Short-Term Weather Forecasting for Resource-Constrained Edge Devices

Vita Komel[2], Gregor Kosec[1], and Aleksandra Rashkovska Koceva[1]

[1] Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
{gregor.kosec,aleksandra.rashkovska}@ijs.si
https://www.ijs.si
[2] Faculty of Mathematics and Physics, University of Ljubljana, Jadranska ulica 19, 1000 Ljubljana, Slovenia
vita.komel@gmail.com

Abstract. Dynamic Thermal Rating (DTR) lets transmission system operators extract additional capacity from existing overhead lines by computing ratings from the weather around the conductor rather than from conservative static assumptions. Because the conductor heat balance is driven by weather parameters, such as ambient temperature, wind, and solar irradiance, accurate short-term weather forecast is a key enabling input, and for the very short horizons relevant to operation the most informative data come from a weather station co-located with the line. We study short-term, multivariate forecasting of local meteorological variables under the constraint that inference must run on a resource-constrained, intermittently connected edge device. Using approximately five years of five-minute measurements from a station on a Slovenian transmission line, we forecast seven weather variables up to six hours ahead and compare persistence and ARIMA baselines with three deep-learning architectures: attention-based Long Short-Term Memory (LSTM) network, Temporal Convolutional Network (TCN), and the Temporal Fusion Transformer (TFT). The TFT achieves the highest accuracy, but cannot be deployed within the target edge runtime, whereas the LSTM and TCN can, exposing a concrete accuracy-versus-deployability trade-off.

Keywords: Weather forecasting · Edge computing · Dynamic thermal rating · Temporal Fusion Transformer · Time-series forecasting.

1 Introduction

The opening of the European-wide electricity market, the increasing penetration of intermittent renewable energy sources, and growing electricity consumption have all contributed to increasingly unpredictable power flows throughout the interconnected European transmission system. This puts transmission system operators (TSOs) under constant pressure to extract more capacity from existing assets without compromising system stability, increasingly through data-driven monitoring and digitalisation at the grid edge. A prominent example is

Dynamic Thermal Rating (DTR), which replaces conservative static line ratings — computed for a fixed set of unfavourable weather conditions — with ratings derived from the actual or forecast weather around the conductor [6]. Because the conductor heat balance is governed by ambient temperature, wind speed and direction, and solar irradiance, the quality of a DTR forecast depends directly on the quality of the underlying short-term weather forecast, and for the very short horizons relevant to operation the most informative inputs are historical observations from a weather station co-located with the line [12].

This motivates the problem we address: short-term, multivariate forecasting of local meteorological variables from a single weather station, under the constraint that inference must run entirely on a resource-constrained, intermittently connected edge device deployed in the field. Such a setting is a representative instance of forecasting from continuous, non-stationary, and seasonally structured sensor data streams, and it introduces a practical tension between predictive accuracy and deployability: the most accurate models are often the least compatible with the limited compute, memory, and runtime support available at the edge.

We study this accuracy-versus-deployability trade-off on real data from a weather station on the Obersielach–Podlog transmission line, provided by the Slovenian TSO, ELES. Using approximately five years of measurements at five-minute resolution, we forecast seven meteorological variables up to six hours ahead, and we compare baseline methods (persistence and the classical ARIMA model [3]) with three deep-learning architectures: an attention-based Long Short-Term Memory (LSTM) network [5], a Temporal Convolutional Network (TCN) [1], and the Temporal Fusion Transformer (TFT) [9]. Crucially, each model is evaluated not only on forecast accuracy but on its ability to be compiled to and executed within the TensorFlow Lite runtime on a 32-bit ARMv7 device without internet access. The main contributions of the paper are:

- We frame and characterise local short-term weather forecasting as the upstream enabling task for edge-based DTR, using an operational, five-year dataset of seven meteorological variables.
- We benchmark statistical, recurrent, convolutional, and attention-based forecasters across all seven variables and the full six-hour horizon.
- We report the deployment outcome of each model on real edge hardware, exposing a concrete accuracy-versus-deployability trade-off relevant to operational grid settings.

