Multi-scale CNN-TCN hybrid architecture for long-horizon prediction of OLT load rate in a functional FTTH network*

Zacharia Damoue [1,*,†], Mohamed Kaba Keita[2,†], Moustapha Der[1,†], Boudal Niang[1,†]and El Hadji Makhtar Faye[1,†]

[1] *Ecole Supérieure Multinationale des Télécommunications (ESMT), Dakar, Sénégal*
[2] *Université Cheikh Anta DIOP (UCAD), Dakar, Sénégal*

Abstract

Optical Line Terminals (OLTs) are the central components of Fiber To The Home (FTTH) networks based on Gigabit Passive Optical Network (GPON) technology, ensuring traffic distribution between the core network and end users. In West African networks, the rapid growth in broadband demand, amplified by events such as football matches and daily peak hours, is placing increasing pressure on this equipment, making proactive overload prevention a critical operational challenge. Traditional management approaches remain primarily reactive, intervening only after congestion has already occurred. This article presents a comprehensive comparative study of deep learning architectures for long-term OLT load prediction using real-world operational data collected from a major operator's FTTH network.

Several prediction models were designed and compared, ranging from classical approaches (ARIMA, Prophet) to advanced Deep Learning architectures (LSTM, CNN+LSTM, CNN+TCN, CNN+TCN+Attention, and CNN+TCN+LSTM+Attention). The results show the predominance of the hybrid CNN-TCN (Convolutional Neural Network - Temporal Convolutional Networks) model, providing reliable forecasting over a five-day period across three Broadband Network Gateway (BNG) sites. The proposed models target a prediction horizon of 5 days (1,440 time steps), far exceeding the short-term horizons reported in previous work.

Indeed, the proposed multi-scale CNN+TCN architecture achieves R^2=0.88 and MAE=0.032, surpassing all benchmarks. Despite certain limitations, the system provides a solid foundation for anticipating network overloads and implementing a self-adaptive network capable of automatically adjusting resources according to the predicted load.

Keywords

OLT load rate prediction, FTTH networks, time convolution network, CNN-TCN, Streamsets, Elasticsearch.

1. Introduction

The rapid expansion of FTTH infrastructure in sub-Saharan Africa is reshaping digital access for millions of users. In Senegal, a major operator has deployed a large-scale FTTH network based on GPON technology, serving residential and business customers in key urban areas. At the heart of these optical access networks are OLTs, central equipment responsible for managing and distributing traffic between the core network and subscriber terminals (ONTs) [1].

The increasing adoption of bandwidth-intensive applications, including high-definition video streaming, online gaming, and remote working platforms, is placing growing pressure on OLT infrastructure. This pressure is further amplified by one-off events such as football match broadcasts, religious holidays, and daily usage peaks, which generate sudden and often unpredictable spikes in

* *CITA 2026-Land, Smart Cities and Sustainable Development, 25-26 June 2026, Cotonou, Benin*
* Corresponding author.
† These authors contributed equally.
zacharia.damoue@esmt.sn (Z. Damoue); keita.mohamed@esp.sn (M.K. Keita); moustapha.der@esmt.sn (M. Der); boudal.niang@esmt.sn (B. Niang); makhtarfaye02@gmail.com (E.H.M. Faye).
0000-0002-8970-0948 (Z. Damoue); 0009-0004-5601-4898 (M.K. Keita); 0009-0007-7489-6363 (M. Der); 0000-0002-9458-0063 (B.Niang); 0009-0009-0506-2706 (E.H.M. Faye)

network load. When an OLT is overloaded, users experience increased latency, packet loss, and degraded quality of service, a situation that directly impacts customer satisfaction and the operator's reputation.

Current mitigation strategies for telecommunications operators include Dynamic Bandwidth Allocation (DBA), load balancing between PON cards, capacity enhancements, and reactive monitoring of performance indicators. However, these approaches are inherently reactive; they respond to congestion once it has already occurred, rather than proactively anticipating it. Some OLT equipment manufacturers, such as Huawei, offer integrated overload control mechanisms that regulate CPU (Central Processing Unit) load, but these remain reactive and do not offer predictive capabilities [2].

This work addresses this gap by developing and evaluating a system capable of predicting OLT load factors over a 5-day horizon with a 5-minute granularity. The proposed approach relies on approximately two (2) million real-world operational measurements collected from three BNG sites in Senegal, making this study one of the few entirely based on operator data from production rather than simulations.

