Towards automated and proactive anomaly detection in a fiber access network

Henrik Abrahamsson
RISE

Daniel Henriksson
Lunet

Karoly Makonyi
Savantic

David Menéndez Hurtado
Savantic

Johan Sandell
Waystream

Abstract—**Communication networks are vital for society and network availability is therefore crucial. There is a huge potential in using network telemetry data and machine learning algorithms to proactively detect anomalies and remedy problems before they affect the customers. In practice, however, there are many steps on the way to get there. In this paper we present ongoing development work on efficient data collection pipelines, anomaly detection algorithms and analysis of traffic patterns and predictability.**

I. Introduction

For a network operator it is important to automatically and quickly detect network traffic anomalies in order to minimize the possible impact on network performance. There is a lot of research going on in the area of network anomaly detection [1], [2], [3], [4], [5], but the problem of automatization remains challenging.

The reason is, in part, because network traffic is unpredictable: access patterns change due to events online, holidays, or even pandemics [6], but it can even be stopped by sudden power outages, fiber cuts and attacks. It also changes over the years as new applications, protocols, and technologies are introduced. Network traffic is also well-known for being bursty, especially on short timescales [7], [8].

Finding a characterisation that is stable and predictable would be very useful from an operational point of view, since a deviation from a forecast is a warning that something might be wrong, and a corrective action is required.

This paper presents ongoing research in the city network operator Lunet's fiber access network where we collect large amounts of telemetry data from Waystream switches in order to:

1) Gain insights into how data collection can be done in the best way. Find relevant tools and implement effective data collection pipelines.
2) Test and evaluate machine learning algorithms to detect anomalies.
3) Add labels to the training data for specific network issues so that we can create fingerprints for these error situations. The fingerprints can then be used in the analysis tool for real time error detection.
4) Analyse traffic patterns and predictability to identify traffic characteristics that are stable and predictable and where a deviation is an indication that something is wrong.

The expected results of the research are increased system availability and a reduced overall cost of network operation, where less time is spent on manual troubleshooting.

We start describing our data in Section II. Then, we describe in Section III our work on autoencoder-based multivariate anomaly detection. In Section IV we present results from an initial study of traffic predictability and the effects of seasonalities.

II. The network and telemetry data collection

Lunet builds and operates the optic fiber network in the city of Luleå, and its goal is to create a future-proof, high-speed, and reliable internet infrastructure for all the residents and companies in the city. Lunet builds its network with state-of-the-art technology including network switches from Waystream. The city network currently consists of more than 1500 access switches that serve more than 22000 households, or 8 out of 10 residents, in all districts of Luleå and 35 nearby villages. This is an open access network, so different ISPs can provide services of Internet and TV through it.

The network has a ring topology where network switches are connected in access node rings through 10 Gbit/s connections, which in turn are connected to core nodes. Downstream from each switch there are 1 Gbit/s connection to the households.

Waystream has recently introduced a set of over 1500 measurements in their products that can provide a deep insight into the health and quality of the network, including packet transmission, service quality, environmental status, memory consumption, optical levels, and more.

A. Pipeline for data collection and the collected data

Continuously collecting and storing detailed telemetry data from hundreds of switches presents a significant challenge. In our ongoing field trial, we have implemented a data processing pipeline using Kafka[1], Telegraph[2], and the time-series database InfluxDB[3], as outlined in Figure 1

We are currently collecting data from roughly 550 switches that provide internet to more than 13000 households. The switches report telemetry data every 10 or 60s, depending on the individual switch, which results in roughly 20GB of data per day. The data collection started in June 2020.