The remainder of this paper is organised as follows. Section 2 reviews related work on deep-learning weather forecasting, weather forecasting for DTR, and inference on edge devices. Section 3 formalises the forecasting problem and describes the dataset and preprocessing pipeline. Section 4 presents the forecasting models. Section 5 reports the experimental setup, the evaluation, the results – including the edge-deployment outcome, and discusses the accuracy-versus-deployability trade-off. Section 6 concludes and outlines the importance of the work.

2 Related Work

Our work sits at the intersection of three lines of research: deep learning for short-term weather forecasting, weather forecasting as an input to DTR, and the deployment of forecasting models on resource-constrained edge hardware.

2.1 Deep Learning for Weather Forecasting

Forecasting meteorological variables from historical observations is a classical time-series problem and a long-standing topic in data mining. Statistical models of the autoregressive integrated moving average (ARIMA) family remain widely used baselines, capturing linear temporal structure and seasonality with few parameters and modest computational cost [3]. With the availability of larger datasets, deep neural networks have come to dominate the field, owing to their ability to learn nonlinear temporal dependencies automatically [7]. Recurrent architectures, and in particular Long Short-Term Memory (LSTM) networks, became standard tools for sequence modelling [5], while Temporal Convolutional Networks (TCNs) showed that causal, dilated convolutions can match or exceed recurrent models on many sequence tasks while remaining fully parallelisable [1]. More recently, attention-based architectures have advanced multi-horizon forecasting; the Temporal Fusion Transformer (TFT) combines recurrent processing of local dynamics with attention over longer contexts and supports multi-variable, multi-horizon prediction with heterogeneous inputs [9]. Comparative studies confirm that no single architecture dominates across all tasks, and that the most suitable model depends on the forecasting horizon, the data characteristics, and the available resources [7]. Our study follows this comparative tradition, but evaluates the candidate models not only on accuracy across seven variables and a six-hour horizon, but also on whether they can actually be deployed at the edge.

2.2 Weather Forecasting for Dynamic Thermal Rating

The practical motivation for our forecasting task comes from Dynamic Thermal Rating (DTR), which replaces conservative static line ratings with ratings computed from the actual or forecast weather around the conductor, thereby unlocking additional transmission capacity from existing infrastructure [6]. The conductor heat balance is governed by ambient temperature, wind speed and direction, and solar irradiance, so the quality of a rating forecast is largely determined by the quality of the underlying weather forecast, with local observations being the primary input for the very short horizons relevant to operation [12]. The sensitivity of the rating to local conditions is especially pronounced in the Slovenian network, where measurements have shown that low wind speeds frequently limit convective cooling and thus dominate the available rating [11]. This body of work establishes *why* accurate, local, short-term weather forecasts matter; in contrast to studies that forecast the rating itself, we focus on the upstream weather-forecasting service and on its feasibility under edge-deployment constraints.

2.3 Inference on Resource-Constrained Edge Devices

Running neural networks directly on embedded and IoT hardware — the field of TinyML — reduces communication dependence and latency and improves resilience and data privacy, but is constrained by limited memory, compute, and runtime support [10]. Deployment therefore relies on model-compression and conversion techniques such as quantisation, together with specialised inference runtimes; TensorFlow Lite and its microcontroller variant are the de facto standard for executing converted models on 32-bit ARM-class devices [4]. A recurring finding in this literature is that the most accurate models are not always deployable: operators used in attention and dynamic graph components are frequently unsupported by static-graph edge runtimes, forcing a trade-off between accuracy and deployability. While much TinyML work targets vision or human-activity recognition, comparatively little attention has been paid to multivariate time-series forecasting under these constraints. Our work addresses this gap directly, reporting which of the evaluated forecasting models can be converted to and executed within the TensorFlow Lite runtime on the target edge device, alongside their forecast accuracy.

3 Problem Formulation and Data

3.1 Problem Definition

We address the problem of short-term, multi-variate weather forecasting at a single meteorological station, subject to the constraint that inference must execute entirely on a resource-constrained edge device without internet connectivity. Models are trained on GPUs and subsequently compressed and deployed to the edge device.