The main contributions of this article are:

(1) A complete benchmark from seven prediction architectures of classical statistical models to advanced deep learning ensembles on real FTTH operator data.

(2) An original multi-scale CNN+TCN architecture reaching R^2=0.88 over a prediction horizon of 5 days.

(3) A production-ready integration pipeline connecting Elasticsearch, model inference and Power BI for operational deployment at the operator.

The remainder of this article is organized as follows: Section II formalizes the problem; Section III reviews related work; Section IV describes the data and collection pipeline; Section V details the proposed methodology and architectures; Section VI presents the experimental results; and Section VII concludes with perspectives.

2. Problem Formulation

Let $\{OLT_1, OLT_2, ..., OLT_n\}$ be the set of n Optical Line Terminals deployed in a given network area. For each OLT k, traffic measurements are collected every Δt = 5 minutes. The load factor at time t is defined by:

$$\tau(k,t) = C(k,t) \;/\; C\_max(k) \quad (1)$$

With:

- C(k,t) = tcp_dw(k,t) + udp_dw(k,t) + icmp_dw(k,t) + other_dw(k,t) represents the total downlink load in bytes,
- C_max(k) the maximum nominal capacity of the OLT k.

The value $\tau(k,t) \in [0,1]$, where values greater than 0.7 indicate a risk of moderate overload and those greater than 0.9 a critical saturation.

The prediction problem is formulated as follows: given a historical window of W = 2016 consecutive observations (7 days at a 5 min resolution) for OLT k, learn a mapping:

$$f : \tau(k,\ t-W+1 : t \rightarrow \hat{\tau}\ (k, t+1 : t+H) \quad (2)$$

With :

- H = 1440 is the prediction horizon (5 days),
- $\hat{\tau}$ (k, t+1:t+H) the vector of predicted load rates.

This formulation captures both the multi-step nature of the prediction (5 days in advance) and the multi-entity nature of the problem (one model per OLT or one global model with an integration of OLTs).

A prediction is considered operationally useful if it reliably identifies impending overload events ($\tau > 0.7$) at least 24 hours in advance, enabling network operations teams to take preventative measures such as load balancing or targeted capacity boosting.

Once the problem has been formulated, a review of related work is necessary.

3. Related Work

3.1. Classical Statistical Approaches

ARIMA (AutoRegressive Integrated Moving Average), introduced by Box and Jenkins [3], remains a widely used reference for time series forecasting. ARIMA captures linear autocorrelation via its AutoRegressive (AR), Integration (I) and Moving Average (MA) components, and handles non-stationarity by differentiation.

Prophet [4], developed by Meta, breaks down a serie into trend, seasonality and holiday effects, making it better suited to series with a strong calendar character.

However, both approaches assume linearity and struggle to capture the complex non-linear dynamics of FTTH traffic, particularly over long multi-day horizons.

3.2. Deep Learning for Network Traffic Prediction

Long Short-Term Memory (LSTM) networks [5], equipped with gate mechanisms designed to capture long-range temporal dependencies, have proven effective for network traffic forecasting.

Labonne et al. [6] demonstrated that LSTM substantially outperforms ARIMA and MLP (Microbial Load Predictor) for predicting link utilization rates (RMSE ≈ 0.012, error < 3%), but on simulated data and a very short horizon of 15 seconds.

Waczyńska et al. [7] evaluated CNN, LSTM, CNN-LSTM and Conv-LSTM models on the CERN LHCOPN network, finding that Conv-LSTM offers the best stability, but their study is limited to two network links and a horizon of less than 30 minutes.

Hatem et al. [8] proposed Deep-DBA, an LSTM-based predictive mechanism for dynamic bandwidth allocation in NG-EPON networks, achieving MSE $\approx 8\times10^{-6}$. Although effective, this work targets very short-term allocation adjustments rather than multi-day load forecasting, and relies entirely on simulated data.

3.3. Time Convolutional Networks (TCNs)

Bai et al. [9] introduced Temporal Convolutional Networks (TCNs), demonstrating that dilated 1D causal convolutions with residual connections equal or surpass recurrent architectures on a wide range of sequential modeling tasks. TCNs offer key advantages for long-horizon forecasting: (i) parallelism to training, (ii) exponentially growing receptive fields via dilation factors, and (iii) a stable gradient flow due to residual connections.

