2ND WORKSHOP "MACHINE LEARNING & NETWORKING" (MaLeNe) PROCEEDINGS

SEPTEMBER 4, 2023

CO-LOCATED WITH
THE 5TH INTERNATIONAL CONFERENCE ON NETWORKED SYSTEMS (NETSYS 2023)
POTSDAM, GERMANY

PicNIC: Image-based Diagnosis for Industrial Blackbox Networks

Marco Reisacher, Andreas Blenk, Hans-Peter Huth
Siemens AG, Munich, Germany

***Abstract*—This research paper presents two tools, SieMonX and PicNIC, that can help monitor and diagnose faults in black box industrial networks. SieMonX is an agent-based network monitoring and data generation tool, while PicNIC is an image-based end-to-end network diagnosis tool that uses a convolutional neural network to diagnose inaccessible networks through simple delay measurements. These tools provide valuable insight into network infrastructure and can aid operators in making informed decisions. A look into both tools provides insight into the functionalities and the implementation.**

***Index Terms*—Machine learning Network Diagnosis, Industrial Networks, Network Monitoring**

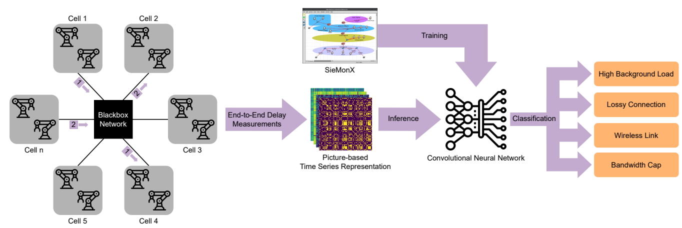

Fig. 1. Training and Inference Pipeline using SieMonX in an Industrial Blackbox Network.

I. Introduction

The Context: In recent years, the convergence of IT and OT networks in industrial environments has become increasingly prevalent. These complex networks involve multiple stakeholders coexisting, making them difficult to monitor and diagnose. Railway networks are a prime example of externally operated networks, which require operators to have insight into foreign networks to diagnose end-to-end connections. However, equipment in these networks is often connected through foreign networks, and there needs to be access to internal network equipment or monitoring data, making it challenging to pinpoint faults.

The Problem: The challenge of monitoring and diagnosing faults in complex industrial networks has been a research topic for many years. Operators often struggle to determine whether a fault is due to broken equipment or an underlying network. This lack of insight can lead to costly downtime and decreased productivity. Dynamic Time Warping (DTW) [1] and deep learning techniques, such as Convolutional Neural Networks (CNNs) [2]–[5]. Moreover, Recurrent Neural Networks (RNNs) [6], [7] are commonly used in time series classification for detecting anomalies, predicting network traffic, and identifying network events. Although these techniques are widely researched, their use case in black box network diagnosis, especially in industrial environments, is a hardly considered topic. This paper presents two tools to address these challenges: SieMonX and PicNIC.

Our Contribution: The contribution of this paper is to provide a comprehensive overview of SieMonX and PicNIC and their potential to address the challenges of monitoring and diagnosing faults in complex industrial networks, as seen in Fig. 1. SieMonX is an agent-based network monitoring and data generation tool for developing data-driven machine learning. It provides a framework for custom monitoring tools or load generators and stores and documents data consistently, making it easy to preprocess. This tool generates valuable data for network analysis and provides a robust foundation for machine learning models to detect faults in the network. PicNIC is an image-based end-to-end network diagnosis tool that uses simple delay-based measurements. This tool visually represents time series data as images, allowing for more straightforward interpretation of network behavior. It uses a Convolutional Neural Network to classify images and detect behavior in the black box network, such as high background load, lossy transmits, and bandwidth restrictions.

II. SieMonX

Motivation: To generate data for training of models or for statistical analysis, often complex tasks or measurement campaigns in testbeds have to be performed. A typical testbed will have several nodes and a communication network in between. Moreover, campaigns must be repeated using different setups to cover a broad range of scenarios. In a non-trivial environment, this can be a time-consuming and error-prone task. Correct and complete documentation of test runs is a must and again adds to the needed efforts. The Siemens Monitoring tool for Experimentations (SieMonX) automizes test campaigns quickly, allowing many test runs with different conditions and storing data consistently. It can even add tags to the data, so manual tagging is not needed in this case. A second problem in those training environments is often that custom functions are needed, which external tools cannot quickly provide. For example, SieMonX includes a simple UDP-based end-to-end monitoring tool that can - on demand - send a short packet immediately followed by a long packet so the effects of a network can be studied under the assumption that both short and long packets receive the same effects. Thus, SieMonX is also a framework for custom monitoring tools or load generators. In this respect, it offers timer handling, a storage service, and more.

State-of-the-Art: Typically, scripts are devolved and deployed in the testbed. This, however requires some development effort and is error prune. A better approach is to have agents on each node and one controller. This can be achieved by several automation tools (e.g., Ansible) or by monitoring tools (e.g., Nagios). However, these tools are built for other purposes, so it is still needed to develop scripts or plugins, and limitations may occur. Moreover, these tools do not address automatic tagging or data consolidation, which requires additional scripting. SieMonX, as a specialized tool, is easier to use and only needs one straightforward high-level script to perform a campaign. There is no need to change something on the nodes remotely.