Let f_θ denote a forecasting model with parameters θ. At each inference step, the model receives an input window of the $L = 96$ most recent observations (24 hours at 15-minute resolution) and produces a sequence of $H = 24$ predictions covering the next 6 hours:

$$\hat{Y}^{(i)} = f_\theta\left(X^{(i)}\right) \in \mathbb{R}^{H \times d_{\text{out}}}$$

where $X^{(i)} \in \mathbb{R}^{L \times d_{\text{in}}}$ is the i-th input window and d_{in}, d_{out} denote the number of input features and output variables respectively.

3.2 Data Description

Data for this study were provided by ELES, the Slovenian transmission system operator (TSO). The dataset includes historical weather measurements from a station on the Obersielach–Podlog transmission line and past forecasts for the specific location of the station. Forecasts were obtained from a micro-scale numerical weather model (NWP) utilized by the TSO, which is downscaled from the Slovenian Environment Agency's (ARSO) regional ALADIN model.

Measurements span 5 November 2020 to 30 September 2025 at 5-minute resolution, yielding approximately 515,000 raw observations. NWP forecasts are available from 30 September 2020 to 30 September 2025 at 15-minute resolution for horizons of 0 to 72 hours ahead. NWP forecasts serve exclusively as a benchmark for model comparison; they are not used as model inputs.

Seven meteorological variables were used as model inputs: ambient temperature (T, in °C), rain intensity (i, in mm/h), relative humidity (RH, in %), solar irradiance (S, in W/m^2), wind direction (ϕ, in °), wind speed (v, in m/s), and air pressure (p, in hPa). Descriptive statistics computed on the raw 5-minute data are shown in Table 1. The negative minimum of solar irradiance is not a sensor error; at night, pyranometers can produce small negative measurements known as the Type A zero offset [2]. Since these values carry physically meaningful information, they were retained rather than being clipped to zero during data preprocessing.

Table 1. Descriptive statistics of the raw meteorological measurements. Longest gap is in 5-minute steps, so the gap of 1,202 steps is approximately 4 days.

Variable	Mean	Std	Min	Max	Missing (%)	Longest gap
Temperature (°C)	10.92	8.59	-11.00	35.40	2.64	1202
Rain intensity (mm/h)	0.18	1.29	0.00	106.70	2.64	1202
Relative humidity (%)	73.67	18.10	12.20	95.10	2.64	1202
Solar irradiance (W/m^2)	137.12	239.29	-5.50	1311.00	1.23	751
Wind direction (°)	147.70	91.56	1.00	360.00	2.64	1202
Wind speed (m/s)	0.64	0.58	0.00	9.18	2.64	1202
Air pressure (hPa)	975.06	7.51	926.40	998.30	2.64	1202

3.3 Preprocessing and Feature Engineering

To prepare the dataset for modeling, the 5-minute weather station measurements were downsampled to a 15-minute resolution using the mean value for each interval. The dataset was then split into training, validation, and test sets in a 70:15:15 ratio, strictly preserving temporal order.

The raw dataset natively included diagnostic status flags monitoring sensor errors or maintenance work; all affected observations were systematically removed. Secondary checks were subsequently executed on the remaining series to detect hidden outliers, specifically checking for stuck values and unphysical step-difference spikes. These checks confirmed that the flag-filtered dataset was already clean and fell entirely within the valid meteorological boundaries, as shown in Table 1, requiring no further data removal.

For missing data, gaps of up to one hour (four or fewer consecutive 15-minute steps) were filled using forward-fill imputation. If an input lookback window still contained missing values after this step, the entire sample window was discarded from the training, validation, and test sets. Crucially, no imputation was applied to the target sequence (the future horizon); instead, a masked loss function and masked evaluation metrics were utilized to compute loss and accuracy exclusively on observed ground-truth values. During edge deployment, input windows with unresolvable missing data are flagged and no prediction is issued.

