

Learning Long- and Short-Term Temporal Patterns for ML-Driven Fault Management in Optical Communication Networks

Moisés Felipe Silva[✉], Alessandro Pacini[✉], Andrea Sgambelluri,
and Luca Valcarenghi[✉], Senior Member, IEEE

Abstract—The deployment of 5G and network slicing has challenged the current network management requirements, triggering the need for programmable and software-driven architectures. Thus, automated real-time fault management for self-managed networks with machine learning and artificial intelligence at the forefront has become necessary. This is especially the case of optical communication systems, accountable for most of the data traffic worldwide. This study introduces the application of a novel failure detection and localization framework capable of forecasting failures in optical systems based on an unsupervised learning strategy. In this approach, the Long- and Short-term Time-series Network (LSTNet) is exploited for modeling the normal behavior of optical systems. Then, failure conditions are properly forecast without explicitly training the model for such cases, easing the data acquisition process. Later, forecast values and actual measurements from optical equipment are used to derive an outlier detection method to detect and locate failures to improve the decision-making process at the network orchestrator. Laboratory experiments comparing the proposed approach with the Recurrent and Long Short-Term Memory models in terms of failure detection and forecasting performance show that using the LSTNet reduces the mean squared errors in 95% for unseen data, indicating robustness and suitability for real-world environments.

Index Terms—Digital orchestration, ETSI zero-touch networks, deep learning, novelty detection, optical communication systems.

I. INTRODUCTION

AT THE core of the fifth generation of mobile communication networks (5G) is the support of a wide range of applications running over slices of the same physical infrastructure. This scenario brings, to the forefront of research and development, issues related to high traffic volume, numerous connected devices, better quality of user experience, near-to-zero latency, and coverage [1]–[3].

These requirements have a profound impact on how operators manage their networks, driving increasing efforts

Manuscript received 30 June 2021; revised 30 November 2021 and 19 January 2022; accepted 24 January 2022. Date of publication 28 January 2022; date of current version 12 October 2022. This work received funding from the ECSEL JU project BRAINE (grant agreement No 876967). The JU receives support from the EU Horizon 2020 research and innovation programme and the Italian Ministry of University, and Research (MUR). The associate editor coordinating the review of this article and approving it for publication was R. Pasquini. (*Corresponding author: Moisés Felipe Silva*)

The authors are with the TeCIP Institute, Scuola Superiore Sant'Anna, 56100 Pisa, Italy (e-mail: m.mellodasilva@santannapisa.it).

Digital Object Identifier 10.1109/TNSM.2022.3146869

into highly modular, flexible, and programmable architectures. With the advent of network function virtualization (NFV) [4]–[6] and software-defined networking (SDN) [7]–[9], full end-to-end automation of network and service management has become an urgent need for delivering the ever-increasing system requirements, at the same pace as guaranteeing economic sustainability [10], [11]. Thus, the ultimate goal is to enable large and fully automated networks capable of performing policy-driven self-diagnostics in accordance with the specific set of applications running over the networks [12], [13]. In this scenario, intelligent system diagnosis are an essential part of network automation, with machine learning (ML) and artificial intelligence (AI) algorithms carrying out the core activities. As networks grow and become increasingly complex, AI- and ML-based services can help providers to deal with the increasing number of parameters and adapt to changing environments. This requires the system operations to be data-driven, or more precisely, ML-driven, which means that systems and services need to be built along AI/ML solutions incorporated from the start [2], [14], [15].

Following this need, in the past decades, most studies leveraged the advantages of using supervised strategies to a variety of goals, ranging from management to forecasting and event detection. Among the myriad of approaches, the ones based on artificial neural networks (ANNs) are undoubtedly the most prominent, for example in 5G systems where the tasks related to their digital orchestration involve pattern recognition, for which ANNs have proven to be extremely powerful [16]–[19].

However, in these cases, data from all possible considered scenarios, e.g., failures, are needed for model training, which in practice can slow down the development process, increase budgetary demands and, if all the failure mechanisms are not taken into account during the training, incur several misclassifications. As ML-based algorithms learn from past experience, data from varied system conditions, including those of failure, must be available during the training phase, which may not be desirable for applications where data from varied failure types are not easily obtained. Although ML model based on *prior-statistics* can be an option to artificially create failure data, their implementation can be critical as for complex networks intensive and expensive efforts are needed.

On the other hand, the idea of novelty detection by means of outlier analysis for self-managed systems is not totally new, with implementations for intrusion detection and system

anomalies in mobile networks [20], [21]. The objective is to monitor a sequence of patterns and trigger an alarm if this one significantly differs from the herd. Therefore, the application to the detection of failures or other types of events in the context of network management is manifest. Outlier detection methods have been utilized for many different applications and cover a broad range of fields of research such as econometrics, computer science, medical and biological sciences, meteorology, or even political science. Some recent advances regarding robust outlier detection can be found in [22], [23]. However, applications involving novel detection method for optical communication networks are few, motivating further research into developing strategies for actual use in practical scenarios. In fact, the importance of the specific monitoring of optical systems become evident during the most recent pandemic of Covid-19, as the traffic demands for optical communications significantly increased during lockdown measures imposed worldwide [24].

In this work, we introduce a novel failure detection approach to forecast, detect, and localize failures in optical communication systems on the basis of an unsupervised framework. On this approach, data from failure conditions are not used during the training of the ML model, disregarding the acquisition of data from equipment that are undergoing failures or to simulate failure conditions from statistical models. Instead, only data from normal and regular network conditions are required for the training phase. As opposed to the conventional supervised techniques, the one here demonstrated simplifies the data collection for training purposes (as data from regular network conditions are easier to obtain) and accelerates the deployment of the ML model in production.

The approach is inspired by the Long- and Short-term Time-series Network (LSTNet) framework [25], which has proven to be especially effective for learning very long temporal dependencies while maintaining high accuracy for predicting short-term patterns. Although its core capability is to handle large sequences of input data, for instance, it is extremely powerful to perform time-series modeling [26], [27]. Most often, Long Short-Term Memory (LSTM)- and Recurrent Neural Network (RNN)-based models are the natural choices for dealing with sequence-dependent data. However, their integration with recurrent-skip units allows the dynamic mapping of extremely long parts of sequential data before any output [28]. Unlike RNNs, for instance, the architecture here introduced derives context-specific predictions as it is capable of recalling long past information. It also disregards the need for the input data to be timely ordered beforehand, simplifying model re-training after deployment, and can be easily scaled to operate with hundreds to thousands of time-series, a crucial feature as the communication networks continue to grow. Moreover, from the studies introduced in [25], the LSTNet is highly scalable in terms of maintaining its forecasting performance as the number of the input time-series and number of time-steps increase. This characteristic, along with its forecasting performance, is desirable when working on data from optical communication networks as the number of equipments and, consequently, telemetry parameters increase dramatically as the networks increase.

We compare the performance of a slightly modified version of the LSTNet architecture to forecast optical parameters from optical equipment. Later, anomalous events are detected and classified as failures following automatic threshold definition. Additionally to the forecasting and detection, the proposed approach can also be used to locate a failure at the equipment level by separately defining thresholds for each optical parameter under monitoring, allowing to locate the problems that occurs both in the optical equipment or in the optical link. The technique is compared in terms of false-positive (Type I errors) and false-negative (Type II errors) indications of failures and forecasting in relation to the classical RNN and LSTM models on performing the same tasks for optical data generated by laboratory experiments.

The rest of the paper is structured as follows. Section II is devoted to introduce closely-related works proposing novelty detection strategies for 5G/IoT applications. In Section III we present the main operating principles of the LSTNet model, along with the specific changes we performed during our tests, and how it can be implemented to perform time-series forecasting and failure detection in the context of optical communication networks. Also in Section III, our deep learning-based novelty detection approach to identify failures using the trained forecasting models is derived. Section IV carries out the description of the optical setup used for data acquisition as well as the optical dataset used in our experiments. Section V introduces the performance results and carries out comparison discussions. Finally, in Section VI we present our conclusions and directions for future works.

II. RELATED WORKS

Bhuyan *et al.* [29] introduced a comprehensive review on the use of anomaly-based intrusion detection methods for a variety of networks, including mobile applications. Several different types of algorithms are reviewed, including data- and model-driven approaches. Following a similar approach, Kwon *et al.* [30] and Lam and Abbas [17] presented machine learning methods for anomalous detection with special focus on cybersecurity for 5G networks. Amid other works, these ones aim at introducing the use of deep learning techniques for autonomous detection of intentional attacks to 5G networks, with particular attention to the proposal of a software-defined security architecture by Lam and Abbas [17], that has got the potentials to provide automated, flexible and scalable networks.