Architecture: SieMonX has two components: a controller and agents. The controller is the user interface and sends commands to the agents, performing the needed actions. The agents need a simple configuration, the controller's IP address, and a directory for storing data. Upon start, the agent then registers at the controller. A researcher then can issue commands using the console of the controller or can start a simple script to initiate several commands in the testbed at once. This script can add supplementary info to the data from the test run, e.g., configuration data. Each measurement campaign stores its data in an individual directory, together with automatically produced metadata. Thus, preprocessing can quickly identify the correct data set and the conditions under which the data was produced.

III. PicNIC

This section introduces PicNIC, a picture-based approach to diagnosing black box industrial networks using machine learning. Fig. 1 shows the complete pipeline from training using SieMonX to inference in an industrial network.

Data Sources: Fig. 1 shows the two primary data sources for PicNIC. SieMonX mainly provides training data for the neural network, while the data from the real factory network is used for inferencing and diagnosis. It is also possible to use labeled factory data to train the model further to increase classification accuracy.

Data Preprocessing: To use a convolutional neural network for time series classification, the data is preprocessed and converted into a visual representation. These representations must be able to represent information in the time domain, not to lose information about the incoming signal. Gramian Angular Fields (GAFs), Markov Transition Fields (MTFs) [8], and Recurrence Plots (RPs) [9] represent this information accordingly and can keep temporal correlations of the incoming data.

The preprocessing step splits incoming time series data into chunks of n items, as each transformed image has a dimension of nxn pixels. Each chunk contains 100 delay measurements to keep images small to decrease the training and inference time and the model size. Small chunk sizes also decrease the delay between recording the chunk's first measurement and the image's classification. The preprocessor converts each chunk into a GAF, MTF, and RP as input for the CNN.

As the CNN needs exactly one image for classification, the last preprocessing step stitches the three separate images into one final image. The final image has n pixels in width and height with three channels, similar to an RGB picture. In contrast to an RGB image, each channel represents either the GAF, MTF, or RP, not the color value.

Training & Inferencing: Training and inference commence similarly, with the picture format as an intermediate between the data and the neural network. An InfluxDB buffers the streamed data from SieMonX or the black box network. From here, the preprocessing pipeline takes the images, transforms them, and feeds the neural network for training or diagnosis.

Use Case & Results: An emulated industrial network inside SieMonX with multiple agents delivers training data for no error and high background load scenarios between two agents to train PicNIC. The trained PicNIC model then classifies measurements between two different agents connected at different endpoints in the emulated network. PicNIC manages to classify high background load in the black box network with an accuracy of above 90%.

IV. Conclusion & Future Work

PicNIC and SieMonX show the potential of black box network diagnosis in industrial networks. We believe that our work opens exciting possibilities to diagnose black box networks for future emerging network layouts. Future work could include a reinforcement learning approach to let PicNIC auto-adapt to changing network behaviors and layouts, without the need for manual retraining.

References

[1] H. Wu, Y.-P. Zhao, and H.-J. Tan, "A novel neural network based on dynamic time warping and Kalman filter for real-time monitoring of supersonic inlet flow patterns," *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104258, 2021.

[2] A. S. Khatouni, F. Soro, and D. Giordano, "A machine learning application for latency prediction in operational 4G networks," *2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2019*, pp. 71–74, 2019, iSBN: 9783903176157.

[3] D. Alekseeva, N. Stepanov, A. Veprev, A. Sharapova, E. S. Lohan, and A. Ometov, "Comparison of Machine Learning Techniques Applied to Traffic Prediction of Real Wireless Network," *IEEE Access*, vol. 9, no. Ml, pp. 159 495–159 514, 2021, publisher: IEEE.

[4] N. Desai and S. Punnekkat, "Enhancing Fault Detection in Time Sensitive Networks using Machine Learning," *2020 International Conference on COMmunication Systems and NETworkS, COMSNETS 2020*, pp. 714–719, 2020, publisher: IEEE ISBN: 9781728131870.

[5] A. H. Ahmed, S. Hicks, M. A. Riegler, and A. Elmokashfi, "Predicting High Delays in Mobile Broadband Networks," *IEEE Access*, vol. 9, pp. 168 999–169 013, 2021, publisher: IEEE.

[6] J.-R. Jiang, J.-B. Kao, and Y.-L. Li, "Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning," *Applied Sciences*, vol. 11, no. 15, p. 6698, Jan. 2021, number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

[7] T. Kubota and W. Yamamoto, "Anomaly Detection from Online Monitoring of System Operations Using Recurrent Neural Network," *Procedia Manufacturing*, vol. 30, pp. 83–89, Jan. 2019.

[8] Z. Wang and T. Oates, "Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks," Jan. 2015.

[9] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence Plots of Dynamical Systems," *EPL*, vol. 4, no. 9, p. 973, Nov. 1987.