POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

PREDator: An Open-Access Deep Learning Framework for Long-Term Industrial Time Series Forecasting

Laurea Magistrale in Chemical Engineering - Ingegneria Chimica

Author: Antonio Mazzitelli

Advisor: Prof. Flavio Manenti

Co-advisor: Luis Felipe Sánchez

Academic year: 2025-2026

---

1. Introduction and Context

1.1. The Role of Predictive Modeling in Industry 4.0

The transition toward Industry 4.0 has established data-driven forecasting as a cornerstone for enhancing operational reliability and optimizing predictive maintenance strategies[4]. While mechanistic models can offer deep insight into process behavior, developing them for complex industrial systems is often prohibitively difficult. Chemical plants involve nonlinear interactions, multi-phase phenomena and equipment-specific characteristics that are challenging to represent accurately through fundamental equations alone. Data-driven methods, by contrast, leverage the large volumes of historical and real-time operational data already available, allowing predictive models to learn system behavior directly from observations, making them more practical, scalable, and adaptable.

1.2. Limitations of Current Commercial Solutions

Currently, major software vendors offer proprietary Asset Performance Management (APM) suites, such as GE Vernova and Oracle. However, while these commercial platforms offer robust capabilities for industrial deployment, they often function as "black boxes" with prohibitive licensing costs and closed architectures. Their proprietary nature restricts access to underlying algorithms and data structures, limiting their utility for open scientific research and reproducibility.

1.3. The Gap in Long-Term Forecasting Methodology

Predictive models contemplate two different time horizons: short-term forecasts are crucial for predictive control, while long-term forecasts reveal slow-evolving trends indispensable for predictive maintenance. In the current literature, many studies report predictive models achieving very high accuracy, but these often apply primarily to one-step-ahead predictions. Relying on a rolling one-step-ahead evaluation strategy introduces a form of data leakage, since each forecast exploits information from the test set that would not be available in a real deployment scenario. Traditional recursive forecasting strategies inherently suffer from the Compounding-Error Problem[1], where any small divergence from the ground truth at the

first predicted step is amplified in all following steps. This phenomenon, often termed Exposure Bias, leads to a rapid degradation of the predictive trajectory, making them inherently unsuitable for long-term forecasting.

2. Objectives of the Research

2.1. Core Objective and Methodology

This work aims to address the gap in robust long-term predictive models for complex industrial settings by proposing a generalized Deep Learning framework based on Long Short-Term Memory (LSTM) networks[3]. To explicitly avoid the compounding-error propagation typical of traditional recursive models, this research employs a direct multi-step forecasting strategy. In this configuration, the model is designed to map a high-dimensional input window directly to a high-dimensional output vector representing the entire future horizon in a single pass. This approach allows the model to capture the overall shape of the degradation curve rather than focusing on local step-to-step variations, which is essential for accurate long-term forecasting.

2.2. The PREDator Framework

A crucial objective of this research is the definition of a Python-based framework, named PREDator. Equipped with a Graphical User Interface (GUI), PREDator is designed to bridge the gap between high-level algorithmic design and practical model deployment. It enables users to load datasets, configure parameters, and execute the proposed models without requiring programming expertise. The tool automates the data-handling, training, and validation phases, transforming the training process from a manual, script-based task into a robust, reproducible, and scalable analytical pipeline.

3. Methodology and Case Studies

3.1. Industrial Case Studies

The proposed methodology was validated through two distinct industrial case studies to ensure robustness and adaptability across different sectors:

- *Batch Fermentation*: The case study involves the fed-batch fermentation process of *Saccharomyces cerevisiae* (baker's yeast) in an aerated stirred tank reactor. This biological process is characterized by nonlinear dynamics, an exothermic nature requiring active cooling, and a continuous accumulation of mass. To effectively monitor the system, the study intentionally isolates three key variables: the molasses feed rate (*Molasses Feed*), representing the continuous pumping of the primary carbon source; the total volume (*Tank Level*), which rises monotonically reflecting the integral mass balance of the fed-batch operation; and the process temperature (*Temp*), a critical environmental condition.
- *Industrial Furnace*: It involves the industrial furnace PH-401B, operating within the thermal de-asphalting section of an oil regeneration plant[2]. This continuous unit is subject to progressive coking (fouling), an aging phenomenon that degrades thermal efficiency and increases flow resistance over time. For the furnace, the predictive model leverages historical time-series data to monitor Methane Fuel Consumption (FI-4091), Tube Skin Temperature (SK-4072), and the critical prognostic variable for maintenance scheduling, the Inlet Pressure Drop (PI-4301).

3.2. Data Processing

The preprocessing pipeline applies a Simple Moving Average (SMA) and an Exponentially Weighted Moving Average (EWMA) for signal smoothing and noise reduction, followed by Min-Max scaling to map values into the [0, 1] interval. A significant challenge in the industrial furnace case study was the availability of only a single operational run. To address this data scarcity, a domain-specific data augmentation pipeline was implemented. Guided transformations were applied to generate 10 synthetic pseudo-runs, ensuring the physical consistency of the process was preserved.

3.3. Forecast Horizon Definition

To ensure consistent evaluation difficulty across heterogeneous datasets, the forecast horizon H is defined relative to the intrinsic seasonality of the data, denoted as τ. The study evaluates

models on two distinct scales: a short-term horizon ($H_{short} = 0.25\tau$) and a long-term horizon ($H_{long} = 0.5\tau$).

3.4. Baseline Models

The proposed LSTM framework is benchmarked against two commonly employed methods to quantify the performance gain provided by more complex architectures:

- *Polynomial Regression (PR):* PR serves as a deterministic baseline to characterize the global trend of the sensor data. While computationally, these models are inherently limited by their rigid functional form and struggle to adapt to stochastic variations or local temporal dependencies.
- *Gaussian Process Regression (GPR):* GPR is employed as a state-of-the-art probabilistic benchmark. It is a Bayesian nonparametric framework that limits overfitting by automatically balancing data fit and model complexity. To construct expressive covariance functions, a greedy tree-based search strategy was implemented. This approach iteratively combines base kernels (Linear, Radial Basis Function, and Rational Quadratic) to capture multi-scale fluctuations and the structure of the signal.

3.5. Proposed LSTM Architecture and Direct Multi-step Strategy

Recurrent Neural Networks (RNNs) are uniquely suited for sequential data. To overcome the vanishing gradient problem typical of standard RNNs and effectively capture long-range dependencies, the proposed framework utilizes Long Short-Term Memory (LSTM) cells.

To explicitly avoid the Exposure Bias inherent in recursive one-step-ahead models, the framework employs a direct multi-step forecasting strategy. This approach treats forecasting as a multi-output regression problem, mapping a high-dimensional input window of length L directly to an output vector representing the entire future horizon of length H in a single forward pass:

$$\hat{y}_{t+1:t+H} = f_\theta(x_{t-L+1:t})$$

The proposed architecture consists of two stacked LSTM layers designed to progressively extract and model both short- and long-term temporal interactions. Dropout regularization is incorporated to prevent the model from overfitting to local temporal fluctuations. Finally, a fully connected dense layer maps these latent temporal representations directly to the H predicted future values simultaneously, ensuring long-range consistency.

3.6. Experimental Design and Study Protocol

To rigorously evaluate the effectiveness of the proposed forecasting framework, the experimental campaign is structured into a hierarchical protocol consisting of three distinct sequential phases.