WaveNet [10], pioneering work on dilated causal convolutions, demonstrated the power of this approach for audio generation, which motivated its adaptation to network traffic.

3.4. Positioning of this Work

To our knowledge, no previous work addresses the following combination: (i) long-horizon prediction of OLT load factor (5 days), (ii) real operational data from a production FTTH network, and (iii) multi-scale CNN+TCN architecture. Table I summarizes related work and its main limitations compared to the proposed approach.

Table I. Summary of Related Work

Reference	Method	Main result / Limit
Labonne et al. [4]	ARIMA/LSTM	LSTM RMSE ≈ 0.012 on simulated links; horizon 15 s only
Waczyńska et al. [5]	CNN-LSTM	Conv-LSTM is more stable on CERN LHCOPN; 2 links, horizon < 30 min
Hatem et al. [6]	Deep-DBA LSTM	MSE ≈ 8×10^{-6} for BW allocation; simulated data, short horizon
Bai et al. [8]	TCN	TCN outperforms LSTM on sequences; dilated causal convolutions

After the literature review, it is essential to focus on data collection, because without data, no predictions can be made.

4. Background

This work focuses on the operation of FTTH optical access networks, deployed by telecommunications operators to provide broadband services to end users. In these architectures, Optical Line Terminals (OLTs) play a central role in managing and distributing traffic between the network core and the optical distribution networks (PONs).

The evolution of digital usage, combined with the increase in the number of subscribers and population growth observed in certain areas, is leading to a significant and sometimes unpredictable rise in network traffic. This situation can be exacerbated by one-off events, such as sporting events, hourly peaks, or technical incidents, generating sudden fluctuations in load at the OLTs.

Traditionally, to cope with this increase in demand, operators scale or resize the capacity of network equipment, which involves hardware investments and medium- or long-term planning. However, these solutions remain costly and do not always allow for effective anticipation of sudden traffic fluctuations, hence the need to implement smarter and more proactive load management mechanisms.

5. Data Collection and Description

5.1. Network Infrastructure and Data Sources

This study is conducted on the FTTH network of the major telecommunications operator in Senegal. The infrastructure relies on GPON OLT equipment from two manufacturers: Huawei (MA5800 series) and Nokia (ISAM 7360 series). The Huawei OLTs have an integrated overload control mechanism for CPU regulation, while the Nokia OLTs do not offer an equivalent reactive protection mechanism, making predictive tools even more critical for the latter. The data is collected from three BNGs at different sites covering distinct urban and suburban service areas with heterogeneous traffic profiles. Each BNG queries its connected OLTs every 5 minutes, storing the measurements in a relational SQL database.

5.2. Collection Pipeline

A StreamSets Data Collector pipeline automates data ingestion; SQL data is extracted via JDBC, preprocessed through several transformation steps (stream selector, Python evaluator, field renamer, type converter), and then stored in Elasticsearch. A cron job triggers this pipeline at regular intervals, ensuring the availability of fresh data. A dedicated Python class (ElasticFetcher) handles data retrieval for model training and inference. Figure 1 shows the streamset pipeline used for data

retrieval.

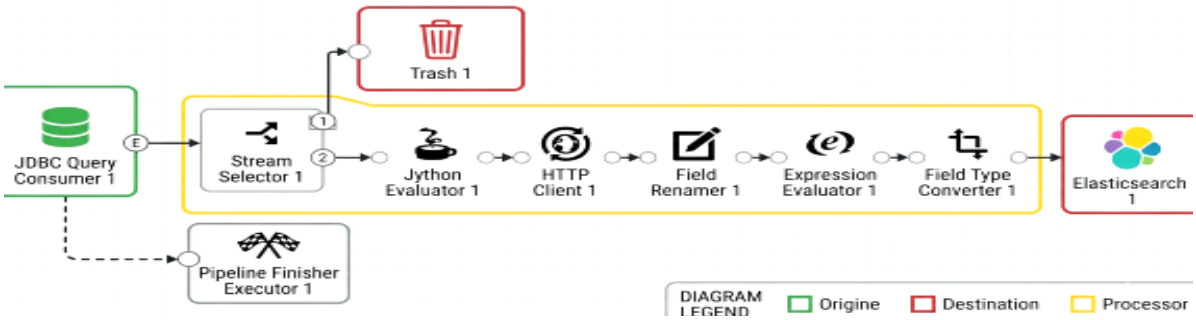

Figure 1. Data retrieval pipeline streamset

5.3. Dataset Characteristics

The final dataset contains approximately two (2) million observations covering several months of continuous operation. Screenshot 1 shows description of the collected data.