Although previous works have been performed on novelty detection methods for intrusion detection, there are also other works proposing the actual use of such methodology to failure and event detection in optical networks. Mayer *et al.* [31] proposed a soft failure localization technique based on a supervised neural network that requires only a few training scenarios to achieve adequate performance. Applied over telemetry data from SDN streams of network parameters, the proposed approach requires the implementation of a numerical model of the physical network to generate simulated data from failure scenarios for training. Similarly, Shimizu *et al.* [18] introduces a neural model trained to detect the presence (or

not) of a failure in an optical link based on the OSNR values from multiple lightpaths, where the output of the model provides a true/false response. Although efficient, this technique is limited to only indicate the presence of a failure in the system. Also, this supervised approach requires data from failure and no failure conditions generated via a numerical model from the targeted network.

In the same sense, several machine learning methods are compared in [32] to perform detection and identification of equipment failures in optical networks, achieving more than 98% in accuracy for identifying the equipment failures using BER traces. However, this is only achieved after using data from both failure and no failure conditions in a supervised learning fashion. Santos *et al.* [33] developed a technique based on fog computing for 5G low power wide area networks, with the aim to apply it over IoT, smart cities, and industry 4.0 scenarios. The authors adopt a cluster-based strategy for anomaly detection and compare to a classical outlier detection method, and conclude that both techniques can be suitably applied to IoT scenarios. However, the main drawback of this study is the clear limitations of both algorithms in terms of system scalability. If the number of devices and, consequently, the complexity of the system network grows, the use of these models can be unsuitable, even in the case of using a fog computing architecture. In the same sense, as standardized by the third generation partnership project (3GPP) [34], works exploring the use of self-organizing networks (SONs) for the processes of configuration, optimization, and healing have been developed.

The authors of [35] introduced an approach based on self-organizing maps (SOM) and k-medoids to detect events using key performance indicators (KPI) from communication cells of mobile networks as input. This technique is capable of detecting anomalous behavior at the cell level but is not properly useful to detect anomalies at the KPI level. In [36], the use of KPIs is also exploited in a supervised framework, with notable improvements in performance compared to other methods; however, this supervised approach requires a very large number of training data from several different types of network conditions and is only able to perform failure detection.

Wang and Handurukande [37] and Wu *et al.* [38] proposed the use of the regression analysis to forecast network parameters and detect system failures based on the similarity of the forecast values and the next real sample. The drawbacks are mainly the need for directly correlated features and samples, although in [37] high precision for chance outliers is reached. Similarly, Song and Liu [39] introduced an outlier detection methodology based on k-nearest neighbors, a technique where only occasional outliers are detected.

One of the issues with previous works proposing novelty detection methods is scalability, where those can behave well for small applications but raise several concerns when applied to larger systems with several devices and parameters to be monitored. These techniques often behave well for several networks with very predictable behavior, however these do not scale their performance in accordance with the network evolution. Additionally, based upon the author knowledge, works have not been found aiming at proposing techniques

capable of performing forecasting, anomaly detection and localization in accordance with the specific monitoring goals. Most works focus on a specific goal, although most envision to include other features in further extensions. Also, it has been found limited literature regarding the application of novelty detection-based approaches for optical communication systems capable of performing all the tasks here proposed.

III. NOVELTY DETECTION METHOD FOR SYSTEM FORECASTING, FAILURE DETECTION AND LOCALIZATION

In this section, the basic steps to accomplish the full LSTNet model are described (Section III-A). Later, the approach to leverage the trained model to perform failure detection, localization and classification are introduced (Section III-B) for completeness.

A. Multivariate Time-Series Modeling With Deep Learning

Time-series modeling and forecasting can be mainly treated as a standard regression problem with time-varying parameters, for which several different strategies are applied to. Among these strategies, the most promising ones are based on the use of deep neural models. Inspired by the framework proposed in [25], in this section, we formulate the task of multivariate time-series forecasting with the LSTNet model.

A time-series matrix is defined $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n]^T \in \mathbb{R}^{n \times m}$ where $\mathbf{Y}_i = [\mathbf{y}_i^1, \mathbf{y}_i^2, \dots, \mathbf{y}_i^m] \in \mathbb{R}^m$ is a vector containing m features observed at time i and n is the number of times where the features are observed. For a given time window, h , constrained not to be larger than n , the goal is to use the past h observations $\mathbf{Y}_{n-h}, \mathbf{Y}_{n-h+1}, \dots, \mathbf{Y}_{n-1}$ to predict the observation at \mathbf{Y}_n . Hence, we can define our input matrix at time n as $\mathbf{X}_n = [\mathbf{Y}_{n-h}, \mathbf{Y}_{n-h+1}, \dots, \mathbf{Y}_{n-1}]^T \in \mathbb{R}^{h \times m}$. Note that the time window h is often defined experimentally or with prior information from the problem and the data set being considered.

For this particular problem formulation, the LSTNet has been proven more effective than vanilla RNNs and LSTMs for similar problems, and the main reason for its success is the coupling of the attention mechanism. After has been demonstrated that the performance of encoder-decoder networks degrade as the size of the input data increases, the attention mechanism was proposed to alleviate this issue. Its role is to keep embeddings of the input data from very long (and very short) data sequences, allowing to the neural model to take into account the context when performing a prediction. In particular, for the case of optical communications, it is desirable that the model remembers possibly long temporal patterns (from many past time-steps). For the cases of equipment replacement (for degraded performance) or network changing condition (for normal oscillations), it is useful that the neural model remembers possibly long temporal patterns in order to accommodate changes in the systems. Figure 1 presents the general overview of the LSTNet model developed in [25] and introduced in our work. Next, we detail the mechanisms composing the LSTNet architecture

1) *Convolutional Layer*: The first layer is composed of k convolutional masks $\omega_k \in \mathbb{R}^{m \times w}$, where w denotes the

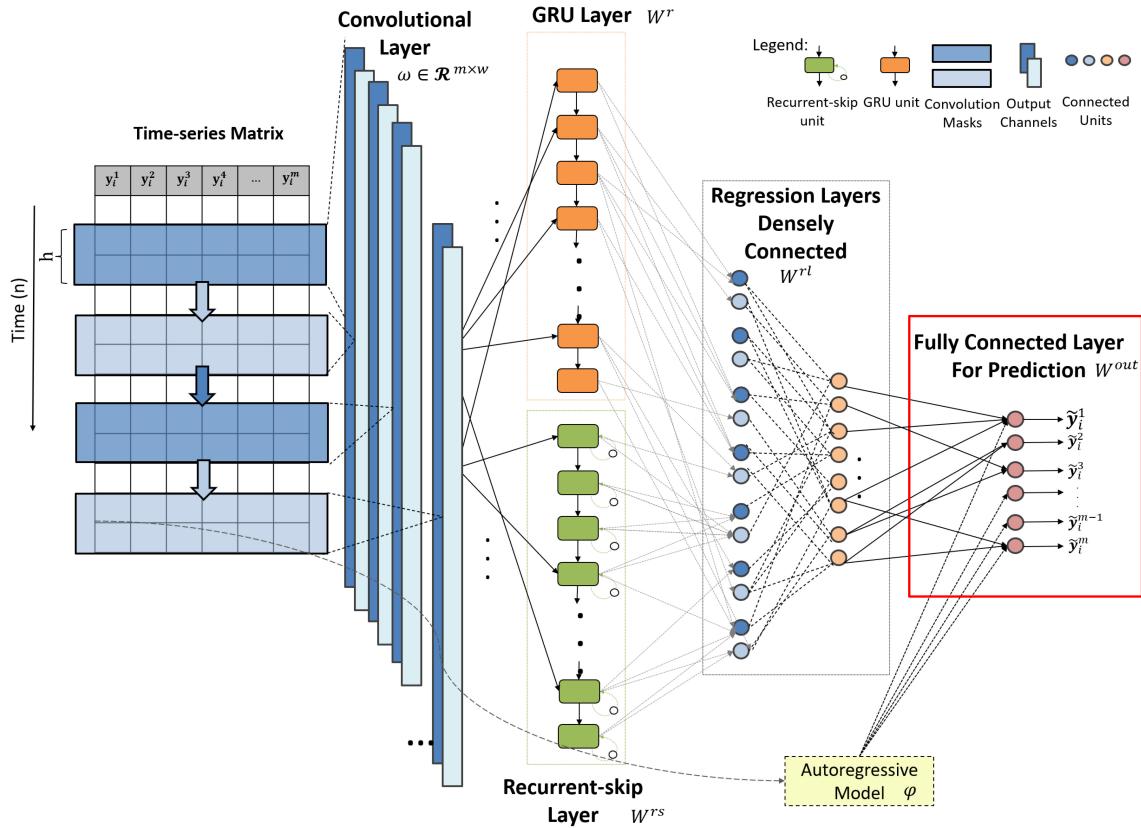


Fig. 1. Overview of the LSTNet architecture for multivariate time-series forecasting, here, applied to the case of forecasting, failure detection and localization in optical communication systems. We have included an additional fully connected layer to integrate the output of the neural model with the autoregressive model. The box in red color indicates the change.

mask width, whose goal is mainly to perform data reduction. Recalling the fact that convolutional models are mainly intended for video/image processing, one should note that in the case of matrix with several time-series, this operation is highly effective in reducing the number of input values and variables and allows the model to deal with data sets with extremely large number of series, which is helpful for real-world optical communication networks as the number of equipments and optical links continues to grow [40].

