InStaFlex
Report D4.3 – Load Forecasting

Submitted to: FOD Economy
Lead Partner: University of Antwerp

Instaflex Project – Results Report

1. Project Information

Project Title	InStaFlex
Client	FOD Economy
Lead Partner Report	University of Antwerp
Partner Organisations	University of Antwerp, Royal Meteorological Institute, Oktow, PropheSea

2. Document Version History

Version	Date	Author(s)	Description / Changes
0	27/05/2025	Stijn Van Raemdonck	General outline, objectives, and methods
1	14/07/2025	Stijn Van Raemdonck	Results from the Ensemble

2	30/07/2025	Stijn Van Raemdonck	Executive summary + document sent to consortium for comments

3. Executive Summary

This report lists the constraints that the load forecasting method has to adhere to, gives a concise summary of basic load forecasting concepts and the most widely used methods, details the developed methods included in the comparison, and compares the performance of the different methods. In addition, this report lists the main findings and provides recommendations for future load forecasting projects.

In short, our results highlight the surprisingly strong performance of the decision tree in our first case study, which we expect not to be transferable to other sites. Furthermore, we developed a novel method called Diff-Ensemble, which proved superior compared to the other methods. Diff-Ensemble is well-suited for high-performance load forecasting applications, but might be less suited for smaller projects given the significant implementation effort.

Moreover, a scientific publication of this work has been submitted to the DDIS workshop at the 2025 3PGCIC conference.

4. Objectives and Scope

Deliverable 4.3 aims to provide a concise overview of the basic concepts, the existing methods, and the methods used during the InStaFlex project, to forecast load (i.e., net consumption) at industrial sites. These load forecasts are necessary to anticipate the net consumption during day-ahead scheduling.

The deliverable (i.e., the load forecasting method) should adhere to the following technical constraints:

1) Lightweight: The computational complexity of the algorithms should be low enough to allow them to run on local computing hardware, which is often limited in its computing capabilities compared to modern laptops and servers, which contain advanced CPUs and GPUs. As the upper limit, the load forecast calculation for the next day can take up to 15m (i.e., the time between sensor readings). When considering local hardware that is about 10 times slower than a modern laptop, the algorithm's execution time should not exceed 90 seconds on a modern laptop. This computation time limit is set to enhance the transferability of results from this research to industry, but is less relevant during the InStaFlex project, given the use of cloud computing resources.
2) Accurate: Evidently, the accuracy of the load forecasts should be maximized to minimize the amount of electricity that has to be bought at real-time prices, thus

reducing financial risks. Moreover, the trade-off between accuracy and computational complexity should be optimized to maximize accuracy without exceeding the computation time constraints. As a minimum requirement, the method should outperform our baseline (i.e., the decision tree that uses the average for that time of the day and day of the week based on historical data as a forecast) given only a few months of historical data.

3) Able to cope with sparse data: Given the limited amount of available data for the industrial sites, the selected methodology and algorithm should reliably outperform the baseline method.
4) Robust: Learning-based methods often lack robustness in low-data environments. Therefore, the developed methodology should be robust to the inputs; that is, the load forecasts should not demonstrate erratic behavior on data points that were not present in the historical dataset.

5. Developed Methods

The task of load forecasting can be divided into three subtasks:

- Renewable energy production forecasting
- Non-controllable load forecasting
- Controllable load forecasting

This report focuses mainly on the first two subtasks, as the third subtask will be performed by the Reinforcement Learning agent.

Load forecasting methodologies can be broken down into the following categories based on their forecasting horizon:

Table 1: Classification of load forecasting methodologies based on the forecasting horizon

Abbreviation	Full name	Forecasting horizon
VSTLF	Very short-term load forecasting	minutes
STLF	Short-term load forecasting	hours
MTLF	Medium-term load forecasting	days up to months
LTLF	Long-term load forecasting	above one year

Since our goal is to provide load forecasts for day-ahead scheduling, only STLF methods will be considered in this report. Multiple methods were considered, of which a subset was implemented and compared against a baseline.

5.1 Traditional methods

Traditional load forecasting is often based on (simple) statistical methods, including linear regression, decision trees, and (S)ARIMA(X).