	capacite_max	icmp_dw	other_dw	tcp_dw	udp_dw	taux_de_charge	charge_canal
count	1.999232e+06	1.999232e+06	1.999232e+06	1.999232e+06	1.999232e+06	1999232.00	1.999232e+06
mean	3.009261e+10	2.475009e+05	6.926494e+04	3.094216e+09	2.814071e+09	0.19	5.908603e+09
std	1.578793e+10	4.185680e+05	9.038780e+05	3.187635e+09	3.079417e+09	0.17	6.248920e+09
min	1.000000e+10	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.00	0.000000e+00
25%	2.000000e+10	3.817472e+04	0.000000e+00	5.656418e+08	4.492648e+08	0.04	1.022144e+09
50%	2.000000e+10	1.608090e+05	0.000000e+00	2.064947e+09	1.751471e+09	0.14	3.823000e+09
75%	4.000000e+10	3.683123e+05	0.000000e+00	4.588133e+09	4.097727e+09	0.30	8.683613e+09
max	8.000000e+10	3.558295e+08	2.536902e+08	1.932374e+10	2.075696e+10	0.90	3.475333e+10

Screenshot 1: Description of digital data

The target variable, the load factor τ(k,t), is calculated using a pairwise method (OLT, timestamp). The main descriptive statistics reveal:

- Average load rate: 19% indicating overall healthy network operation.
- Standard deviation: 17% reflecting significant variability according to OLT and periods.
- Median: 14% confirming that the distribution is right-skewed with episodic peaks.
- Maximum observed: 90% revealing critical saturation events threatening the quality of service.

Screenshot 2 shows the various instances where the threshold was exceeded.

```
Global overruns:
> 70% : 11587 occurrences
> 80% : 3484 occurrences
> 90% : 1 occurrence
```

Screenshot 2: Showing threshold exceedances

Analysis of threshold exceedances shows that moderate overload events ($\tau > 70\%$) are relatively frequent, while critical saturation ($\tau > 90\%$) remains rare but has a significant impact. This class imbalance guides the choice of evaluation metrics.

5.4. Temporal Patterns

Exploratory analysis reveals strong cyclo-stationarity. A clear daily cycle is observed with four phases:

- Off-peak period (0h–6h, $\tau \approx 5–15\%$).
- gradual morning rise (6am–11am, $\tau \approx 15–25\%$).
- daytime plateau (11am–5pm, $\tau \approx 25–35\%$).
- evening peak (6pm–11pm, $\tau \approx 35–45\%$), corresponding to the return of residents to their homes

and their engagement in streaming, games and social networks.
Figure 2 shows the load factor as a function of the time of day

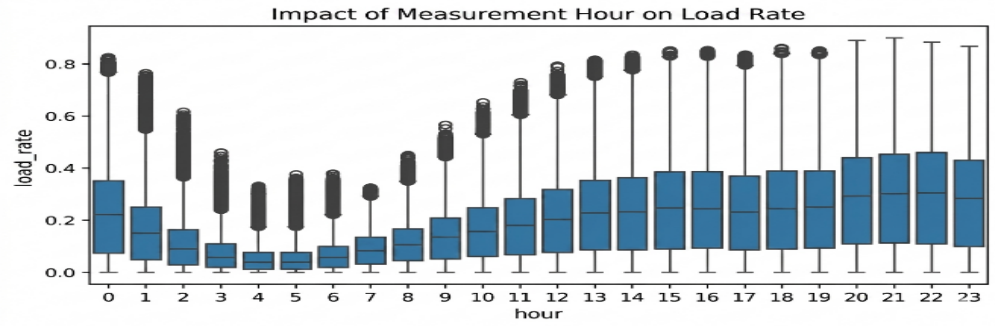

Figure 2: Box plot of the load factor by time of day

The weekly profiles show similar dynamics during the week ($\bar{\tau} \approx 28–30\%$), with a slight difference on weekends characterized by a shifted morning peak. Figure 3 shows the load factor by day of the week.