To improve models' performance, features were engineered and transformed:

– Wind direction & time: Both wind direction and cyclical time components — day-of-year (DOY) and time-of-day (TOD) — were encoded into sine and cosine pairs to preserve their continuous, cyclical nature.
– Feature scaling: Input variables, excluding the cyclically transformed wind direction and temporal features, were normalized using standard scaling (Z-score normalization).

The processed data were structured into a sliding window format, using a lookback window of the past 24 hours as input to forecast the weather parameters over a target horizon of 6 hours.

4 Forecasting Models

Several forecasting approaches were evaluated in this study, ranging from simple baseline methods to deep learning architectures. The selected models represent different paradigms for time-series forecasting, allowing comparison between statistical, recurrent, convolutional, and attention-based approaches.

4.1 Baselines

Two baseline approaches were used for comparison: persistence forecasting and ARIMA. Persistence assumes that future conditions remain equal to the most recent observation. Although simple, it provides a strong benchmark for short-term weather forecasting.

ARIMA (Autoregressive Integrated Moving Average) is a classical statistical forecasting model that combines autoregressive and moving-average components while accounting for non-stationarity through differencing [3]. Separate ARIMA models were trained for each target variable. Seasonal information was incorporated through exogenous cyclical regressors representing time-of-day and day-of-year patterns. Model orders were initially selected using auto-ARIMA and subsequently refined through grid-search optimization.

4.2 Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network architecture specifically designed to model long-term temporal dependencies through gated memory cells that regulate the flow of information across time steps [5].

A sequence-to-sequence architecture with attention was used for multi-horizon forecasting, consisting of an LSTM encoder that processes the input sequence and encodes it into a latent representation via its final hidden state. Instead of autoregressive decoding, the decoder is initialized with this final encoder state, which is repeated across all prediction steps to form a fixed-length decoding input, enabling fully parallel prediction over the forecast horizon. A scaled dot-product cross-attention mechanism is then applied between decoder outputs and encoder hidden states to compute context vectors that capture the most relevant temporal dependencies in the input sequence, and the final forecasts for all meteorological variables are produced simultaneously through a linear projection applied at each prediction step.

4.3 Temporal Convolutional Network (TCN)

The Temporal Convolutional Network (TCN) is a convolution-based architecture for sequential data. It uses causal convolutions to ensure that predictions depend only on past observations and dilated convolutions to efficiently capture long-range temporal dependencies [8].

The TCN employed in this work consists of stacked residual blocks with exponentially increasing dilation factors. Unlike recurrent models, TCNs process entire sequences in parallel, resulting in efficient training and inference. This property makes them well suited for deployment on resource-constrained edge devices.

4.4 Temporal Fusion Transformer (TFT)

The Temporal Fusion Transformer (TFT) is an attention-based model specifically optimized for multi-horizon time-series forecasting. It handles heterogeneous inputs, including static covariates, past observed features, and known future inputs. It combines recurrent layers for local temporal patterns with a multi-head attention mechanism that captures longer-term dependencies.

The architecture consists of two more core components: a gating mechanism, which gives the model the flexibility to skip redundant layers, and a variable selection network, which assesses the relevance of input variables at any given time step. The architecture also provides a degree of interpretability through variable importance scores and attention weights [9].

5 Experiments and Results

5.1 Experimental Setup

All deep learning architectures were implemented in PyTorch (or PyTorch Forecasting) and trained on a NVIDIA GeForce RTX 3090 GPU. The neural network models were optimized using the AdamW optimizer with an initial learning rate of 10^{-3}, a cosine annealing learning-rate scheduler, and an early stopping mechanism based on validation loss. Training was performed using the masked Huber loss function to account for missing target values. A batch size of 16 was used for all training runs. Hyperparameter tuning and model selection were performed on the validation set; the selected configurations are reported in Table 2.

Table 2. Architectural hyperparameters of evaluated models.