For this deliverable, we implemented two traditional/statistical methods:

Decision tree (baseline): The decision tree approach will be used as the baseline since it does not take into account the complex relationship between the load forecasts and the historical data. This method is based on a decision tree using time-based branching as visualized in Figure 1. For a given moment in time, the decision tree is followed up to a leaf node, which retrieves the average load value of the next timestep over all training data (i.e., for a non-holiday Monday at 0:15, the decision tree will use the average over all Mondays at 0:15).

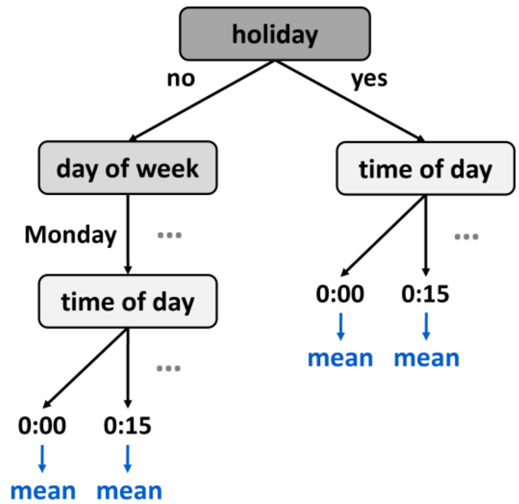

Figure 1: Decision tree (baseline) method using time-based branching

ARIMAX: ARIMAX (autoregressive integrated moving average with exogenous input) is a statistical method derived from ARIMA. ARIMAX computes a weighted sum of previous load values and the errors on those load values, whilst also taking into account exogenous variables (represented by the 'X' in ARIMAX). An advantage of ARIMAX is that it can provide confidence intervals along with the mean prediction, making it a probabilistic method. Although an ARIMAX implementation was implemented during this project, the results were highly inaccurate during weekends since ARIMAX has no way to incorporate time information. Therefore, ARIMAX was not included in the final performance comparison.

5.2 Learning-based methods

Learning-based methods, also referred to as AI-based or data-driven methods, learn the underlying patterns in the data by fitting a universal approximator (e.g., a fully-connected neural network) on the training data using an optimization algorithm, often gradient descent. In the context of STLF, methods such as ANNs (artificial neural networks) and RNNs (recurrent neural networks) have been widely applied. However, state-of-the-art STLF applications are incorporating more complex methods such as Transformers and Diffusion models. Given the limited amount of data, we did not implement the Transformer, although we do include a diffusion model with a U-Net-based denoising network.

The following learning-based methods were implemented:

LSTM: Multiple methods based on the LSTM (long short-term memory) architecture were implemented during our comparison, each one using different input and output features. An LSTM is a specific implementation of an RNN that has an internal memory state, which is controlled with multiple gates, allowing the architecture to learn both long and short-term dependencies in the time series data. The overarching LSTM architecture used for this project is illustrated in Figure 2, where X represents a sequential input (e.g., historical load or time information), Y_t is the vector of exogenous variables (e.g., outside air temperature forecasts, solar irradiance forecasts, and time information) at time t, h_t is the hidden state of the LSTM at time t, and MLP is a fully-connected neural network. Three MLP architectures were evaluated: a simple MLP with one linear layer, an MLP with two linear layers using the ReLU non-linear activation function, and an MLP with three linear layers using the ReLU non-linear activation function and dropout to reduce overfitting.

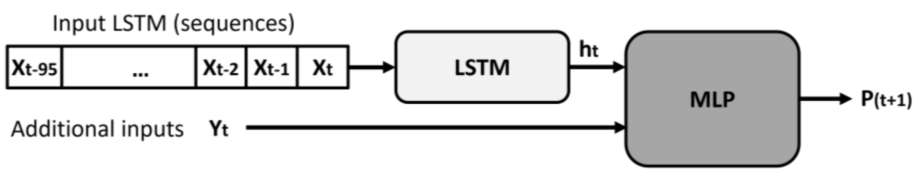

Figure 2: Architecture of the LSTM-based method included in the comparison

Diffusion model: A denoising diffusion probabilistic model (DDPM) was developed using a denoising network based on the U-Net from TimeGrad (Rasul et al., 2021, [Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting]). Similar to the LSTM-based implementation, the DDPM was evaluated using multiple input and output features and hyperparameter configurations. Contrary to the LSTM models, the DDPM diffusion models forecast the load for the next 36 hours all at once (i.e., the model outputs a sequence of length 144, representing a 36-hour forecasting horizon with 15-minute intervals).