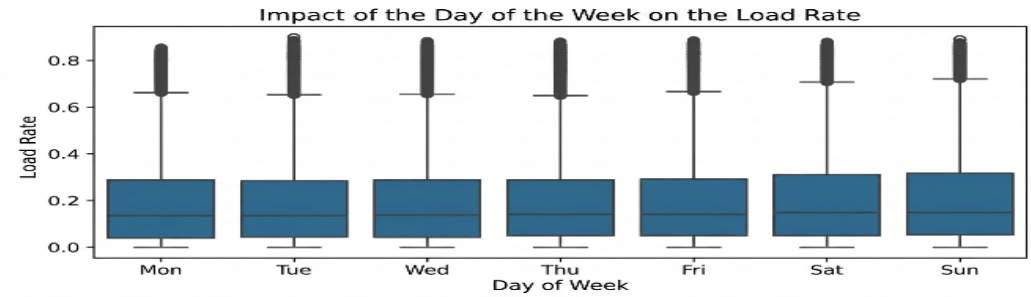

Figure 3: Box plot of the occupancy rate by day of the wee

Exceptional values are attributable to equipment failures, live football match broadcasts, and other episodic events.

An analysis of the collected data allows us to propose the methodology used.

6. Proposed Methodology

6.1. Preprocessing and Sequence Generation

After retrieving the raw data, a CustomPreprocessor normalizes the date formats, removes duplicates, and selects the three columns (OLT name, timestamp, load rate). For deep learning models, a SequenceGenerator transforms the time series into supervised training windows: input sequences X of length W = 2016 (7 days) and output vectors Y of length H = 1440 (5 days). A MinMaxScaler normalizes the load rates to [0,1]. The time slicing allocates the first 80% of each OLT's history to training and the remaining 20% to testing, strictly preserving chronological order (shuffle=False).

6.2. Reference Models

ARIMA [3]: Grid search on the parameters (p,d,q) for each OLT, with evaluation on the last 1440 training values. ARIMA serves as the main classical reference.

Prophet [4]: Incorporates trend, seasonality (daily/weekly), and Senegalese national holidays as external regressors. Prophet handles missing values and provides confidence intervals.

Simple LSTM [5]: Single-layer LSTM of 128 units, fed by the raw time series with an early-fusion concatenated OLT identity integration (8 dimensions), establishes the reference in deep learning.

6.3. Hybrid Architectures

Three hybrid models were evaluated:

CNN+LSTM:

Three 1D CNN blocks (kernels k=7, k=3, k=3 with dilation=2) extract local features with causal padding and MaxPooling, halving the sequence length before passing it to a 128-unit LSTM. The output is dense layers predicting the horizon by 1440 steps.

CNN + TCN + Attention:

After the 1D CNN blocks, a multi-scale TCN block is introduced. Three parallel TCN branches with dilation factors D=1 (short term), D=2 (medium term), and D=4 (long term) are concatenated, followed by deeper TCN blocks with D=8, D=16, and D=32 for ultra-long-term dependencies.A temporal attention mechanism is added after the deep TCN. Attention scores (calculated via Dense + tanh + Softmax) weight the contribution of each time step to the final prediction, allowing the model to focus on the most informative historical periods.

CNN + TCN + Attention +LSTM

The model integrates the four components into a unified pipeline. As conceptually illustrated in Fig. 4, the processing flow is as follows:

(1) CNN feature extraction (3 blocks with causal padding, kernels k=7 and k=5, MaxPooling reducing 2016⟶1008 time steps).

(2) Multiscale TCN (parallel branches D=1/2/4, concatenated; then deep blocks D=8/16/32 with residual connections).

(3) Hybrid LSTM (applied to the TCN output with return_sequences =True; residual addition to the TCN output, followed by a Conv1D merge).

(4) Attention mechanism (Dense + tanh + Softmax producing weights per time step; aggregation of the weighted context vector).

(5) Prediction head (last step + global pooling by average; three dense layers; linear output of H=1,440 neurons).

CNN blocks capture short-term local patterns (sub-hourly fluctuations), multi-scale TCN captures medium- and long-term cyclic dependencies (daily and weekly cycles), LSTM adds sequential refinement sensitive to scheduling, and the attention mechanism identifies the historical moments most predictive of future load. Dropout (0.2–0.3) is applied for regularization. Training uses the Adam optimizer, Early Stopping (patience=5), Reduce LR On Plateau (factor=0.5, patience=3), and Model Checkpoint on the best validation loss, over 30 epochs with a batch size of 64.