The second goal of this convolutional layer is to exploit local temporal correlations and to enforce local sparsity, a crucial feature when attempting to extract latent representations from the input data. Note that, instead of coupling a convolutional layer with pooling operations, the LSTNet architecture operates without pooling to directly extract short-term patterns in the time dimension. Moreover, the function $\text{ReLU}(x) = \max(0, x)$ is chosen for activation, and it is used after each convolution operation for every convolutional mask.

2) *Recurrent and Recurrent-Skip Components:* In this case, the output of the convolutional layer serves as input to both recurrent and recurrent-skip layers without further treatment. For the recurrent layer, gated recurrent units (GRU) are employed, along with the ReLU activation function, which is not common when working with GRU units. More often, hyperbolic tangent is used for the hidden state activations; however, using the ReLU function leads the backpropagation of gradient and leads to more reliable performance, mainly

for large data sets with several variables. The output of this module is the hidden state for each timestep.

On the other hand, to keep in memory very long seasonal information, the mechanism introduced in [28] is used. The recurrent-skip module adds skip-links between the adjacent hidden state units to pass over to the next layer only the hidden states that may benefit the prediction for a given timestep. Therefore, each hidden unit has knowledge on the hidden states of the previous units, and the selection of which states are passed over to the following units is decided internally through the hidden gates of each unit. This sequence shortening technique enforces the model to keep into its local memory very long hidden states in order to benefit from its local information later. For the units in this layer, a binary state update gate, $u_t \in \{0, 1\}$, indicating whether or not the model state should be updated or copied at t -th training timestep. Considering $\tilde{u}_t \in [0, 1]$ as the probability of updating the state, the process is characterized as follows:

$$u_t = \text{binarize}(\tilde{u}_t) \quad (1)$$

$$s_t = u_t \cdot S(s_{t-1}, v_t) + (1 - u_t) \cdot s_{t-1} \quad (2)$$

$$\Delta \tilde{u}_t = \sigma(W s_t + b) \quad (3)$$

$$\begin{aligned} \tilde{u}_{t+1} &= u_t \cdot \Delta \tilde{u}_t + (1 - u_t) \\ &\cdot (\tilde{u}_t + \min(\Delta \tilde{u}_t, 1 - \tilde{u}_t)) \end{aligned} \quad (4)$$

where σ is the usual sigmoid function, $\text{binarize}(\cdot)$ is a step function turning the input values into binaries, v_t is the input

data of the layer at timestep t , W and b are the model weights and bias. s_t is the hidden state sequence at the training timestep t while $S(\cdot)$ is an update function for a given hidden state and input value. The update gate u_t opens and closes in accordance to a cumulative update potential \tilde{u}_t , incremented by $\Delta\tilde{u}_t$. Whenever this update gate reaches its full potential, an update on the weights is triggered, and the accumulated value in \tilde{u}_t is flushed and $\tilde{u}_{t+1} = \Delta\tilde{u}_t$. On the other hand, when an update is skipped, the update potential for the following time step, \tilde{u}_{t+1} , is incremented by $\Delta\tilde{u}_t$.

Note that both the recurrent and recurrent-skip modules run in parallel over the outputs from the convolutional layer. Then, the outputs from both recurrent modules are combined through regression layers using densely connected units. The inputs to this regression module are the p hidden states h_t^r from the recurrent module and the hidden states h_t^{rs} from the recurrent-skip module. The output of this module is thereby given as

$$h_t^D = \text{ReLU}\left(W^{rl}\left[W^r h_t^r + \sum_{j=0}^{p-1} W^{rs} h_{t-j}^{rs} + b\right]\right) \quad (5)$$

where p is the hyperparameter defining how many hidden states should be used, h_t^D is the regression module output. W^r and W^{rs} are the weights from the recurrent and recurrent-skip modules, W^{rl} are the weights for the regression layers, while h_t^r and h_t^{rs} are their respective hidden states.

3) Autoregressive Model: To alleviate the issue of changing input scaling, a linear autoregressive (AR) component is created in parallel. The linear AR model focuses on the local scaling issue, while the deep network performs nonlinear predictions based on the recurring patterns, thus focusing on global predictions. In summary, the AR model is formulated as

$$h_t^{AR} = \epsilon + \sum_{j=0}^{q-1} \varphi_j Y_{i-j} \quad (6)$$

where ϵ is a constant, q is the model order, φ the model coefficients, and h_t^{AR} is the output of the AR component. Formulated with the classical AR model, the original LSTNet linearly combines the outputs of the neural network with the AR component to obtain the final predictions. However, in our experiments, we found improved and consistent results after coupling the whole framework with an extra neural network layer. The goal of this addition is to combine the outputs of the neural model with the AR component in a more direct fashion without using integration to create the global output. This aspect can be described as

$$\tilde{h}_t^{out} = h_t^D + h_t^{AR} \quad (7)$$

$$\tilde{Y}_t = \text{ReLU}\left(W^{out}\tilde{h}_t^{out} + b^{out}\right) \quad (8)$$

where \tilde{Y}_t is the global output, whose dimension is equal to the number of input time-series, m . W^{out} and b^{out} are the weights and bias, respectively, while \tilde{Y}_t accounts for the summation of the outputs from the regression layers and the AR model.

4) Model Optimization: Amid several possible objective functions, the LSTNet conveys the use of a simplified version

of the Linear Support Vector Regression (SVR), described as

$$\underset{\Theta}{\text{minimize}} \sum_t^{t_{\max}} \sum_{i=0}^{n-1} \left| Y_{t,i} - \tilde{Y}_{t-h,i} \right| \quad (9)$$

where Θ is the set of parameters to optimize during training, \tilde{Y}_i the predicted values and t_{\max} the maximum number of training iterations (epochs). This function, empirically chosen, has the advantage of being robust to possible anomalies into the training data, therefore indicated for real data sets (as attested in our experiments). With the objective function defined, the training is then carried out with the Adam optimization algorithm, widely known for stochastic gradient descent-based training of deep learning models.

B. Novelty Detection Strategy for Failure Detection, Localization and Classification

The complete methodology applied in this study is depicted in Fig. 2. First, a supervised training of the machine learning model is carried out over a multivariate time-series data set composed of equipment parameters from an optical communication system. At this phase, only raw telemetry data from regular working conditions of the system are used. The goal is to derive a model that is only able to forecast values for normal system conditions. The forecasting can be multi-step ahead or single-step, if the goal is to simply identify and classify events from the optical network for real-time decision making.

Next, after the model learning step, fault indicators (FIs) for the training data are calculated on the basis of the Euclidean distance between the forecasted values, \tilde{Y}_i , and the actual measurements, Y_i . Assuming the test data to be obtained under the same operational conditions as the training data, the FIs should be nearly invariant for feature vectors extracted from the normal conditions of the optical system. On the other hand, if the system is passing through any abnormal behavior (or an actual failure) this error should grow in accordance to the level of discrepancy to the normal system conditions.

The FIs from the training data are later used to estimate linear thresholds for each of the monitored parameters. Note that these thresholds are simply cut-off values defined over a certain percentage of confidence from the fault indicators of the training data (composed of data from normal conditions), as is often performed in classical outlier detection. Also is important to highlight that the computation of these individual thresholds allow to perform the failure detection and localization. The idea is to monitor, individually, each telemetry parameter from the network devices, and to generate alarms whenever an abnormal behavior occurs in the individual equipments.