[1] https://kafka.apache.org

[2] https://www.influxdata.com/time-series-platform/telegraf/

[3] https://www.influxdata.com

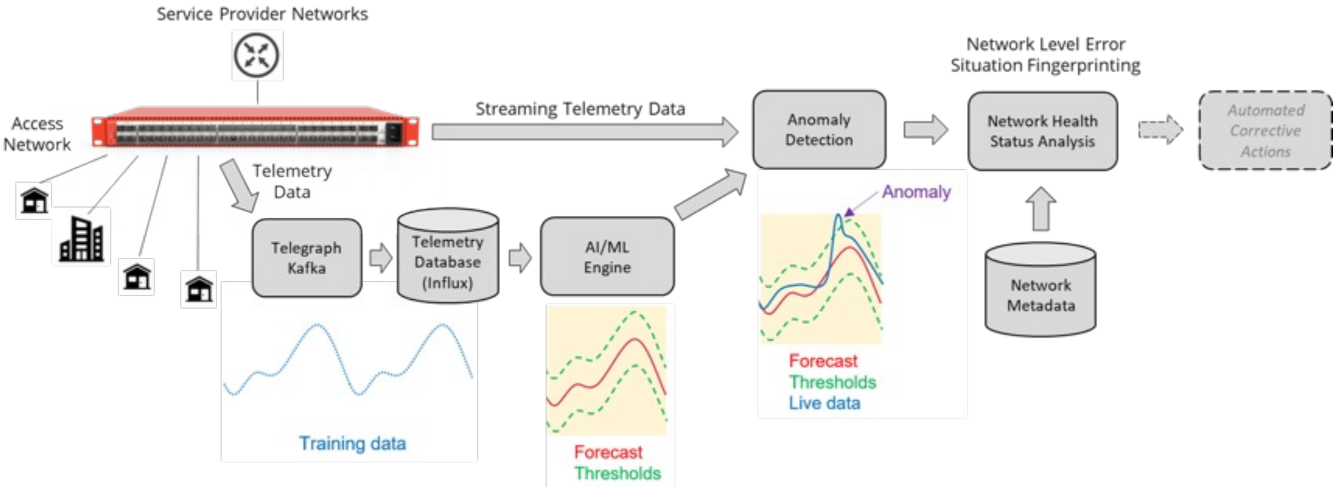


Fig. 1: Overview of telemetry data collection and anomaly detection.

III. Multi-variate anomaly detection

A. Algorithms

We developed a multivariate instantaneous anomaly detection based on autoencoders: given a vector describing the current status of the network, try to predict itself. We expect the machine learning model to be able to faithfully reconstruct points that are similar to the training data, while struggling with those that are uncommon. The reconstruction error can then be used as an anomaly score. Each data point includes all the selected measurements (bit rates, histogram of packet sizes, and broadcast and multicast speeds) for all the switches in a ring, transformed with $\log(1+x)$ to get a roughly normal distribution. The data is then stored in a 3D array, where the dimensions are (time, port, feature) and fed into the network. The first layer of the model takes the 2D inputs (port, feature), applies batch normalisation, and 1D spatial dropout [9] with a rate of 0.5. A spatial dropout is similar to standard dropout, but whole feature maps are dropped. The output is projected into 64 dimensions through a 1D convolution of shape (64, 1), the output flattened, and sent through two fully connected layers with batch normalisation and dropout. After that, we reverse the process symmetrically, re-projecting into a a 64 dimensional space per port, and finally predicting the inputs again. The non-linearity in each layer is ELU [10], except for the last one that has a linear output. The hyperparameters are the number of neurons in the intermediate layers, the L2 regularisation of all the layers – except for the last one –, and the optimiser, which were refined with a grid search with 3-fold cross validation for each ring. The selected architectures for were the largest networks tested (up to 4096 neurons) with the RMSProp optimiser, and a small amount of dropout (0.05-0.1). At inference time, anomalous points will have a larger reconstruction error, concentrated in the offending ports if the anomaly is isolated. The same thing can be done grouping at a switch level: for each switch, all the features of all the ports are stacked together, in order to identify anomalies at a switch level. The architecture remains unchanged. We can also increase the sensitivity at the cost of increased computational times of the search sampling with Dropout at inference time, as described by [11]. Anomalous times will show increased variance. This method does not depend on an explicit inclusion of time-related data, such as day of the week or holidays, and yet it doesn't show strong sensitivity to holiday effects. We hypothesise it is because holidays resemble other moments in the training data without need for explicit labelling. As a result the Figure 2 shows the reconstruction error as color code for different ports on different switches (horizontal axis) at given time (vertical axis).

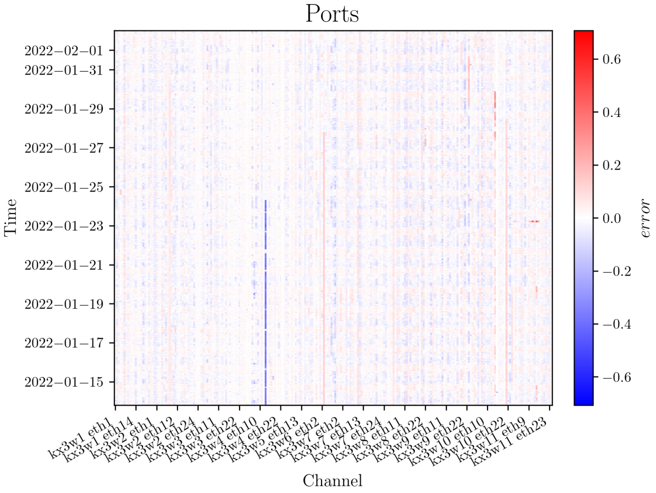


Fig. 2: Reconstruction error of RX/TX bitrates of different switches/ports (X axis) at different time (Y axis). The darker colours indicates higher probability of an anomalous behavior.