6.4. Proposed Architecture

CNN+TCN: Following the 1D CNN blocks, a multiscale TCN block is introduced. Three parallel TCN branches with dilation factors D=1 (short term), D=2 (medium term), and D=4 (long term) are concatenated, followed by deeper TCN blocks with D=8, D=16, and D=32 for ultra-long-term dependencies. Formally, the dilated causal convolution with dilation factor d is:

$$f(t) = \Sigma_i \; f(i) \cdot \; x(t \; - \; d \cdot i), \; t \; \in \; [0, \; T] \tag{3}$$

With:

- y(t) is the output at time step t
- f(i) is the filter of size k
- d is the dilation factor
- $x(t - d \cdot i)$is the input sequence

The dilation factor "d" controls the spacing between the filter elements, allowing the network to expand its receptive field exponentially with depth, without increasing the number of parameters.

Residual layers are added to maintain the gradient flow, and the overall transformation can be

represented by the following equation:

$$H(x) = F(x) + x \tag{4}$$

With:

- H(x) is the output
- F(x) is the function learned by the convolution layers
- x is the input that is transmitted via the residual connection

Calculating the receptive field of a convolutional neural network (CNN) is essential because it indicates the amount of input the network can handle. The receptive field R is determined by the network depth d, the nucleus size k, and the dilation factor f:

$$R = 1 + (k-1) \cdot \sum_{i=0}^{d-1} f^i \tag{5}$$

This equation shows the behavior of TCNs which develop their receptive field exponentially by increasing the dilation factor, giving the network the ability to model long-range dependencies.

The CNN+TCN architecture combines the advantages of local feature extraction using CNNs with the efficient temporal modeling of TCNs. Figure 4 shows the architecture of the hybrid CNN+TCN model.

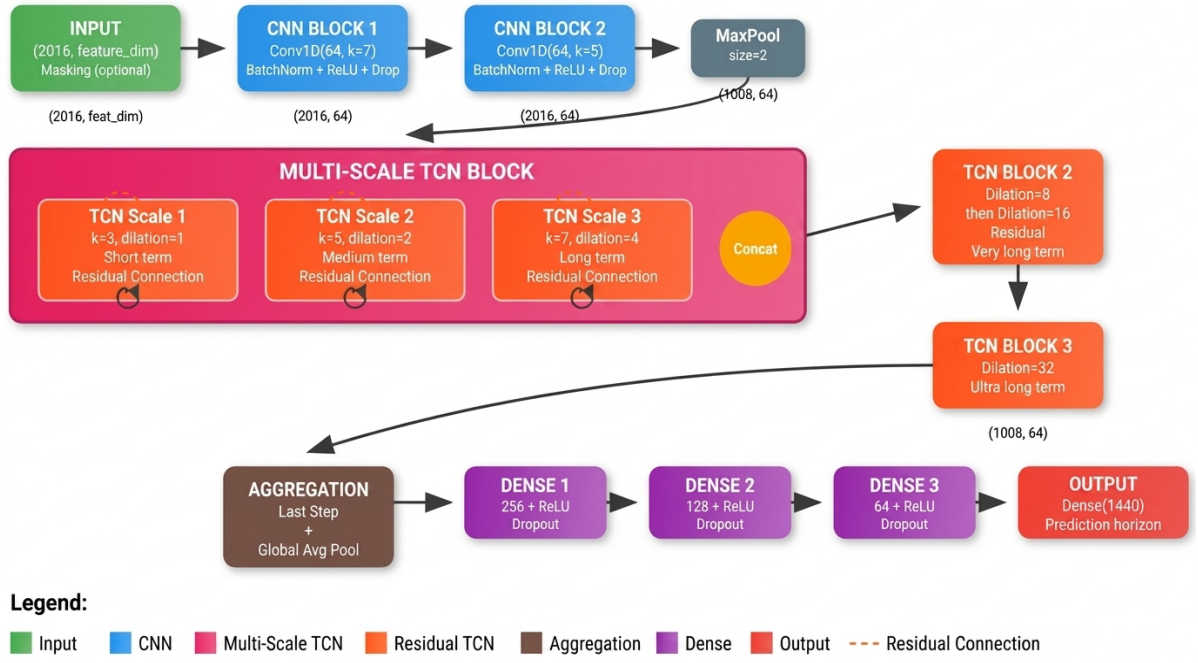

Figure 4. Architecture of the CNN + TCN model

6.5. Evaluation Metrics

Four metrics are used for a comprehensive evaluation:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{7}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad (8)$$

The MAE provides a linear penalty, the MSE amplifies large errors (important for high-load events), the RMSE is comparable to the MAE in the original units, and the R^2 measures the proportion of variance explained by the model.