Next, in failure detection and localization step, the machine learning algorithm is fed with new telemetry data. Then, based on the errors between the actual measurements and the forecasted values, each input feature vector is transformed into a fault indicator for the new telemetry data. At last, the FIs are compared to the linear threshold previously calculated to perform failure detection and localization. As previously described, the general intuition is that if new readings are from normal system conditions, the error between the forecasting values and the actual values approaches zero, and thus the

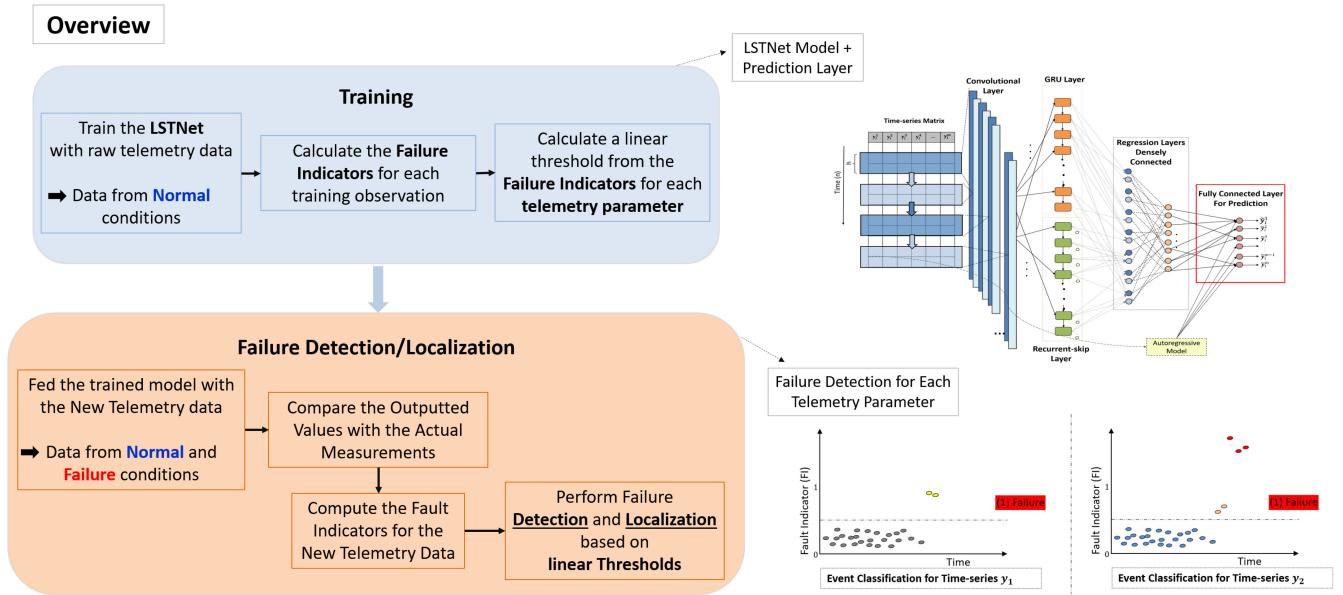


Fig. 2. Overview of the time-series modelling and failure detection strategy developed on the basis of a novelty detection framework.

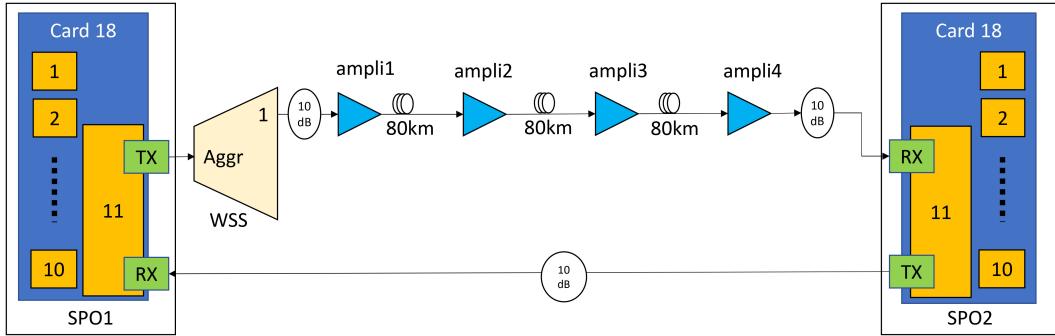


Fig. 3. Considered testbed.

corresponding FI values should be smaller than the pre-defined threshold. However, for data arisen from failure conditions the error should increase accordingly to the level of the problem, and therefore, the corresponding FI values should increase to above the pre-defined threshold, which corresponds to the detection of a failure.

IV. EXPERIMENTAL SETUP AND DATA ACQUISITION

A. Experimental Details

In order to collect the required data, an experimental testbed has been considered. This testbed is part of the ARNO testbed at the Tecip Institute of Scuola Superiore Sat'Anna in Pisa, Italy. The considered testbed, is shown in Fig. 3 and presents optical network devices dedicated to research activities for studying novel control, management and monitoring schemes.

It includes two Ericsson SPO 1400 devices (i.e., SPO1 and SPO2) which are commercial transport nodes with Optical Transport Network (OTN) functionalities. Each SPO device is equipped with a 100Gb/s Optical Transport Network (OTN) muxponder (installed at slot 18) presenting one Dense Wavelength Division Multiplexing (DWDM) optical line (port 11) and ten 10Gb/s Ethernet tributaries. The muxponder

performs coherent detection, collecting the main coherent metrics (i.e., pre-FEC Bit Error Rate (BER), Optical Signal to Noise Ratio (OSNR)). The testbed includes one Wavelength Selective Switch (WSS) that receives as input the line port of the 100G muxponder of SPO1 and as output the multi-span link. At the exit of the WSS a 10dB attenuator is installed.

The link going from SPO1 to SPO2 consists of 3 spans. A single-mode-fiber spool with length of 80km is included in each span, resulting to 240km total link length. 4 EDFA amplifiers are used to compensate for the power attenuation experienced along each traversed span. All the EDFA amplifiers are controlled via SPO devices (ampli1 and ampli2 are controlled by SPO1, while ampli3 and ampli4 are controlled by SPO2) and present a configurable gain in the range 15-25dB, with output mute power of 0.4dBm. All the amplifiers are configured in constant gain mode, with a gain value that allows to enter each span with 0dBm of optical power. The reverse link (i.e., from SPO2 to SPO1) is in back-to-back configuration, presenting only an attenuator of 10dB.

To collect the data, the monitoring system proposed in [41] has been used. By selecting the appropriate metric at each traversed element (i.e., the input/output optical power levels at each amplifier and the OSNR and the BER at the

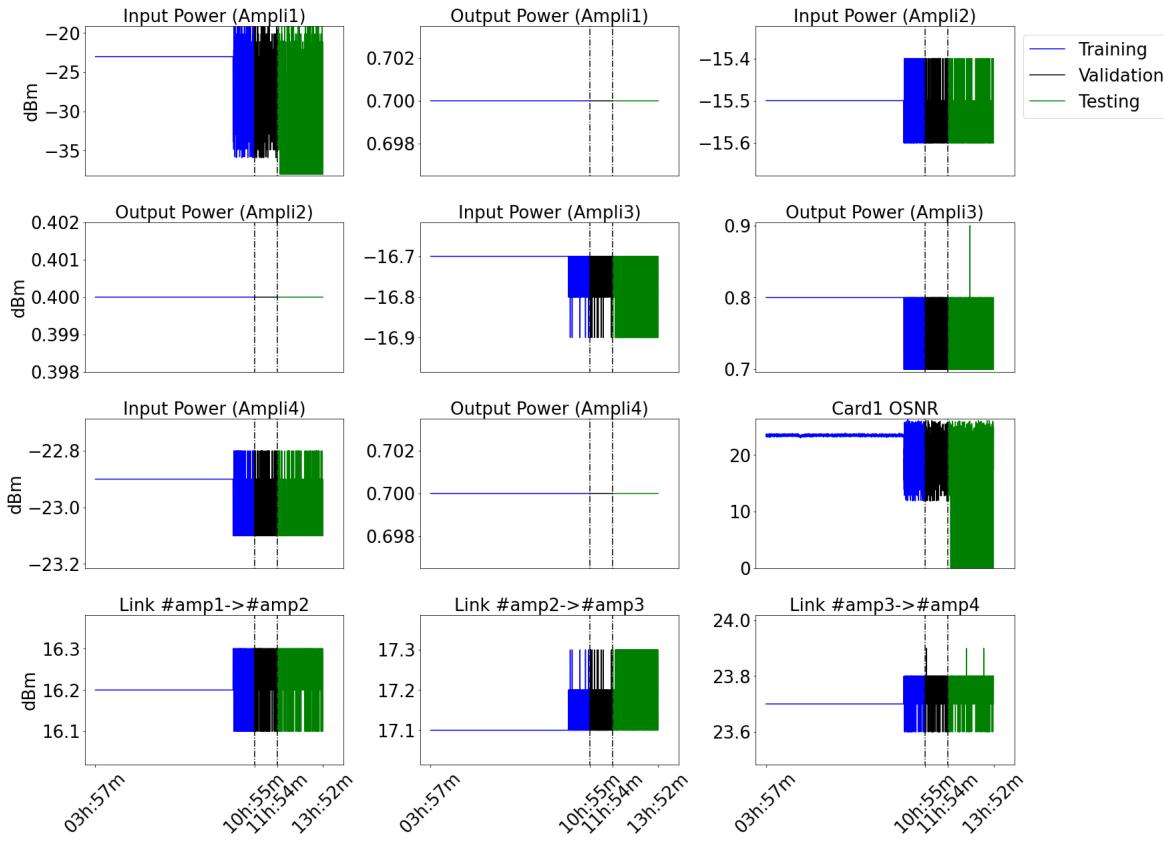


Fig. 4. Optical dataset along with the training, validation and test data split.

transponders), we relied on the Kafka-based telemetry system to distribute the selected metrics while a new collector (i.e., Kafka consumer, receiving the telemetry) has been designed in order to export the data in a csv format. The dataset has been uploaded into a public Github repository [42].