Model	Configuration
LSTM	hidden=32, non-autoregressive decoder, scaled dot-product attention
TCN	filters=32, layers=4, kernel=3, dilation=exp
TFT	hidden=32, heads=4

5.2 Evaluation Metrics

Model performance was evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). For wind direction, standard error metrics are not directly applicable due to the circular nature of angular variables; therefore, specialized metrics denoted as $\mathrm{MAE}_{\mathrm{circ}}$ and $\mathrm{RMSE}_{\mathrm{circ}}$ were used. The absolute angular difference $\Delta\theta_i$ accounts for the 360° periodicity by computing the shortest angular distance: $\Delta_i = \min\left(|y_i - \hat{y}_i|,\ 360^\circ - |y_i - \hat{y}_i|\right)$. Using this angular adjustment, the circular metrics are defined as:

$$\mathrm{MAE}_{\mathrm{circ}} = \frac{1}{N}\sum_{i=1}^{N}\Delta_i, \qquad \mathrm{RMSE}_{\mathrm{circ}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\Delta_i^2}.$$

5.3 Forecast Accuracy

The evaluation was performed on the held-out test set, including NWP forecasts for comparison. A masked error formulation was used to properly handle missing values. Results aggregated over the full forecast horizon are reported in Tables 3 and 4.

Table 3 presents results for temperature, wind speed, and wind direction — the three variables that most directly affect DTR calculations. ARIMA provides a strong statistical baseline for temperature, yielding lower errors than both persistence and NWP. However, it is consistently surpassed by all deep learning models, with the TCN achieving the lowest MAE of 1.00 °C. Wind speed and wind direction are more difficult to model across all approaches due to their highly stochastic behavior. Although the absolute errors in wind speed appear small, they become more pronounced in relative terms due to the low mean wind speed at the site (0.64 m/s, Table 1), where even minor deviations translate into large percentage errors.

Table 4 reports results for rain intensity, relative humidity, air pressure, and solar irradiance. All models show similarly low errors for rain intensity. However, this reflects limited predictive signal rather than strong model performance, as precipitation events are sparse and most observations correspond to zero or near-zero rainfall. Learned models roughly halve persistence error on humidity and solar irradiance. The most pronounced result is air pressure, where NWP produces an MAE of 22.39 hPa — likely due to an altitude reference mismatch during downscaling, which causes it to consistently underestimate pressure with a mean forecast of 951.91 hPa compared to the actual station mean of 975.06 hPa (Table 1) — whereas the TCN achieves a much closer MAE of just 0.50 hPa.

Table 3. Performance comparison for temperature, wind speed and wind direction. Metrics are reported as MAE (or $\mathrm{MAE}_{\mathrm{circ}}$ for wind direction) and RMSE (or $\mathrm{RMSE}_{\mathrm{circ}}$) with their respective units. For each variable, the best result (lowest error) is shown in bold.

	Temperature (°C)		Wind speed (m/s)		Wind direction (°)	
Model	**MAE**	**RMSE**	**MAE**	**RMSE**	**$\mathrm{MAE}_{\mathrm{circ}}$**	**$\mathrm{RMSE}_{\mathrm{circ}}$**
Persistence	2.46	3.66	0.40	0.59	64.26	82.81
ARIMA	1.48	2.09	0.31	0.44	56.45	71.49
LSTM	1.08	1.50	**0.28**	**0.42**	49.39	63.02
TCN	**1.00**	**1.45**	**0.28**	**0.42**	**49.30**	**62.87**
TFT	1.18	1.61	0.29	**0.42**	50.15	63.50
NWP	1.97	2.51	0.65	0.85	83.76	99.83

Table 4. Performance comparison for rain intensity, relative humidity, pressure and solar irradiance. Metrics are reported as MAE and RMSE with their respective physical units. For each variable, the best result (lowest error) is shown in bold.

	Rain (mm/h)		Humidity (%)		Pressure (hPa)		Solar (W/m^2)	
Model	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Persistence	0.27	1.74	9.56	14.65	0.78	1.09	145.36	255.28
ARIMA	**0.19**	1.32	6.55	9.05	0.70	1.01	117.86	193.67
LSTM	0.22	1.31	**4.50**	**6.79**	0.71	0.99	**50.23**	102.92
TCN	0.22	**1.30**	4.49	6.81	**0.50**	**0.72**	51.78	**101.45**
TFT	0.22	**1.30**	4.57	6.83	0.80	1.13	50.47	102.73
NWP	0.22	1.32	12.87	16.16	22.39	23.23	57.96	110.23

Edge-optimized variants of the LSTM and TCN models were also evaluated. No measurable difference in forecasting accuracy was observed between the standard and edge-optimized versions. Both achieved essentially identical prediction errors, indicating that the optimizations required for embedded deployment did not degrade forecasting performance.