- *Phase I: Probabilistic Benchmarking:* This phase establishes a performance baseline using conventional modeling techniques on a single operational run. Polynomial Regression (PR) and Gaussian Process Regression (GPR) are employed to document the limitations of standard models, specifically their inability to capture high-frequency stochastic fluctuations and long-term non-stationarity.
- *Phase II: Assessment of Model Instability:* This phase addresses the critical issue of applying Deep Learning architectures in data-scarce environments. By training the LSTM network on a single fouling cycle of the industrial furnace, it demonstrates that deep learning models are highly susceptible to overfitting and local noise memorization when forced to learn from a single sequence.
- *Phase III: Comparative Evaluation:* The final phase evaluates the proposed model using abundant training data, 100 original runs for the batch fermentation case and 11 runs (including 10 synthetically generated ones) for the industrial furnace. The LSTM framework is systematically compared against the GPR baseline across short and long forecast horizons, using the relative Root Mean Square Error (rRMSE).

4. The PREDator Framework

4.1. Purpose and Architecture

To facilitate the experimental phase and manage the complexity of recurrent neural network

architectures, a specialized Graphical User Interface (GUI) named PREDator was developed in Python. The primary rationale behind this tool is to bridge the gap between algorithmic design and practical model deployment, providing a standardized environment for training and testing predictive models on industrial datasets. PREDator enables users to load datasets and execute advanced predictive models without requiring programming expertise, and offers an efficient solution even for experienced programmers with limited time. The architecture follows a modular approach, leveraging the Tkinter framework for a lightweight frontend and the TensorFlow and Keras libraries for the core deep learning analytical engine. To maintain application stability during computationally intensive training phases, PREDator implements an asynchronous execution strategy via multithreading, preventing the interface from freezing. Furthermore, it explicitly includes a native package for Gaussian Process Regression (GPR) to facilitate probabilistic comparative analyses.

4.2. Automation and Traceability

PREDator automates the entire data-handling and modeling pipeline. Through the interface, critical parameters such as the input rolling window (L), the forecast horizon (H), and specific network hyperparameters can be adjusted without direct modification of the underlying source code. Once the data is loaded, the tool executes automated preprocessing, scaling, and rolling window generation. A core feature of PREDator is its automated result traceability. For every execution, it autonomously generates a structured output repository acting as a comprehensive "digital snapshot" of the experiment. This includes serialized model weights (.h5 format) for future deployment, high-resolution graphical representations of loss convergence, and comprehensive .csv logs documenting performance metrics and prediction results.

5. Results and Discussion

The experimental campaign was structured into a hierarchical protocol consisting of three distinct sequential phases to progressively isolate the challenges of industrial time-series forecasting.

5.1. Phase I: Probabilistic Benchmarking (Single-Run)

The first phase established a performance baseline using conventional modeling techniques on a single operational run. Polynomial Regression (PR) models were employed to characterize global deterministic trends, but they failed to capture high-frequency stochastic fluctuations and sudden decreases caused by blowing operations. Gaussian Process Regression (GPR) demonstrated good ability to follow local trends in short horizons but suffered a progressive loss of predictive accuracy over extended horizons. The GPR tends to produce conservative predictions, failing to track critical non-linear upward trends such as the inlet pressure drop (P14301) in the furnace over long periods.

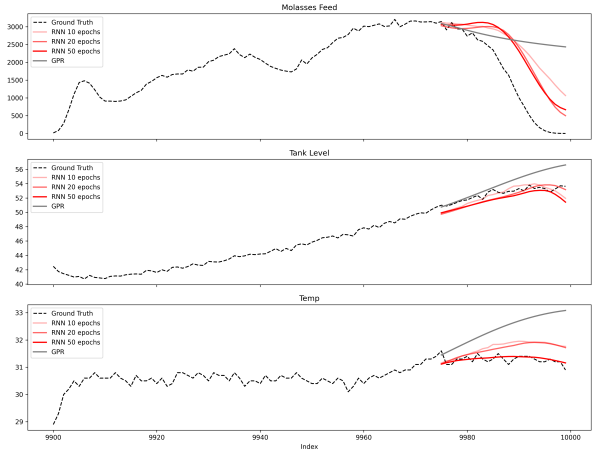
5.2. Phase II: Assessment of Model Instability (Limited Data Scenario)

The second phase addressed the critical issue of applying Deep Learning architectures in data-scarce environments, specifically focusing on the industrial furnace case with only a single historical run. The results confirmed that deep learning models are not inherently robust; when forced to learn from a single sequence, the RNN architecture exhibited severe instability, local noise memorization, and persistent forecasting bias regardless of the number of training epochs. The model failed to provide historical context to discrete interventions like steam blowing operations, causing erratic high-frequency oscillations. This "garbage in, garbage out" outcome definitively justified the necessity of richer datasets.

