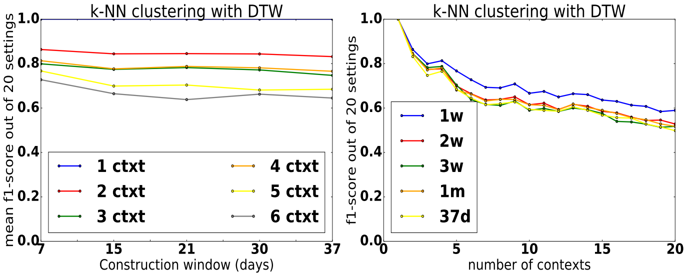


Fig. 6: Accuracy of the context construction when increasing the T_c and the number of contexts.

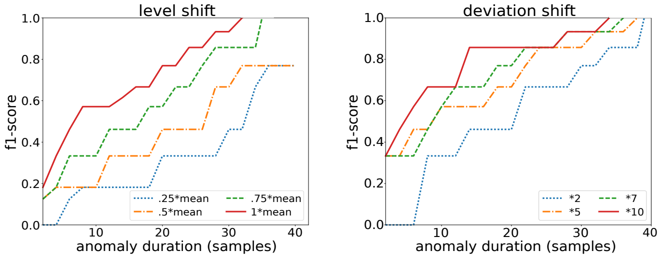
The last phase of the evaluation is performed while modifying the length of T_c with values equal to 7, 15, 21, 30 and 37 days. The plots in Figure 6 show the accuracy for each T_c value while the number of contexts is increased. Both plots show that the increment of the construction window has a very small effect on the accuracy. This result is logical since a time series in a context is synthesized with the same settings and the window size only determines then number of samples that the series will have.

B. Anomaly Detection Evaluation

As discussed in Section III-C, we evaluate the anomaly detection capabilities of the algorithm with two types of artificial anomalies that are introduced to the data. We measure the detection accuracy with the f1-score which is equal to the harmonic mean of the the Precision and Recall metrics. The f1-score is then used to monitor the detection performance while modifying the duration or the amplitude of the introduced anomaly.

In more detail, we randomly select one of the contexts that were created in the context construction phase. Next, we run the anomaly detection and remove any detected anomalies that exist before injecting the artificial ones. In this way, we ensure that the context does not contain anomalous events that may affect the detection accuracy.

More specifically, 10 anomalies of each type are generated for each step of the evaluation and next, the evaluation is performed while gradually incrementing first the duration of the anomalies and next their amplitude. This approach will show the correlation of the detection accuracy with the different anomaly durations and levels of severity.



(a) (b)

Fig. 7: Detection accuracy for level shift (a) and deviation shift (b) anomalies.

In the case of the level shift, the amplitude is controlled by adding or subtracting a fraction of the time series' mean. For the deviation shift anomalies, the amplitude is modified by multiplying all the points of selected region of the time series with values higher than the mean, with a variable fraction of the entire sequence's standard deviation.

Figures 7 (a) and (b) illustrate the performance for the evaluation with the two anomaly types. The duration of the events is counted in number of samples as the actual duration of the anomaly depends on the measurement frequency. In both figures we have four graphs that correspond to equal number of intensity levels.

For the level shift anomalies, these intensity levels are the fractions of the time series' mean that is added in order to generate the anomaly. In our evaluation we specifically select increments equal to 25, 50, 75 and 100% of the mean. The deviation shift anomaly level is modified by multiplying all points of the time series that are above the mean with a given integer, which will result in the increase of the upper part of the standard deviation only. In this evaluation we created four different levels by multiplying with 2, 5, 7 and 10.

From the two figures we observe that for both types of anomalies the algorithm is capable of achieving 100% accuracy for anomalies with duration equal or higher than 30 samples. For different levels of intensity we see that the deviation shift anomalies are overall detected with higher accuracy. Even for very small changes in the deviation, the accuracy is increased significantly. This is an expected finding given that the detection algorithm uses the standard deviation as a threshold and therefore changes in the deviation of a series are picked-up more easily. Furthermore, the detection of the deviation change anomalies with shorter duration outperforms the level shift as well. This is attributed to the fact that changes in the time series' mean need to be longer in duration in order to have a significant impact on the characteristics of the series.