5.4 Forecast Horizon Behavior

Forecast error accumulation over the 6-hour prediction horizon is illustrated in Figure 1. For smooth variables such as temperature, relative humidity, and solar irradiance, the ARIMA baseline exhibits rapid degradation in accuracy with increasing horizon time, whereas deep learning models maintain a more stable error progression. In contrast, highly variable quantities such as wind speed and rain intensity exhibit consistently high errors across the entire forecast horizon.

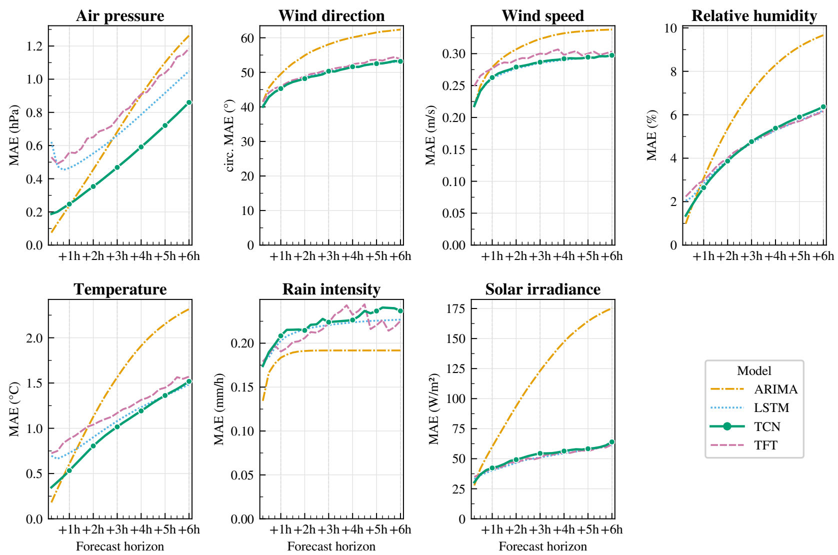

Fig. 1. Forecast error (MAE or MAE_{circ}) across the 6-hour prediction horizon for the seven target variables on the test set. Each sub-panel illustrates how predictive performance degrades over time (at 15-minute intervals) for ARIMA, LSTM, TCN and TFT.

5.5 Edge Deployment

The models are intended for operational deployment on a resource-constrained edge device. The target inference device is a 32-bit ARMv7 edge device without internet access and with limited available RAM. These constraints rule out ONNX Runtime, for which no official ARMv7l wheel exists; inference therefore uses the `tflite-runtime` package.

The TFT could not be successfully converted to TensorFlow Lite due to unsupported operations in the attention and variable-selection components, which are incompatible with the static computation graph requirements of the TFLite runtime. In contrast, the LSTM and TCN architectures were successfully exported and executed on the target device. For these models, the trained PyTorch implementations were first exported to the ONNX format and subsequently converted to TensorFlow Lite. Both models rely on statically defined computation graphs, which enables compatibility with the `tflite_runtime` inference engine. Table 5 reports the number of trainable parameters as well as the corresponding model sizes after conversion to TFLite format.

Table 5. Model complexity and deployment size after conversion to TFLite.

Model	No. parameters	TFLite size
LSTM	14,857	93.0 KB
TCN	17,504	79.2 KB
TFT	69,760	–

5.6 Discussion

The TCN provides the best overall balance of accuracy and efficiency, achieving top performance on key DTR variables such as temperature (1.00 °C MAE). Its convolutional structure enables a compact 79.2 KB TensorFlow Lite model (Table 5). The LSTM attains comparable accuracy in some cases but could have higher inference overhead due to sequential processing. The TFT does not improve predictive performance over simpler architectures and cannot be deployed on the target hardware. This indicates that transformer-level complexity is not justified for this local forecasting task.