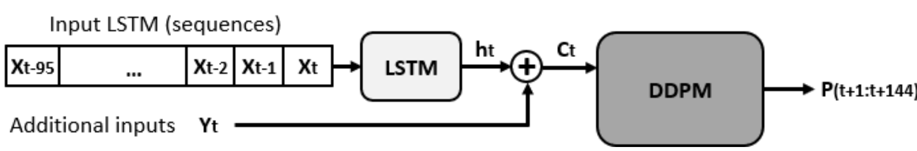

Figure 3: Architecture of the DDPM Diffusion Model included in the comparison

All learning-based methods were manually implemented in Python, a popular programming language for AI research.

5.3 Ensemble methods

Ensembles use multiple forecasting models and combine them into a single forecast. This increases the overall accuracy and robustness since individual load forecasting models are often biased. We used the following ensemble method:

> **Diff-Ensemble:** For this work package, a custom ensemble method was developed, called Diff-Ensemble. Diff-Ensemble aggregates the forecasts of the decision tree, multiple LSTMs, multiple DDPMs, and the median forecast of the individual models. This aggregation is performed by a multi-layer perceptron (MLP), which acts as the meta-learner, as visualized in Figure 4. This approach was used to improve the low-data load forecasting capabilities compared to the individual models.

Figure 4: Architecture of the ensemble model included in the comparison

6. Key Results

Experimental results on the historical consumption data from our first case study are visualized in Figure 5, which demonstrates solid performance for the decision tree, which can be explained by the high correlation between the daily load curves. We expect the decision tree to perform worse at sites with more variation in their daily consumption. Since LSTM2 does not clearly outperform LSTM1, we can conclude that weather-based features are less relevant for predicting the inflexible consumption at this industrial site. Furthermore, Diff-Ensemble achieved the best results, both for inflexible consumption forecasting and net inflexible demand forecasting, as evidenced by Figure 5a and Figure 5b.

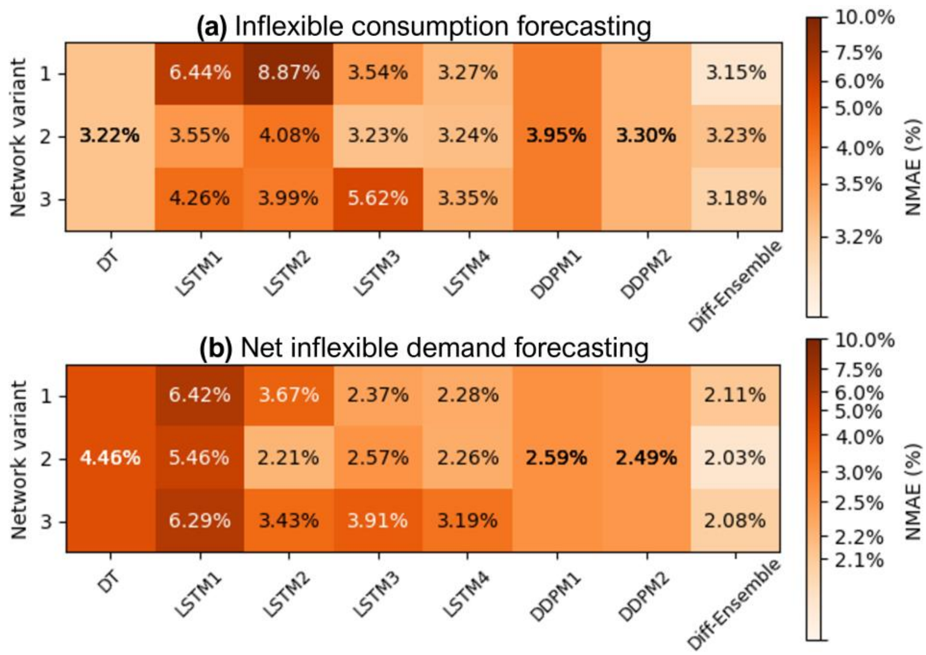

Figure 5: Comparison between the range-normalized mean absolute error (NMAE) of the decision tree, LSTM, DDPM diffusion model, and Diff-Ensemble using different input and output features and network architectures.

Figure 6 depicts the results for various training data availability levels, from four days up to eight weeks of training data. An overall decline in the forecasting error can be observed when the amount of training data increases. For inflexible consumption forecasting, both the decision tree, the ensemble, and the LSTM and DDPM with decision tree-based features (LSTM-DT and DDPM-DT) achieve similar normalized mean absolute error (NMAE) values, pointing to the high regularity in the daily load curve as mentioned above. For net inflexible demand forecasting, Diff-Ensemble demonstrates superior results compared to all other methods when the amount of training data exceeds seven days.