IV. Traffic patterns and predictability

In this part of the work we have so far studied the predictability of two metrics: the network activity level (the share

Metric	Method	MAE	MAPE
Network Activity (percentage points)	Seasonal naive	2.12	4.26%
	Prophet	2.38	4.81%
Bitrate (Gbit/s)	Seasonal naive	0.040	7.75%
	Prophet	0.044	8.41%

TABLE I: Results on traffic patterns predictability, as shown in Figure 3

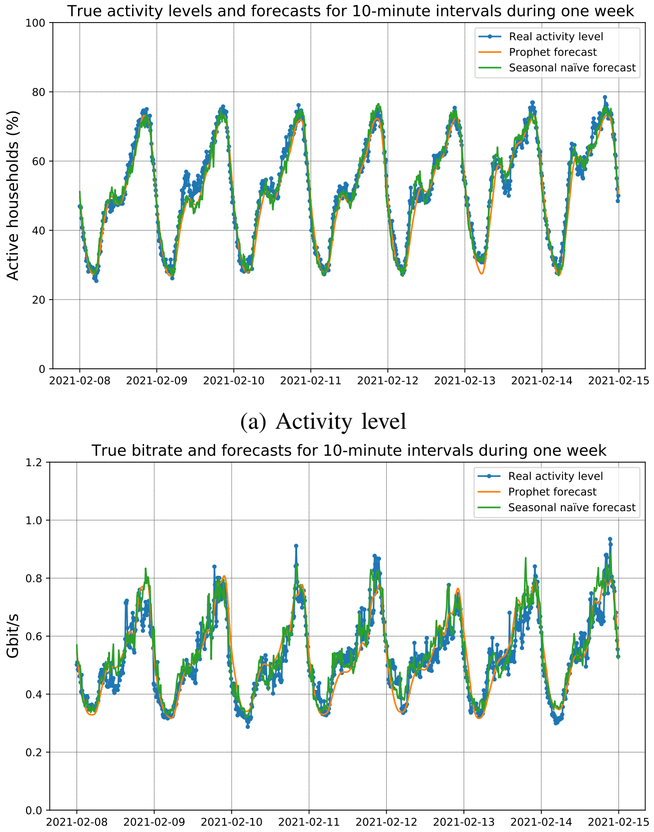
of customers that are active at a given time) and the downstream bit rate from a network switch towards the customers. Our working hypothesis is that forecasts and deviations from the forecasts can be used to detect anomalies.

Network traffic and activity levels are examples of highly seasonal data with typical daily and weekly patterns. We use a seasonal naive forecast method to study and quantify predictability. We make one week ahead predictions of the activity level and the bit rate and evaluate the forecasting errors. There are several possible measures of forecast accuracy. Here we use two commonly used measures suggested in [12]: mean absolute error (MAE) and mean absolute percentage error (MAPE). The seasonal naive forecast is the simplest possible method for forecasting seasonal data but it can be surprisingly effective [12]. The method is often used as a benchmark and is therefore a reasonable method to use to quantify a lower limit of the predictability. A simple example explains the method we use: In order to predict the activity level next Thursday between 18:50-19:00 we consider the activity levels on the previous two Thursdays 18:50-19:00, and use the average value as a forecast. For comparison, we also use Prophet [13], [14] to see if it can capture more information in the data and improve the predictions. Prophet is a time-series forecasting tool developed by Facebook. It was released as open-source in 2017.

In both cases we have studied a small subset of the data described in Section II covering 26 weeks (Monday January 18 2021 - Sunday July 18 2021). For the activity level, we considered 260 households on timescales of 10 and 60 minutes, where we define “active” as receiving more than 0.5 Mbits/s. For the bitrate, we considered a single switch. Figure 3 shows one week as an example, with the error metrics collected in table I.

The weeks with the worst forecasts are the ones with national holidays, Easter and Midsummer in this case. The activity levels mirror human habits and behavior, and we behave differently when there is a holiday compared to a working day. In future work, we plan to study data from longer time periods and consider a yearly seasonal naive model where the activity levels during, for example, the Easter holidays are predicted based on the previous Easter.

The activity level is interesting because it is the customers who create the traffic in the access network, and a sudden drop in the activity level may indicate a problem in the network. We have considered here only 260 households, but it is reasonable to believe that more households and a higher level of aggregation would reduce the effect of random variations in the data and provide better predictions or allowing for shorter timescales. We can also build it in a hierarchical fashion: if there is a deviation – for example, an overall drop of activity – we could analyse each individual ring and switch in order to pinpoint the problem. The share of active customers could also be monitored based on what service provider they belong to, in order to quickly identify if a problem is related to a specific operator rather than to the fiber infrastructure of the network. Future work on this area also includes considering other traffic measurements that are of interest.