Now that the evaluation metrics have been defined, it is time to apply them to the models under study.

7. Experimental results

7.1. Comparative Performance

Table II presents the aggregate metrics across all tests for the seven models, averaged across all OLTs. Predictions are generated on the retained 20% of the time series from each OLT, then concatenated for the overall evaluation. All models were trained on a system equipped with 2 NVIDIA T4 GPUs (each with 15 GiB of GPU memory), 30 GiB of RAM and 57.6 GiB of disk storage.

Table 2. Comparison of Models (Test Set, 5-Day Horizon)

Model	MAE	MSE	RMSE	R^2	Time
ARIMA	0,150	0,030	0,190	–1,18	-------
Prophet	0,120	0,024	0,150	0,02	-------
LSTM	0,064	0,007	0,086	0,53	-------
CNN+LSTM	0,035	0,003	0,045	0,85	1 h
CNN+TCN (proposed)	0,032	0,002	0,048	0,88	2 h
CNN+TCN+Att.	0,034	0,003	0,052	0,86	2 h
CNN+TCN+Att.+LSTM	0,031	0,002	0,045	0,88	6 h

7.2. Analysis

Classical statistical models prove inadequate for this task.

ARIMA achieves a negative R^2 (–1.18), indicating that it performs worse than a naive mean-based predictor a consequence of its inability to handle multi-scale nonlinear dynamics and long-memory dependencies in FTTH load series.

Prophet improves marginally (R^2=0.02) by incorporating seasonality and holidays, but remains fundamentally limited by its linear trend assumption.

Simple LSTM offers a substantial leap (R^2=0.53), confirming the need for deep learning for this task. Its sequential modeling captures patterns day after day, but its hidden state bottleneck limits performance over 5-day time horizons.

CNN+LSTM (R^2=0.85) demonstrates the value of convolutional preprocessing: local feature extraction by CNN reduces noise in the input before LSTM models temporal dependencies.

The complete CNN+TCN+Attention+LSTM architecture achieves R^2=0.88 with the best overall MAE (0.031) and MSE (0.0022), at the cost of six (6) hours of training, three (3) times the training period of the CNN+TCN architecture, which is two (2) hours. The increased complexity of this architecture does not provide a significant benefit in accuracy, which calls into question its suitability

for practical use.

The CNN+TCN+Attention architecture (R^2=0.86) slightly reduces average metrics but improves robustness on irregularly profiled OLTs: the attention mechanism helps the model focus on structurally similar historical windows rather than uniformly weighting all time steps.

CNN+TCN achieves the best raw efficiency (R^2=0.88, training time ≈ 2h): the exponentially growing receptive field of dilated convolutions naturally models daily and weekly cycles without the sequential bottleneck of RNNs. The CNN + TCN displays the best raw results (MAE = 0.032; RMSE = 0.048; R^2 = 0.88), with a record training time of two (02) hours with a relatively low computational cost.

7.3. OLT Analysis

Significant heterogeneity is observed among the OLTs: some equipment exhibits $\bar{\tau} \approx 8\%$ (consistently underloaded) while others regularly exceed 60%, reflecting differences in subscriber density and local event profiles. The OLT-based modeling strategy, resulting in a separate model for each piece of equipment, effectively captures these individual dynamics. OLTs with frequent overload events ($\tau > 70\%$) show a slightly higher prediction error, suggesting that modeling extreme and rare events remains a challenge and warrants future work on anomaly-sensitive loss functions.

7.4. Production Integration

The deployment architecture connects Elasticsearch (data storage), a Python inference module, and Power BI (visualization dashboard). A scheduled process extracts the last 7 days of data via OLT, generates 5-day load predictions, and writes the results to a dedicated Elasticsearch index. Network operations teams can then monitor the predicted load curves alongside real-time measurements, enabling proactive intervention before service degradation occurs. Figure 5 illustrates the integration architecture of the proposed model.