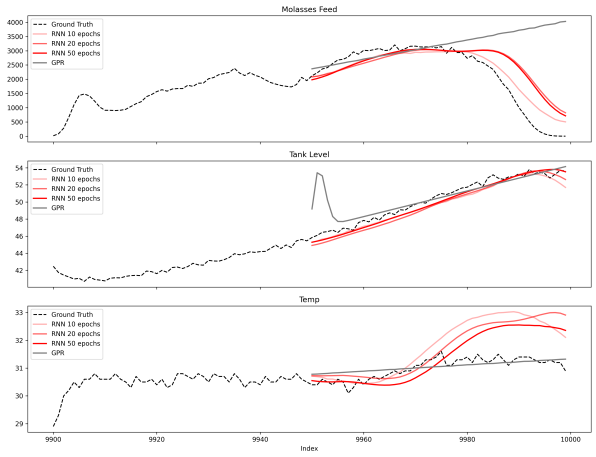
5.3. Phase III: Comparative Evaluation on Multi-Run Datasets

In the final phase, the LSTM framework was trained on abundant data (100 original runs for the batch case, and 10+1 augmented runs for the furnace) and compared against the GPR baseline. The direct multi-step LSTM architecture demonstrated superior generalization capabilities, effectively tracking the gradual drift caused by fouling without succumbing to numerical instability over the long-term horizon (H_{long}). For the industrial furnace, the LSTM successfully captured the non-linear dynamics of the P14301

pressure drop, reducing the Root Mean Square Error (RMSE) to 1.06, compared to the substantial divergence of the GPR model (RMSE 4.98). Similarly, in the batch fermentation case, the LSTM accurately anticipated rapid structural shifts, such as the sudden drop-off in Molasses Feed, achieving an RMSE of 2.10 versus the GPR's 5.48. A slight overshoot appears in the long-horizon Temperature predictions. This discrepancy is not due to the model's architecture but to the specific test run analyzed, which showed anomalous thermal behavior compared to the historical trends in the training set. A qualitative visualization of the results is shown in Figure 1,2, while a summary of the quantitative performance of the model is reported in Table 1,2.