(a) Activity level

(b) Bit rates

Fig. 3: Data and predictions during week 6, 2021. Note how activity levels are noticeably easier to predict than the more bursty bit rates. See metrics in Table I.

V. Discussion and future work

Today we see an ongoing digital transformation of society, where people and society heavily rely on functioning communication networks. Communication networks are vital for society and network resilience is therefore crucial. In this paper we have presented ongoing research towards automated and proactive network anomaly detection. The basis for the research is an ongoing field trail in the Lunet fiber access network, where we currently collect telemetry data from 550 switches. We continuously add more switches, with the goal

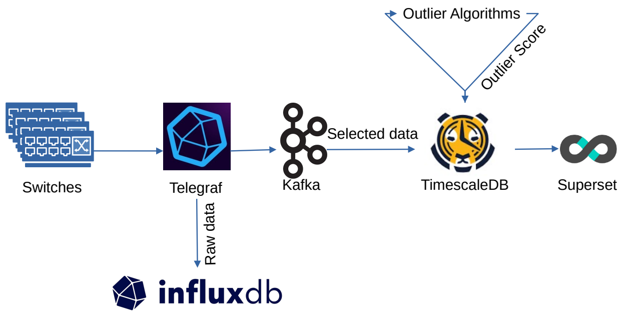


Fig. 4: Planned data processing pipeline for anomaly detection and visualization.

to reach one thousand, to study the scalability of the data collection pipeline.

Work is ongoing to extend the data collection pipeline with machine learning and real-time anomaly detection as outlined in Figure 4. The work includes adding TimescaleDB[4] to store data that is used for anomaly detection and visualization, and to evaluate the capabilities and usability of Apache Superset[5] as a graphical frontend.

Concerning traffic predictability and forecasts, we have so far studied network activity levels and bit rates only for a small subset of switches and household ports for which data in continuously collected. We plan to extend the data analysis to cover a larger part of the network and longer periods of time. We also plan to study the predictability of additional traffic characteristics including packet size distribution and the ratio between ingoing and outgoing traffic in the network. Furthermore, we will use the forecasts and deviations from the forecasts to construct anomaly scores. Part of the ongoing field trail we plan to calculate forecasts and anomaly scores in real-time and evaluate scalability and performance.

Another topic for future work is to collect more labeled data of network anomalies. To evaluate the accuracy of anomaly detection techniques such as forecasts and unsupervised autoencoders a tool is needed where the operator easily can label network event as anomalies [15]. Supervised machine-learning methods also need labeled data for training.

For research and to develop algorithms to automatically detect network anomalies, collecting large amounts of telemetry data is essential. An interesting question for the future (when more knowledge and algorithms are in place) is how much data needs to be collected centrally by an operator (or a network management tool), and what anomaly detection can be done distributed without telemetry data having to leave the switch (for example deviations in bit rate at the switch).

VI. Acknowledgement

This work was supported in part by Sweden's Innovation Agency (VINNOVA) via the project AI-NET-PROTECT (2020-03506).

References

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[3] S. Eltanbouly, M. Bashendy, N. AlNaimi, Z. Chkirbene, and A. Erbad, "Machine learning techniques for network anomaly detection: A survey," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 156–162.

[4] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, pp. 447–489, 2019.

[5] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152 379–152 396, 2021.

[6] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez *et al.*, "The lockdown effect: Implications of the covid-19 pandemic on internet traffic," in *Proceedings of the ACM internet measurement conference*, 2020, pp. 1–18.

[7] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[8] W. Willinger, M. S. Taqqu, and D. V. Wilson, "Lessons from "on the self-similar nature of ethernet traffic"," *SIGCOMM Comput. Commun. Rev.*, vol. 49, no. 5, p. 56–62, nov 2019. [Online]. Available: https://doi.org/10.1145/3371934.3371955

[9] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," 2015.

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2016.

[11] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2016.

[12] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice, 2nd edition*. Melbourne, Australia: OTexts, 2018, OTexts.com/fpp2 [Accessed on: 2023-02-12].

[13] "Facebook prophet," https://facebook.github.io/prophet/ [Accessed: 2023-05-31].

[14] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[15] D. Liu, Y. Zhao, H. Xu, Y. Sun, D. Pei, J. Luo, X. Jing, and M. Feng, "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *Proceedings of the 2015 Internet Measurement Conference*, ser. IMC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 211–224. [Online]. Available: https://doi.org/10.1145/2815675.2815679

[4] TimescaleDB, https://www.timescale.com

[5] Apache Supserset, https://superset.apache.org