B. Evaluation Metrics

The dataset is composed of 10,945 samples, collected with sampling frequency of around 3.5 seconds. This value is limited by the minimum time required by the considered commercial equipment to provide transmission coherent data (i.e., BER). The total length of the considered dataset is around 10 hours. Among the samples, the first 70% of the data are used for training, the following 10% for validation and the last 20% for testing. It is important to note that only in the percentage of the data corresponding to the test phase faulty conditions are simulated. More specifically, after every 4 minutes of data collection 1 minute of random failures are simulated into the optical system which correspond to 857 samples related to failures. The WSS, shown in the system at the beginning of the first span of the optical link, has been used to introduce variable attenuation to the optical power. In particular, we implemented a WSS routine, with steps of 1 minute, to configure different attenuation values. In normal condition, the attenuation varies in the range [0, 18]dB, while, in case of failure, after four steps the attenuation is set to 25dB, reproducing hard failures. The considered scenario is based on the injection of hard failures into the system (by setting 25dB

of attenuation, the received signal goes below the RX sensitivity). In our opinion, the generation of the link failures has been designed in order to make the detection process more complicated, since the hard failures are created in the middle of a dynamic link condition (without a fix power level during the experiment duration). However, in the training and validation data, only samples from the normal system conditions are available, allowing the novelty detection approaches to classify deviations from the normal conditions as possible outliers. This data partitioning can be clearly seen in Fig. 4, from which it is possible to note that only the data from the input power of the amplifier 1 and the OSNR of the Card 1 have clear and noticeable drops in the magnitude, as expected due to the experimental setup. Therefore, detected outliers should be expected only for the data arisen from these two equipment.

Moreover, to evaluate how the signal degrades in a given optical link, we created additional variables that are the difference between the output power of a given amplifier and the input power of the next amplifier. For this specific experimental setup, this strategy for monitoring specific links generates 3 extra features for model training.

The outlier detection performance of the compared techniques is evaluated in terms of Type I (false-positive) and Type II (false-negative) indications of problems based on a linear threshold defined for 99% of confidence over the training data. For comparing the forecasting results we used three conventional metrics:

- 1) *Mean Squared Error (MSE)*: Popular measure for discretizing the error rate of a regression model. Allows to

compare only models whose errors are measured in the same units, although the resulting values vary without a specific range.

- 2) *Pearson's Correlation (Corr)*: Common test statistics for measuring the relationship between variables, known for its good performance in evaluating the association of two variables because of its covariance-based formulation. Ranging between $[-1, 1]$, highly correlated variables are those approaching -1 and 1 , on the other hand uncorrelated variables approach 0 values with this statistic.
- 3) *Coefficient of Determination (R^2)*: Also known as the R^2 index, this coefficient summarizes the exploratory capability of the regression model and describes the proportion of variance of the dependent variable explained by the regression model. The intuition is that for “perfect regressors” the corresponding R^2 index is 1 , indicating that the model successfully explains the entire variance of the data, otherwise the output is 0 .

C. Methods for Comparison

From the experimental setup we conducted intensive experiments with 3 neural-based methods for time-series forecasting and novelty detection: recurrent neural network (RNN), long short-term memory (LSTM) and the long- and short-term time-series network (LSTNet) described previously. For statistical purposes, a total of 20 different trials with randomly initialized weights were performed for each algorithm, from which average and standard deviation for each comparison metric were estimated.

In terms of model architectures, for the RNN and LSTM models 6 hidden layers were created in a sequential manner, each one with the dropout mechanism activated (0.2 was the defined drop value). The first 4 hidden layers were set with 100, 80, 60 and 40 units each, along with and additional 2 densely connected layers for outputting the model predictions (each dense layer has $3 * m$ and $2 * m$ units, respectively). The learning rate, batch size and number of epochs were defined as 0.0001, 128, and 100 for both algorithms, as well as the mean squared error was adopted as the loss function. Z-score normalization was employed for both techniques. Finally, the time window, that is the number of past observations used for predicting future values, in this work, is set to 5.

In a similar manner, for the LSTNet model, the number of convolutional filters and kernel sizes are 10 and 3, respectively. On the other hand, the number of GRU units and recurrent-skip units are 3 and 2, respectively. The autoregressive model order is set as 2 to keep this module simple. Finally, the learning rate and batch size are 0.001 and 32, respectively. For the data normalization scheme it was found that using the min-max approach improved the forecasting results, reason why this is adopted in the following experiments.

Is important to highlight that, for all the methods, the architectural projects were defined after several tests varying the number of layers and hidden units with the training parameters for the specific optical dataset created, which may vary

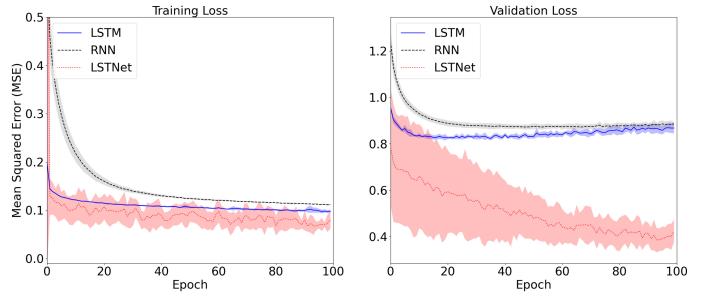


Fig. 5. Average learning curves along with the standard deviation for each technique.

for other datasets. For sake of brevity we do not report them in this paper.

V. RESULTS AND DISCUSSIONS

Table I introduces the numerical results in terms of average and standard deviation after 20 trials of all the techniques over the optical data set using the comparison metrics. Note that the values are separated in accordance to the performance of each technique for the different subsets of the optical data (training, validation and test sets). The best result for each pair (technique, metric), in terms of the average performance, is highlighted in bold face. The table shows that the LSTNet-based approach presents the best results among the compared techniques over all the comparison metrics. It is clearly shown that the LSTNet consistently presents the best performance over the widely used RNN and LSTM for this specific dataset, specially for the testing dataset, indicating enhanced generalization performance and capabilities to forecast even cases of abnormal behaviors which can be used to instantaneously predict system failures.

Initially, the first intuition should be that, as the LSTNet was only trained on data from normal conditions, the model should not perform properly over the test dataset that contains data from abnormal conditions. However, it is important to recall that the main goal of the LSTNet is to perform reasonably good to predict both long- and short-term patterns. As the model uses a subset of previously collected values (short-term information) to forecast new values, the LSTNet can perform reasonably over data not directly correlated to the normal conditions. This is mainly assigned to the use of the autoregressive module, whose role in the forecasting is to account for short-term patterns.

To evaluate the learning process of all techniques, the learning curves are shown in Fig. 5. The average learning curves and their respective standard deviation are shown after for the 20 trials of each technique. From the figure at the left is seen that the LSTNet reaches the best performance in terms of minimizing the loss function, and does so in a faster pace. On the other hand, the figure showing the validation loss indicates a lack of generalization performance for the RNN and LSTM, as the loss curves indicates the models are underfitting the validation data. When considering the LSTNet only, the loss function does converge to significantly smaller values than the compared techniques, and considerably reduces its margin for

TABLE I
SUMMARY OF THE FORECASTING RESULTS OF ALL THE COMPARED METHODS. THE BEST RESULTS ARE INDICATED IN BOLD FACE

| | | LSTM | RNN | LSTNet |
|-----------------|------------|-----------------------------|-----------------------------|---|
| MSE | Training | 1,246,046.778 \pm 641.545 | 1,245,690.595 \pm 775.323 | 24,673.245 \pm 42,218.523 |
| | Validation | 173,196.906 \pm 301.459 | 172,855.887 \pm 123.744 | 8,598.267 \pm 5,549.601 |
| | Testing | 343,411.900 \pm 924.589 | 343,586.638 \pm 406.123 | 26,459.646 \pm 12,467.152 |
| Pearson's Corr. | Training | -0.313 \pm 0.0005 | -0.313 \pm 0.0006 | 0.999 \pm 0.0004 |
| | Validation | 0.341 \pm 0.0007 | 0.340 \pm 0.0005 | 0.993 \pm 0.003 |
| | Testing | 0.371 \pm 0.002 | 0.362 \pm 0.001 | 0.982 \pm 0.008 |
| R^2 | Training | 0.0016 \pm 0.001 | 0.0014 \pm 0.0009 | 0.77 \pm 0.41 |
| | Validation | 0.022 \pm 0.0042 | 0.017 \pm 0.0018 | 0.98 \pm 0.0068 |
| | Testing | 0.0191 \pm 0.0063 | 0.0154 \pm 0.0028 | 0.9632 \pm 0.0172 |