V. Evaluation with the FCC Data

A. Context Construction Evaluation

The evaluation with the FCC data is done using the average RTT measurements from August 2015 and following the same approach as with the synthetic data. The evaluation is performed for different number of contexts, while the construction window is fixed to 1 week. Each context is defined as the collection of time series that correspond to measurements against the same server. In this way the context represents clients in the same geographical area that are using the network of the same provider.

In contrast to the evaluation with the synthetic data, we do not initially modify the number of members in each context. The number of members is adjusted when applying class balancing by means of under-sampling before the training phase. A balanced training set where all the classes have equal number of instances is necessary for creating a model that is not biased by under- or over-represented classes.

Next, we evaluate again using 1NN-DTW and we compare the benefits from using DTW over the K_{th} distance. The process is repeated while increasing the number of contexts in the dataset from 1 to 20 (Figure 8 (left)) while the construction window is fixed to 1 week.

From the figure we see that the use of DTW always results in a higher accuracy score as compared to the K_{th} distance. More specifically, the improvement from using DTW instead of K_{th} can reach up to 20% while the average accuracy gain throughout the evaluation is approximately 11%.

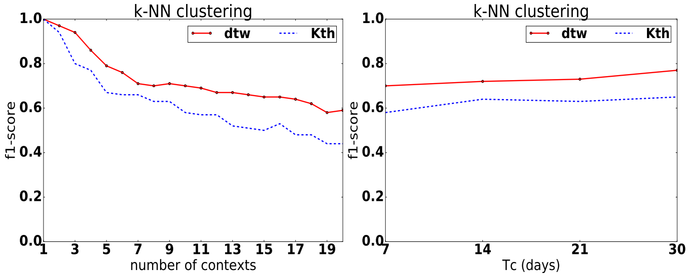


Fig. 8: Context construction evaluation using k-NN with k_{th} distance and DTW, while increasing the number of contexts (left) and the construction window (right).

The second phase of the evaluation is performed with a fixed number of 10 contexts and an increasing T_c from 7 to 30 days in 7 day steps. Figure 8 (right) shows the accuracy in each step for both DTW and K_{th} distances.

Again the results indicate that there is significant improvement in the context construction accuracy when using DTW. Similar to the findings in the synthetic data evaluation, we find that the length of the construction window has a small impact on the overall accuracy. Moreover, we see that the accuracy is improving when the T_c is increased, while a T_c = 30 days can lead to approximately 10% accuracy gain when compared to the respective result for T_c = 7 days.

The performance increase for larger time windows which was also observed in Section IV, is attributed to the information gain obtained by introducing longer sequences. This allows a more precise reconstruction of the context by the classifier, since the distances are calculated more accurately when considering a larger part of the sequences.

Overall, the results in this section show that the context construction can be performed successfully with real network measurements and maintain satisfactory accuracy even when the dataset consists of a large number of contexts. Moreover, we verified with the FCC data as well that there are significant improvements in accuracy when using DTW instead of the K_{th} distance.

B. Anomaly Detection Evaluation

The detection evaluation with the FCC data is following the same steps as in Section IV-B. We again inject level and deviation shift anomalies and we evaluate the detection accuracy for different intensities and anomaly durations.

Figures 9 (a) and (b) show the results from the evaluations with the two anomaly types. The overall accuracy in both cases is high although slightly lower than the respective findings in Section IV-B. This is expected, since the synthetic contexts have less variation and thus the anomalies are easier to detect even when the deviation from the context is smaller.

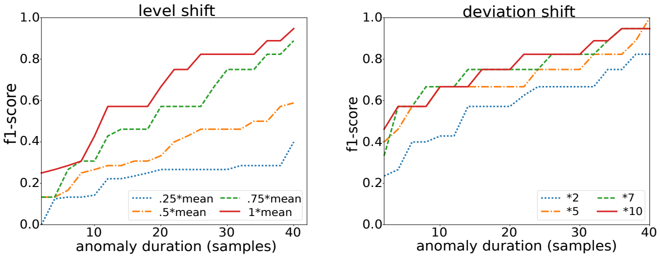


Fig. 9: Detection accuracy for level shift (a) and deviation shift (b) anomalies.