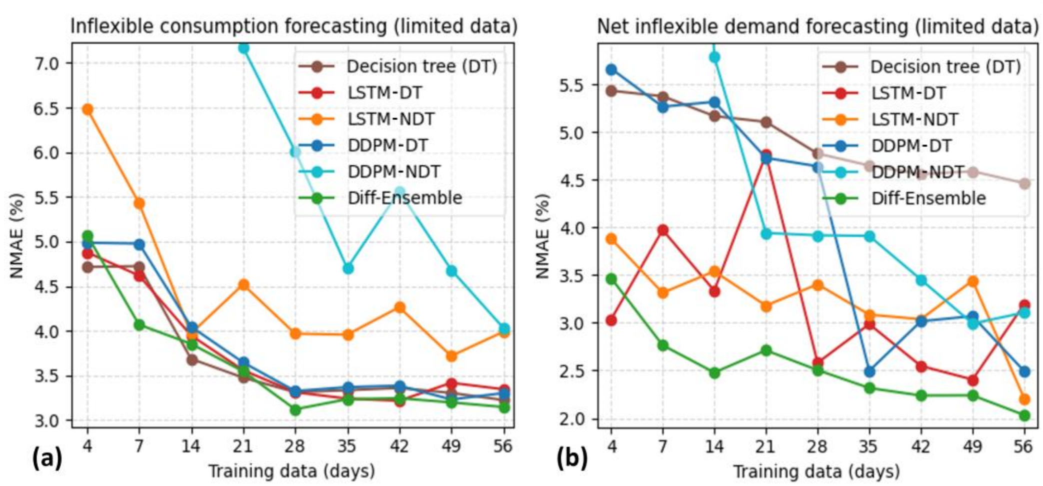

Figure 6: Performance of the different methods given different levels of training data availability

A summary of the key findings is provided below:

1) The decision tree provides surprisingly accurate forecasts when the PV generation is excluded, indicating that the consumption of the industrial site is highly repetitive with only small daily deviations.

2) Incorporating weather data does not improve the inflexible consumption forecasts of the learning-based methods, indicating that this consumption does not heavily depend on the outside temperature or solar irradiance.

3) Incorporating weather data significantly improves the net inflexible demand (i.e., inflexible consumption + PV generation) forecasts of the learning-based methods, indicating that the PV generation is highly correlated with weather forecasts.

4) Overall, Diff-Ensemble provided the lowest load forecasting errors of all learning-based methods. Furthermore, Diff-Ensemble is well-suited for load forecasting using limited data, as demonstrated in Figure 6.

5) Hyperparameter optimization has the potential to meaningfully improve the performance of the learning-based load forecasting methods.

6) All methods were able to provide load forecasts for the next day in less than 10 seconds on a modern (2024) laptop, which is well below our threshold of 90 seconds.

7) All learning-based methods proved robust to unseen data, which is likely to be attributed to the periodicity of the daily consumption and PV generation.

8. Recommendations

We recommend the adoption of Diff-Ensemble for large-scale load forecasting applications, unless resources allow for further research into other promising technologies, including Transformers and S4-based Diffusion models.

For smaller projects, such as residential applications or small businesses, the decision tree provides the most straightforward implementation and the lowest implementation cost. However, we do acknowledge that other methods not included in our comparison exist, which might be more suited when the load curve shows high daily variations.

When renewable energy generation has a considerable effect on the net electricity demand of an industrial site or building, learning-based solutions provide significant performance improvements since they can take weather forecasts into account. The decision tree and ARIMAX methods proved inferior when local renewable energy generation is present.

Finally, we recommend incorporating weather forecasts into these methods since we expect the consumption of most industrial sites or buildings to be correlated with the local weather conditions.

10. Contact and Acknowledgements

Organisation	Contact Person	Email
PropheSea	Tomas Van Oyen	tomas.vanoyen@prophesea.eu
Oktow	Brecht Zwaenepoel	brecht@oktow.be
University of Antwerp	Peter Hellinckx	peter.hellinckx@uantwerpen.be
Royal Meteorological Institute	Joris Van den Bergh	joris.vandenbergh@meteo.be