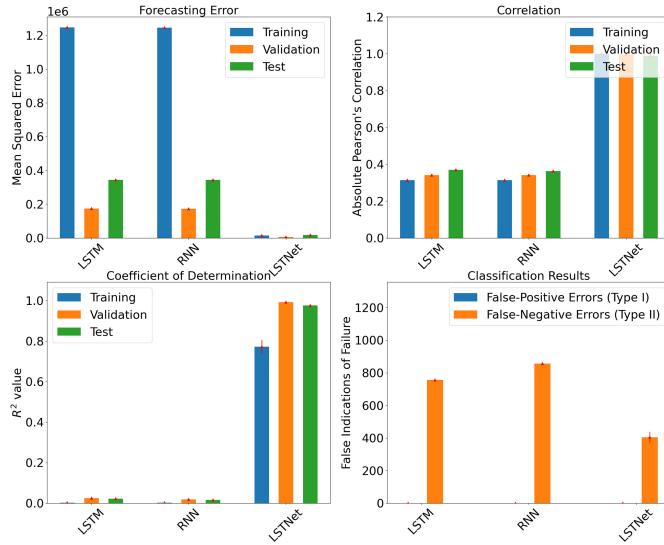


Fig. 6. Summary of the results in terms of each comparison metric after 20 different trials with random initialization of the weights for each technique. The results are shown in terms of average and standard deviation (indicated as red lines on top of each bar) on the training, validation and test sets.

variations as the training epochs proceeds. This is also verified in Tab. I for the comparison metrics on the test data. From the figures in the second row, we can confirm the problems in terms of stability verified before in Tab. I. Although the average performance of the LSTNet is better than the compared methods, its standard deviation indicates lack of stability regarding the first 5 epochs of the training. As the training moves forward, the LSTNet converges to better results when compared to the other techniques.

The main issue with the LSTNet results is the stability along several trials of the model, as one can verify by standard deviation of each metric. Although the other techniques are stable their average performance is poor, which makes the LSTNet the best choice among the compared methods, regarded its lack of stability. To aid a more intuitive comparison, the performance of each technique in terms of the comparison metrics are shown in Fig. 6. The Figure clearly shows that the LSTNet significantly outweighs the other methods in all of the metrics, mainly when considering the model error in terms of the MSE values. Compared to the LSTNet model, for the training data, the RNN and LSTM models performed quite poorly, behavior that propagates for the validation and test sets. In the latter set, the LSTNet outperformed

the RNN and the LSTM by approximately **95%**. For the Pearson's correlation, the forecasting results of the LSTNet approaches the perfect score of 1, while the other techniques have poor performance in attempting to forecast values with actual correlation to the baseline data. Finally, when considering the R^2 value, the LSTNet model demonstrated the best average performance, retaining approximately 77% of the variance for the training, 98% for validation, and 96% for the test sets, which is expected after considering the MSE values.

Additionally, to evaluate the performance of the models to perform outlier detection, Fig. 6 shows the average number of Type I and Type II errors for the test set. In general, the LSTNet-based approach outputs the lowest number of false-negative alarms, with an average of 351 Type II errors (58% of failure detection rate). Note that from the total amount of simulated failures (857) the RNN-based model misclassify almost all the samples, with 856.25 (99.92%) Type II errors, on average. The LSTM performs slightly better, with 753.75 (87.95%) Type II errors, on average. This is explained by the poor forecasting performance reached out by both techniques, directly impacting the outlier detection. Regarding false-positive alarms (Type I errors), in average, all the techniques outputted quite small amounts of Type I errors with approximately 5.1 misclassifications. The explanation for this behavior is quite intuitive for the RNN- and LSTM-based approaches. Both models are biased by the poor modeling performance, as they tend to classify any sample as from the normal system condition (which is tied to their poor performance in detecting failures). On the other hand, the explanation for the low number of Type I errors in the LSTNet has a different direction. In fact, as verified before when analysing the forecasting metrics, the LSTNet performs a proper learning of the normal conditions; this reflects in good performance for classifying new samples as normal, and reasonable performance for detecting failures.

It is important to note that when training phase is well performed, the corresponding percentage of misclassifications on data derived from normal condition must be lesser than 1% as the threshold was defined on 99% of confidence over the training data; thus, if an algorithm misclassifies more than 1% of data from the normal condition it may be an indication of a poor training process caused by inappropriate algorithm or by a limited set of training data. In this case, all the techniques presented good performance, although, as summarized before, this is due to different reasons.

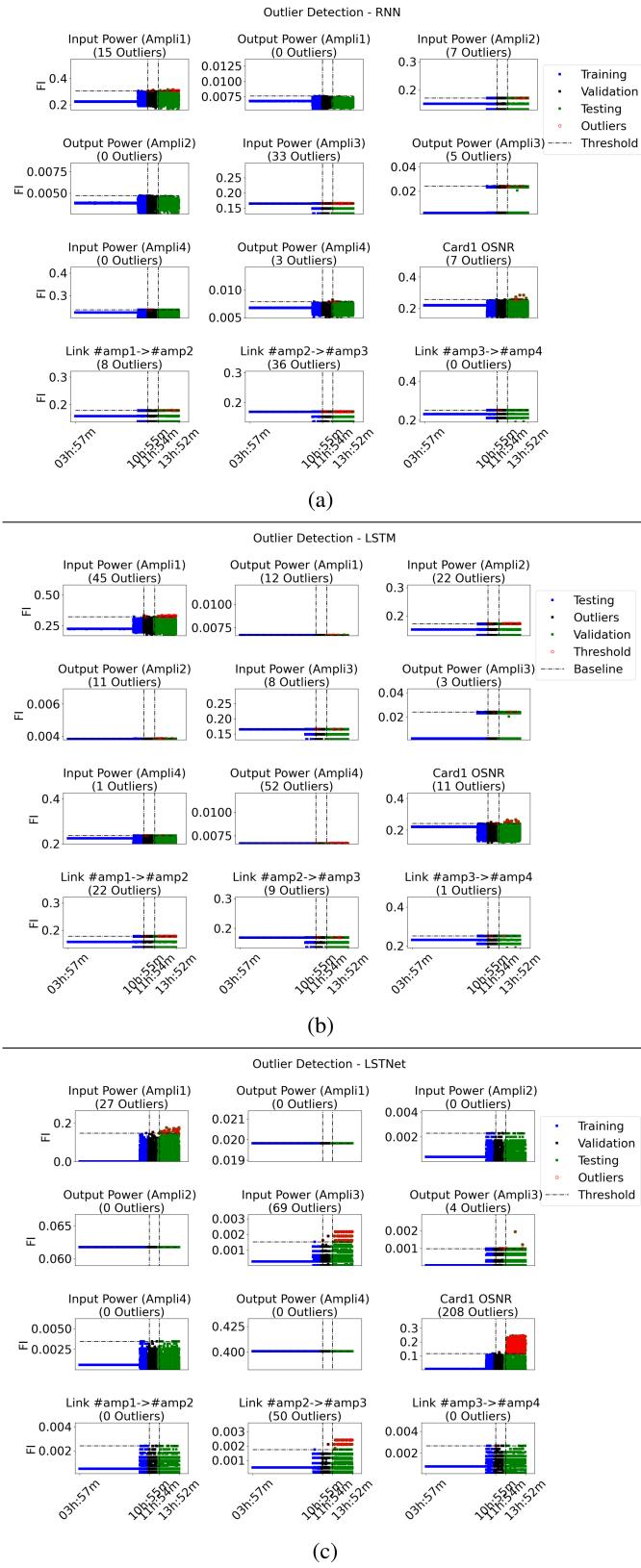


Fig. 7. Outlier detection results for a single trial: (a) RNN, (b) LSTM and (c) LSTNet. The number of detected outliers for a given optical parameter is indicated on the top of the corresponding figure.