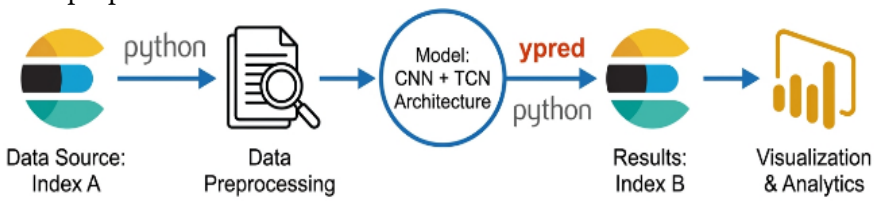

Figure 5. Model integration architecture

8. Conclusion and Outlook

This article resented the first comprehensive long-horizon prediction study of OLT load factor based on real-world operational data from a production FTTH network in West Africa. Seven (7) architectures were evaluated using approximately two (2) million real-world measurements, demonstrating that classical statistical models are unsuitable for this task and that the architecture CNN + TCN displays the best raw results (MAE = 0.032; RMSE = 0.048; R^2 = 0.88), confirming the effectiveness of dilated causal convolutions for modeling long dependencies with a relatively low computational cost.

The multi-scale TCN approach, with dilated convolutions covering dilation factors from 1 to 32, naturally captures the multi-scale cyclic structure of FTTH traffic daily and weekly cycles, evening peaks, and irregular event peaks without the sequential bottleneck of purely recurrent networks. Several directions are identified for future work: (i)study the federated learning model to account for trends by remote areas over the full horizon of 5 days with dedicated GPU resources, and evaluate non-IID (Independent and Identically Distriuted) aggregation strategies (FedProx, SCAFFOLD); (ii) integrate external contextual features such as sporting events, religious holidays (Ramadan) and rainfall that significantly influence traffic in West Africa; (iii) develop overload-sensitive loss functions that more strongly penalize prediction errors in the vicinity of critical thresholds ($\tau > 0.7$)

and (iv) broaden the scope of the study with a large number of OLTs.

Declaration on Generative AI

During the preparation of this work, the author(s) used Deepl to translate from French to English. Further, the author(s) used Gemini to make Figure 4 clearer and more readable. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] ITU-T Recommendation G.984.1, "Gigabit-capable Passive Optical Networks (GPON): General Characteristics," International Telecommunication Union, 2008.

[2] Huawei Technologies, "MA5800 Series OLT Product Documentation Overload Control (OLC) Feature Guide," Huawei, Shenzhen, 2022.

[3] G. E. P. Box, G. M. Jenkins, G. C. Reinsel et G. M. Ljung, "Time Series Analysis: Forecasting and Control," 5e éd., Wiley, 2015.

[4] S. J. Taylor et B. Letham, "Forecasting at Scale," The American Statistician, vol. 72, n° 1, pp. 37–45, 2018.

[5] S. Hochreiter et J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, n° 8, pp. 1735–1780, 1997.

[6] A. Labonne, L. Chatzinakis et A.Olivereau, "Learning to Route: Machine Learning-based Routing for SDN Networks," dans IEEE CNSM, 2020.

[7] J. Waczyńska et al., "Convolutional LSTM Models to Estimate Network Traffic," EPJ Web of Conferences, vol. 245, 2020.

[8] M. Hatem, A. R. Dhaini et S. Elbassuoni, "Deep-DBA: Deep Learning-Based Dynamic Bandwidth Allocation in NG-EPON," IEEE Access, vol. 10, pp. 42230–42241, 2022.

[9] S. Bai, J. Z. Kolter et V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv:1803.01271, 2018.

[10] A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," arXiv:1609.03499, 2016.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson et B. A. y Arcas, "Communication-EfficientLearningof Deep Networks from Decentralized Data," dans AISTATS, pp. 1273–1282, 2017.

[12] S. Niknam, H. S. Dhillon et J. H. Reed, "Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges," IEEE Communications Magazine, vol. 58, n° 6, pp. 46–51, 2020.

[13] A. Vaswani et al., "Attention Is All You Need," dans NeurIPS, pp. 5998–6008, 2017.

[14] K. He, X. Zhang, S. Ren et J. Sun, "Deep Residual Learning for Image Recognition," dans IEEE CVPR, pp. 770–778, 2016.

[15] GSMA Intelligence, "Sub-Saharan Africa Mobile Economy Report," GSMA, Londres, 2024.