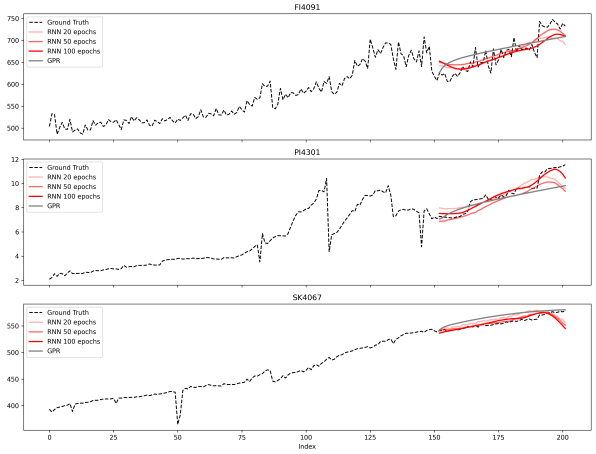


(a) Batch – H_{short}

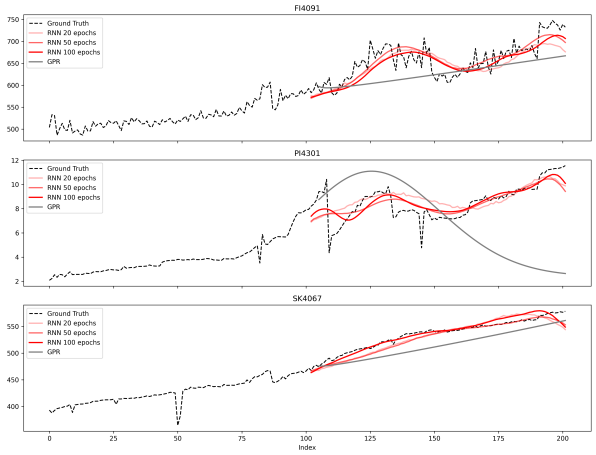


(b) Batch – H_{long}

Figure 1: Qualitative comparison between Ground Truth, GPR baseline and LSTM multi-step predictions for the batch fermentation case study.



(a) Furnace – H_{short}



(b) Furnace – H_{long}

Figure 2: Qualitative comparison between Ground Truth, GPR baseline and LSTM multi-step predictions for the industrial furnace case study

Batch Fermentation Case

	H_{short}	H_{long}
GPR	4.56/0.16/0.20	5.48/0.23/0.06
LSTM	2.63/0.10/0.02	2.10/0.09/0.18

Table 1: Relative RMSE comparison between GPR and LSTM models and forecast horizons. The errors are reported as Molasses Feed/Tank Level/Temp

Industrial Furnace Case

	H_{short}	H_{long}
GPR	0.28/0.64/0.15	3.05/4.98/0.42
LSTM	0.23/0.27/0.11	0.38/1.06/0.15

Table 2: Relative RMSE comparison between GPR and LSTM models for different forecast horizons. The errors are reported as FI4091/PI4301/SK4067.

6. Conclusions and Future Developments

6.1. Conclusions

This work successfully addressed the challenge of long-term forecasting in industrial time series by proposing a robust data-driven approach based on LSTM architectures. Unlike conventional recursive schemes that implicitly exploit future observations, the direct multi-step strategy explicitly avoided ground-truth injection, enabling a realistic assessment of model performance under deployment-like conditions. The results across both the batch fermentation and industrial furnace case studies demonstrated that the proposed framework effectively captures slow-evolving dynamics and discrete operational shifts, preserving predictive accuracy over long forecasting windows. By outperforming traditional baseline approaches, the methodology aligns with the increasing adoption of predictive analytics within Industry 4.0 frameworks, offering a reliable tool for maintenance planning and early anomaly detection.

6.2. Limitations and Future Developments

Despite these contributions, the current approach relies solely on historical observations and does not exploit physical knowledge, which could mitigate extrapolation errors in unseen regimes. Future research should focus on the integration of data-driven forecasts with physics-based knowledge, such as Physics-Informed Neural Networks (PINNs). Additionally, the inclusion of exogenous variables (e.g., setpoints or ambient factors) could further increase the predictive power and flexibility of the models. Finally, a long-term objective is the deployment of the PREDator framework within digital twin architectures, embedding predictive models into real-time virtual replicas to enable continuous performance monitoring and scenario simulation.

7. Acknowledgements

I would like to warmly thank Luis Felipe Sánchez, whose support and patience accompanied me throughout this entire journey. My gratitude also goes to Prof. Flavio Manenti, who offered me the opportunity and the means to dive into a new research topic, and provided key guidance in shaping the direction of this work. I am grateful to Prof. Mattia Vallerio as well for his guidance and constructive feedback. Finally, I would like to acknowledge the SuPER team for their quiet support, sometimes even without my noticing. To all of them, I extend my deepest esteem and respect for their contribution to this work.

References

[1] Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michel L Littman. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320*, 2019.

[2] Andrea Galeazzi, Francesco de Fusco, Kristiano Prifti, Francesco Gallo, Lorenz Biegler, and Flavio Manenti. Predicting the performance of an industrial furnace using gaussian process and linear regression: A comparison. *Computers & Chemical Engineering*, 181:108513, 2024.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Jay Lee, Behrad Bagheri, and Hung-an Kao. Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. *Procedia CIRP*, 16:3–8, 2014.