Edge models substantially outperform NWP across all variables in the short-term forecasting setting, particularly for air pressure, where NWP exhibits strong systematic error. This reflects the limitations of coarse-resolution regional models at local, short-horizon scales. However, NWP models are generally designed for larger spatial scales and longer forecast horizons, where they may provide more reliable performance. In contrast, sensor-trained edge models better capture site-specific dynamics and are therefore more suitable for short-term operational forecasting.

6 Conclusion

We studied short-term, multivariate weather forecasting as the upstream enabling task for edge-based Dynamic Thermal Rating, under the constraint that inference must run on a resource-constrained device. Using approximately five years of measurements from a weather station on a Slovenian transmission line, we forecast seven meteorological variables up to six hours ahead and compared persistence and ARIMA baselines with LSTM, TCN, and TFT architectures, evaluating each on both accuracy and deployability on a 32-bit ARMv7 device.

The results expose a clear accuracy-versus-deployability trade-off: all deep-learning models outperform the baselines and the operational NWP forecast, but the most accurate model, the TFT, cannot be converted to the target edge runtime. The TCN offers the best balance, matching or exceeding the other models on the variables most relevant to DTR while compiling to a compact, deployable model; for this task, transformer-level complexity is not justified. Coupling the edge forecasts to the DTR computation and quantifying the resulting gain in usable line capacity would directly demonstrate the operational value of local, on-device weather forecasting.

Acknowledgments. This work was supported by the HEDGE-IoT project (European Commission, Grant No. 101136216) and the Slovenian AI Factory (SLAIF), funded by the European High Performance Computing Joint Undertaking (Grant No. 101254461) and the Ministry of Education, Science and Youth of Republic of Slovenia. The weather data were provided by ELES, the Slovenian transmission system operator.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018). https://doi.org/10.48550/arXiv.1803.01271
2. Bian, Z., Wei, C., Ding, L., Yang, W.: Analysis and research on quality control method of global radiation observation data. The Journal of Engineering **2019** (10 2019). https://doi.org/10.1049/joe.2018.9161
3. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. John Wiley & Sons, Hoboken, NJ, 5 edn. (2015)
4. David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., Warden, P., Rhodes, R.: TensorFlow Lite Micro: Embedded machine learning for TinyML systems. In: Proceedings of Machine Learning and Systems (MLSys). vol. 3, pp. 800–811 (2021)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
6. Karimi, S., Musilek, P., Knight, A.M.: Dynamic thermal rating of transmission lines: A review. Renewable and Sustainable Energy Reviews **91**, 600–612 (2018). https://doi.org/10.1016/j.rser.2018.04.001
7. Lara-Benítez, P., Carranza-García, M., Riquelme, J.C.: An experimental review on deep learning architectures for time series forecasting. International Journal of Neural Systems **31**(3), 2130001 (2021). https://doi.org/10.1142/S0129065721300011
8. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 156–165 (2017)
9. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021). https://doi.org/10.1016/j.ijforecast.2021.03.012
10. Lin, J., Zhu, L., Chen, W.M., Wang, W.C., Han, S.: Tiny machine learning: Progress and futures. IEEE Circuits and Systems Magazine **23**(3), 8–34 (2023). https://doi.org/10.1109/MCAS.2023.3302182
11. Maksić, M., Djurica, V., Souvent, A., Slak, J., Depolli, M., Kosec, G.: Cooling of overhead power lines due to the natural convection. International Journal of Electrical Power & Energy Systems **113**, 333–343 (2019). https://doi.org/10.1016/j.ijepes.2019.05.005
12. Michiorri, A., Nguyen, H.M., Alessandrini, S., Bremnes, J.B., Dierer, S., Ferrero, E., Nygaard, B.E., Pinson, P., Thomaidis, N., Uski, S.: Forecasting for dynamic line rating. Renewable and Sustainable Energy Reviews **52**, 1713–1730 (2015). https://doi.org/10.1016/j.rser.2015.07.134