The main issue with the LSTNet model is the same as for the forecasting results: stability. This is clearly seen when verifying the standard deviations for the Type II errors. As the

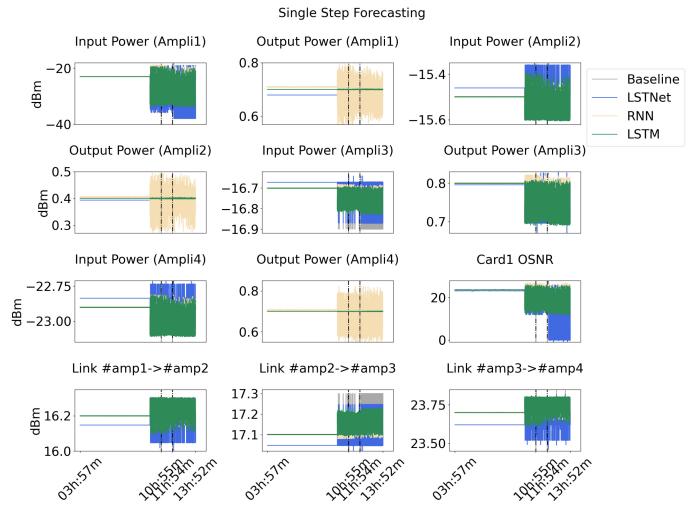


Fig. 8. Forecasting results for a single trial of each technique.

forecasting performance for the LSTNet is not stable for different trials, the outlier detection performance follows the same behavior, indicating an obvious correlation in performance between the forecasting and detection.

Finally, to visually infer the performance of the compared models, Fig. 7 and Fig. 8 show the outlier detection and forecasting results for a single trial of each compared technique, respectively. Specifically for the outlier detection, the number of detected failures for the test data (named as outliers) is indicated at the top of each optical parameter in the figure.

From the values of Tab. I and the comparisons of Fig. 7 is seen that although the amplitude of the forecasted values for the RNN and LSTM are within the range of the baseline data, the values are far from closely matching the actual data values, mainly for the case of the RNN model whose shows the worst performance among the compared techniques. On the other hand, the LSTNet shows outstanding performance with a visual fitting of the training, validation and test sets while keeping the best performance. However, as previously demonstrated by the numerical evaluation, the LSTNet is not overfitting the training data as it performs quite good predictions over the validation and test sets.

In the case of failure detection, the RNN detected few samples from the input power of amplifier 1 as failures, but also detected other samples as failures across the other parameters. The LSTM does a much better job in detecting failures in the amplifier 1 at the cost of misclassifying samples from the normal condition; at the same time, the model performs quite poorly in detecting failures in Card 1 and points out samples from other equipment as indications of failure. As mentioned previously, it is expected that only the amplifier 1 and Card 1 have indication of failure. On the other hand, the LSTNet performs quite reasonable in detecting failures for the amplifier 1 but misses some other samples related to failures as it misclassifies these samples as from the normal condition; the model also detects the majority of the failures in Card 1 at the cost of misclassifying several samples from the amplifier 3 as failures, which reflects also in wrong detections for the optical link between amplifier 2 and amplifier 3.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a novel unsupervised strategy to forecast and detect equipment failures in optical networks using the recent LSTNet architecture, capable of learning long- and short-term temporal patterns. The usage of the LSTNet for the particular tasks of forecast and failure detection allows to use a training dataset limited by only data from the normal network condition, without requiring explicit acquisition of data from failure conditions for training. Compared to the widely known RNN and LSTM models, the proposed approach coupled with the LSTNet outperforms the other techniques, mainly in terms of the forecasting metrics, but also for detecting failures. Regarding similar works, at the best of our knowledge, ours is the first to propose a methodology capable to exploit the usage of a limited dataset to forecast and detect failures of an optical system on a novelty detection fashion.

For future research, there are several promising directions in extending the work. First, further studies evaluating the performance of the LSTNet in performing multi-step ahead forecasting for anticipation of possible long-term failures. In second, a multi-threshold approach will be developed to classify the detected failures into different levels. This can be carried out by clustering the fault indicators, allowing to create thresholds for the different clusters that should be related to specific types of normal and abnormal system conditions. Finally, model scalability and failure localization must be further evaluated through additional tests, varying the number of optical devices in the test bed with different model parameters for sensitivity analysis.

REFERENCES

- [1] *View on 5G Architecture (Version 2.0)*, 5GPPP Architecture Working Group, Heidelberg, Germany, Jul. 2017.
- [2] C. Benzaid and T. Taleb, "AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions," *IEEE Netw.*, vol. 34, no. 2, pp. 186–194, Mar./Apr. 2020, doi: [10.1109/MNET.001.1900252](https://doi.org/10.1109/MNET.001.1900252).
- [3] S. Mwanje, G. Decarreau, C. Mannweiler, M. Naseer-ul Islam, and L. C. Schmelz, "Network management automation in 5G: Challenges and opportunities," in *Proc. IEEE 27th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2016, pp. 1–6, doi: [10.1109/PIMRC.2016.7794614](https://doi.org/10.1109/PIMRC.2016.7794614).
- [4] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 84–91, Apr. 2016, doi: [10.1109/MCOM.2016.7452271](https://doi.org/10.1109/MCOM.2016.7452271).
- [5] A. Cárdenas and D. Fernández, "Network slice lifecycle management model for NFV-based 5G virtual mobile network operators," in *Proc. IEEE Conf. Netw. Funct. Virtualization Softw. Defined Netw. (NFV-SDN)*, 2020, pp. 120–125, doi: [10.1109/NFV-SDN50289.2020.9289883](https://doi.org/10.1109/NFV-SDN50289.2020.9289883).
- [6] M.-A. Kourtis *et al.*, "5G network slicing enabling edge services," in *Proc. IEEE Conf. Netw. Funct. Virtualization Softw. Defined Netw. (NFV-SDN)*, 2020, pp. 155–160, doi: [10.1109/NFV-SDN50289.2020.9289880](https://doi.org/10.1109/NFV-SDN50289.2020.9289880).
- [7] S. K. Syed-Yusof, P. E. Numan, K. M. Yusof, J. B. Din, M. N. Bin Marsono, and A. J. Onumanyi, "Software-defined networking (SDN) and 5G network: The role of controller placement for scalable control plane," in *Proc. IEEE Int. RF Microw. Conf. (RFM)*, 2020, pp. 1–6, doi: [10.1109/RFM50841.2020.9344741](https://doi.org/10.1109/RFM50841.2020.9344741).
- [8] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," *China Commun.*, vol. 15, no. 10, pp. 86–98, Oct. 2018, doi: [10.1109/CC.2018.8485472](https://doi.org/10.1109/CC.2018.8485472).
- [9] S. K. Routray and K. P. Sharmila, "Software defined networking for 5G," in *Proc. 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, 2017, pp. 1–5, doi: [10.1109/ICACCS.2017.8014576](https://doi.org/10.1109/ICACCS.2017.8014576).
- [10] "Zero Touch Network & Service Management (ZSM)," ETSI Technical Committee. [Online]. Available: <https://www.etsi.org/technologies/zero-touch-netwtrk-service-management> (accessed Jun. 2, 2021).
- [11] E. Z. Nurit Sprecher, "ETSI ZSM Architectural Framework for End-to-End Service and Network Automation," [Online]. Available: <https://sdn.ieee.org/newsletter/november-2018/etsi-zsm-architectural-framework-for-end-to-end-service-and-network-automation> (accessed Jun. 2, 2021).
- [12] E. Coronado, S. N. Khan, and R. Riggio, "5G-Empower: A software-defined networking platform for 5G radio access networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 715–728, Jun. 2019, doi: [10.1109/TNSM.2019.2908675](https://doi.org/10.1109/TNSM.2019.2908675).
- [13] H. Kabir, M. H. Bin Mohsin, and R. Kantola, "Implementing a security policy management for 5G customer edge nodes," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, 2020, pp. 1–8, doi: [10.1109/NOMS47738.2020.9110321](https://doi.org/10.1109/NOMS47738.2020.9110321).
- [14] G. Carrozzo *et al.*, "AI-driven zero-touch operations, security and trust in multi-operator 5G networks: A conceptual architecture," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2020, pp. 254–258, doi: [10.1109/EuCNC48522.2020.9200928](https://doi.org/10.1109/EuCNC48522.2020.9200928).
- [15] M. Xie *et al.*, "AI-driven closed-loop service assurance with service exposures," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2020, pp. 265–270, doi: [10.1109/EuCNC48522.2020.9200943](https://doi.org/10.1109/EuCNC48522.2020.9200943).
- [16] J. B. Porch, C. H. Foh, H. Farooq, and A. Imran, "Machine learning approach for automatic fault detection and diagnosis in cellular networks," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2020, pp. 1–5, doi: [10.1109/BlackSeaCom48709.2020.9234962](https://doi.org/10.1109/BlackSeaCom48709.2020.9234962).
- [17] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5G networks," 2020, *arxiv:2003.03474*.
- [18] D. Y. Shimizu, K. S. Mayer, J. A. Soares, and D. S. Arantes, "A deep neural network model for link failure identification in multi-path ROADM based networks," in *Proc. Photon. North (PN)*, 2020, p. 1, doi: [10.1109/PN50013.2020.9166978](https://doi.org/10.1109/PN50013.2020.9166978).
- [19] A. Mohamed *et al.*, "An inter-disciplinary modelling approach in industrial 5G/6G and machine learning era," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2020, pp. 1–6, doi: [10.1109/ICCWorkshops49005.2020.9145434](https://doi.org/10.1109/ICCWorkshops49005.2020.9145434).
- [20] A. K. Saxena, S. Sinha, and P. Shukla, "General study of intrusion detection system and survey of agent based intrusion detection system," in *Proc. Int. Conf. Comput. Commun. Autom. (ICCCA)*, 2017, p. 471, doi: [10.1109/CCAA.2017.8229866](https://doi.org/10.1109/CCAA.2017.8229866).
- [21] R. Parsamehr *et al.*, "A novel intrusion detection and prevention scheme for network coding-enabled mobile small cells," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1467–1477, Dec. 2019, doi: [10.1109/TCSS.2019.2949153](https://doi.org/10.1109/TCSS.2019.2949153).
- [22] A. Blazquez-Garcia, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," 2020, *arXiv:2002.04236*.
- [23] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, 2021, doi: [10.1145/3439950](https://doi.org/10.1145/3439950).
- [24] "Optical communication and networking equipment market," AM Res., Rajasthan, India, Rep. MSR4345799, Feb. 2021.
- [25] G. Lai, W. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2018, pp. 1–11. [Online]. Available: <http://arxiv.org/abs/1703.07015>
- [26] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 54, no. 108, pp. 1421–1441, 2019, doi: [10.1007/s10994-019-05815-0](https://doi.org/10.1007/s10994-019-05815-0).
- [27] H. Ouyang, X. Wei, and Q. Wu, "Agricultural commodity futures prices prediction via long- and short-term time series network," *J. Appl. Econ.*, vol. 22, no. 1, pp. 468–483, 2019, doi: [10.1080/15140326.2019.1668664](https://doi.org/10.1080/15140326.2019.1668664).
- [28] V. Campos, B. Jou, X. Giró-i-Nieto, J. Torres, and S. Chang, "Skip RNN: learning to skip state updates in recurrent neural networks," 2018, *arXiv:1708.06834*.
- [29] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2014, doi: [10.1109/SURV.2013.052213.00046](https://doi.org/10.1109/SURV.2013.052213.00046).
- [30] S. Kwon, S. Park, H. Cho, Y. Park, D. Kim, and K. Yim, "Towards 5G-based IoT security analysis against Vo5G eavesdropping," *Computing*, vol. 103, no. 1, pp. 425–447, 2021, doi: [10.1007/s00607-020-00855-0](https://doi.org/10.1007/s00607-020-00855-0).