Here, the level shift anomalies are identified with better accuracy for the two higher intensity levels but the performance is reduced for the remaining two levels. In all cases the accuracy is improved significantly as the duration of the anomaly is increased. However, the level shift anomalies are simulating performance issues due to path changes. These issues are rarely very short in duration and therefore we can assume that in real scenarios the algorithm will be able to detect them with high accuracy. Moreover, we observe that the deviation shift anomalies which correspond to underperforming links due to congestion, can be identified with very good accuracy even if these events are very short-lived. This feature can be very beneficial in the wild, since congestion can result in short but intense variations in the latency of the affected network segment.

VI. Related Work

This section covers two categories of related works, i.e. those that although not related to network performance anomalies, they deal with CAD and those that deal with anomaly detection specific to network performance.

CAD in Other Fields: To the extend of our knowledge, this is the first work that uses contextual information for network performance anomaly detection. Nevertheless, the concept of context-based detection algorithms is not new and has been presented in a few different fields in the past.

The paper from Chen et al. [9], which is the most related to our work, presents a contextual change detection approach that uses the K_{th} distance for context construction and the TAD metric to detect changes. In contrast, we use DTW distances and standard deviation for the anomaly detection respectively.

CAD has also been applied in big sensor data [10], where point anomalies are identified using a univariate Gaussian predictor, while the k-means-based contextual detection is used as a post-processing step. [11] used a prediction-based CAD for detecting stock market manipulation. Here, instead of using a time series' historical data to predict future values, predictions are made from contextual information.

Network Performance Anomaly Detection: Although there is extensive work previously done to cover network security and network intrusion detection, significantly fewer articles have dealt with network performance anomalies. Here we present a few notable related publications in this field.

In [2] Lakhina et al. use PCA on backbone network traffic to capture the variance of anomalous time series. Other statistical methods such as Kalman filters in [12] and wavelets in [13] and [14] were also successfully used to perform anomaly detection. However, in contrast to our work none of the these methods takes into consideration the contextual information when identifying anomalies.

VII. Conclusions

In this paper we have presented a novel approach for detecting network performance anomalies using contextual information. We have shown that not only this method can be successfully applied in both synthetic and real network traffic, but it also offers improvements in terms of detection accuracy and performance when compared to the state of the art algorithms. Finally, in the evaluations with both the synthetic and the FCC data, we found that the CAD methodology can be effectively used to detect performance issues such as path changes and congestion with high accuracy.

VIII. Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness and EU FEDER under grant TEC2014-59583-C2-2-R (SUNSET project) and by the Catalan Government (ref. 2014SGR-1427). This work was also supported by a research grant from the National Science Foundation (award number 1440585).

References

[1] Wang H. et al. “Detecting SYN flooding attacks”. In *INFOCOM*, 2002.

[2] Lakhina A. et al. “Diagnosing network-wide traffic anomalies”. In *ACM CCR*, 2004.

[3] Berndt D. et al. “Using Dynamic Time Warping to Find Patterns in Time Series”. In *KDD workshop*, 1994.

[4] Salvador S. et al. “Toward accurate dynamic time warping in linear time and space”. *Intelligent Data Analysis*, 2007.

[5] “FCC Measuring Broadband America 2015”. https://goo.gl/qWTfDD.

[6] “SamKnows: The global platform for internet measurement”. https://www.samknows.com/.

[7] “FCC Measuring Broadband America 2015 Technical Appendix”. https://goo.gl/Hch7jY.

[8] Johnson N. “Systems of frequency curves generated by methods of translation”. *Biometrika*, 1949.

[9] Chen X. et al. “Contextual Time Series Change Detection”. In *SDM*, 2013.

[10] Hayes M. et al. “Contextual anomaly detection in big sensor data”. In *IEEE BigData*, 2014.

[11] Golmohammadi K. et al. “Time series contextual anomaly detection for detecting market manipulation in stock market”. In *IEEE DSAA*, 2015.

[12] Soule A. et al. “Combining filtering and statistical methods for anomaly detection”. In *ACM IMC*, 2005.

[13] Barford P. et al. “A signal analysis of network traffic anomalies”. In *ACM SIGCOMM IMW*, 2002.

[14] Huang P. et al. “A non-instrusive, wavelet-based approach to detecting network performance problems”. In *ACM SIGCOMM IMW*, 2001.