- [31] K. Mayer, J. Soares, R. Pinto, C. Rothenberg, D. Arantes, and D. Mello, "Machine-learning-based soft-failure localization with partial software-defined networking telemetry," *J. Opt. Commun. Netw.*, vol. 13, no. 10, pp. E122–E131, 2021, doi: [10.1364/JOCN.424654](https://doi.org/10.1364/JOCN.424654).
- [32] S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore, "Machine-learning-based soft-failure detection and identification in optical networks," in *Proc. Opt. Fiber Commun. Conf. Expo. (OFC)*, 2018, pp. 1–3.
- [33] J. Santos, P. Leroux, T. Wauters, B. Volckaert, and F. De Turck, "Anomaly detection for smart city applications over 5G low power wide area networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, 2018, pp. 1–9, doi: [10.1109/NOMS.2018.8406257](https://doi.org/10.1109/NOMS.2018.8406257).
- [34] "Self-Organizing Networks," 3GPP Initiative. [Online]. Available: <https://www.3gpp.org/technologies/keywords-acronyms/105-son> (accessed Jun. 2, 2021).
- [35] X. Qin, S. Tang, X. Chen, D. Miao, and G. Wei, "SQoE KQIs anomaly detection in cellular networks: Fast online detection framework with hourglass clustering," *China Commun.*, vol. 15, no. 10, pp. 25–37, Oct. 2018, doi: [10.1109/CC.2018.8485466](https://doi.org/10.1109/CC.2018.8485466).
- [36] J. Burgueño, I. de-la-Bandera, J. Mendoza, D. Palacios, C. Morillas, and R. Barco, "Online anomaly detection system for mobile networks," *Sensors*, vol. 20, no. 24, p. 7232, 2020, doi: [10.3390/s20247232](https://doi.org/10.3390/s20247232).
- [37] M. Wang and S. Handurukande, "A streaming data anomaly detection analytic engine for mobile network management," in *Proc. Symposia Workshops Ubiquitous Auton. Trusted Comput.*, 2016, pp. 722–729, doi: [10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0117](https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0117).
- [38] J. Wu, P. P. C. Lee, Q. Li, L. Pan, and J. Zhang, "CellPAD: Detecting performance anomalies in cellular networks via regression analysis," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, 2018, pp. 1–9, doi: [10.23919/IFIPNetworking.2018.8697027](https://doi.org/10.23919/IFIPNetworking.2018.8697027).
- [39] R. Song and F. Liu, "Real-time anomaly traffic monitoring based on dynamic k-NN cumulative-distance abnormal detection algorithm," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst.*, 2014, pp. 187–192, doi: [10.1109/CCIS.2014.7175727](https://doi.org/10.1109/CCIS.2014.7175727).
- [40] "Global coherent optical equipment market (2019 to 2026)—By technology, equipment, application and end-user," Res. Markets, Dublin, Ireland, Rep. 5019876, 2020.
- [41] A. Sgambelluri, A. Pacini, F. Paolucci, P. Castoldi, and L. Valcarenghi, "Reliable and scalable kafka-based framework for optical network telemetry," *J. Opt. Commun. Netw.*, vol. 13, no. 10, pp. E42–E52, 2021, doi: [10.1364/JOCN.424639](https://doi.org/10.1364/JOCN.424639).
- [42] InRete Lab, 2021, "Optical Failure Dataset," Scuola Superiore Sant'Anna. [Online]. Available: <https://github.com/Network-And-Services/optical-failure-dataset>



Moisés Felipe Silva received the Ph.D. degree in electrical engineering from the Federal University of Pará, Belém, Brazil, in 2020. He is a Postdoctoral Fellow with the Scuola Superiore Sant'Anna, Pisa, Italy, whose research efforts are on the development of intelligent fault detection algorithms for network automation in 5G environments. His scientific career has primarily focused on the development of machine learning and artificial intelligence techniques for data-driven Structural health monitoring applications with emphasis on damage identification.

He has also published numerous peer-reviewed articles in leading international journals and international conference proceedings (source Google Scholar, June 2021), honored with one best paper award. Other research interests include intelligent systems, evolutionary computing, outlier detection, and theoretical computing.



Alessandro Pacini received the bachelor's degree in computer science from the University of Camerino in 2018, and the joint master's degree in computer science and networking from the University of Pisa and the Scuola Superiore Sant'Anna of Pisa, Italy, where he is currently pursuing the Ph.D. degree in emerging digital technologies, curriculum Photonic Technologies. During this master's degree, he won a one-year research scholarship at the SSSA, focused on building up a scalable and reliable monitoring architecture for optical networks. His main research interests are in the field of the closed loop automation, with a specific focus on reusing existing network architectures to be shifted over a Zero-Touch paradigm.



Andrea Sgambelluri has been an Assistant Professor with the Scuola Superiore Sant'Anna of Pisa, Italy, since 2019. He published around 100 papers (source Google Scholar, May 2021) in international journals and conference proceedings. His main research interests are in the field of control plane techniques for both packet and optical networks, including software defined networking protocol extensions, network reliability, industrial Ethernet, switching, segment routing application, YANG/NETCONF solutions for the dynamic management, telemetry, (re)programming and monitoring of optical devices. In March 2015, he won the grand prize at 2015 OFC Corning Outstanding Student Paper Competition with the paper "First Demonstration of SDN-based Segment Routing in Multi-Layer Networks."



Luca Valcarenghi (Senior Member, IEEE) has been an Associate Professor with the Scuola Superiore Sant'Anna of Pisa, Italy, since 2014. He published almost 300 papers (source Google Scholar, May 2020) in international journals and conference proceedings. His main research interests are optical networks design, analysis, and optimization; communication networks reliability; energy efficiency in communications networks; optical access networks; zero touch network and service management; experiential networked intelligence; 5G technologies and beyond. He received a Fulbright Research Scholar Fellowship in 2009 and a JSPS "Invitation Fellowship Program for Research in Japan (Long Term)" in 2013.