A benchmark study of supervised learning methods for predicting the live steam production of thermal power plants

G. Prokhorskii[1,2,3*], E. Eder[1,2], S. Rudra[3], M. Preißinger[1,2]

[1]illwerke vkw Endowed Professorship for Energy Efficiency, Research Center Energy, Vorarlberg University of Applied Sciences, Hochschulstraße 1, 6850 Dornbirn, Austria

[2]Josef Ressel Centre for Intelligent Thermal Energy Systems, Vorarlberg University of Applied Sciences, Hochschulstraße 1, 6850 Dornbirn, Austria

[3]University of Agder, Jon Lilletuns vei 9, 4879 Grimstad, Norway

*Corresponding author: gleb.prokhorskii@fhv.at

Abstract

Power plant operators increasingly rely on predictive models to diagnose and monitor their systems. Data-driven prediction models are generally simple and can have high precision, making them superior to physics-based or knowledge-based models, especially for complex systems like thermal power plants. However, the accuracy of data-driven predictions depends on (1) the quality of the dataset, (2) a suitable selection of sensor signals, and (3) an appropriate selection of the training period. In some instances, redundancies and irrelevant sensors may even reduce the prediction quality.
We investigate ideal configurations for predicting the live steam production of a solid fuel-burning thermal power plant in the pulp and paper industry for different modes of operation. To this end, we benchmark four machine learning algorithms on two feature sets and two training sets to predict steam production. Our results indicate that with the best possible configuration, a coefficient of determination of $R^2 = 0.95$ and a mean absolute error of $MAE = 1.2\ \mathrm{t/h}$ with an average steam production of $35.1\ \mathrm{t/h}$ is reached. On average, using a dynamic dataset for training lowers *MAE* by $32\ \%$ compared to a static dataset for training. A feature set based on expert knowledge lowers *MAE* by an additional $32\ \%$, compared to a simple feature set representing the fuel inputs. We can conclude that based on the static training set and the basic feature set, machine learning algorithms can identify long-term changes. When using a dynamic dataset the performance parameters of thermal power plants are predicted with high accuracy and allow for detecting short-term problems.

Keywords: Thermal power plant, Live steam prediction, Supervised learning, Monitoring system, Predictive maintenance

Nomenclature

Acronyms

ANN	Artificial neural network
DD	Data-driven
KB	Knowledge-based
KNN	K-nearest neighbors
ML	Machine learning
MLP	Multilayer perceptron
PB	Physics-based
PM	Predictive maintenance
RFR	Random forest regressor

Symbols

α	Regularization parameter $(-)$
$\hat{y}$	Predicted value $(\mathrm{t/h})$
$\overline{y}$	Arithmetic mean of value $(\mathrm{t/h})$
n	amount $(-)$
R^2	Coefficient of determination $(-)$
t_{calc}	Calculation Time (s)
y	Value $(\mathrm{t/h})$
MAE	Mean absolute error $(\mathrm{t/h})$

Subscripts

l	Layers
max	Maximum
min	Minimum
n	Neighbors
t	Trees

Introduction

The complexity of thermal power plants inevitably leads to downtime, which cannot be completely avoided by maintenance at predefined intervals or based on conditions. Predictive maintenance could be a solution and needs to be investigated more for thermal power plants [1]. The term ”predictive maintenance” (PM) consists of the two words ”predictive” and ”maintenance”, which implies that accurate predictions are the foundation of appropriate PM. For thermal power plants, high-accuracy predictions of performance parameters are of particular interest. For such predictions, appropriate models are necessary.
Three general methods to model systems like thermal power plants exists [2]: data-driven (DD), knowledge-based (KB), and physics-based (PB). KB predictions depend on knowledge from operational personnel, and this knowledge is usually not centralized. Shift changes and retirements often lead to the loss of information. Physical or mathematical approaches have limited validity for modeling complex systems, resulting in a high computational effort. On the other hand, DD methods based on machine learning (ML) are ideal for thermal power plant modeling. Predictions based on ML are powerful tools for the operator to detect specific issues and identify the involved equipment in this process. In literature, DD methods are the most used ones either as standalone or in combination with other methods [1].
Performance predictions are generally more meaningful when they are based on actual data. We use the data of a project partner who builds thermal power plants and is interested in a prediction model, which is efficient, accurate, and has a low computational demand. To fulfill such a task, the dataset in use should be consistent, complete, and easy to process. Performance parameters of thermal power plants can then be predicted by means of regression. Regression is commonly understood as the prediction of a continuous target value using attributes or features. It has plenty of real-world applications. However, the performance of the regression methods strongly depends on (1) the feature set, (2) the selected training period, and (3) the algorithm and its working principle.
Khalid et al. [3] developed an optimal sensor selection approach based on ML, however, it just works when normal and abnormal behavior is labeled, and feature selection becomes simple. In the case of unknown labels, expert knowledge, and feature selection must be applied based on the whole dataset. Thota and Syed [4] analyzed different data-driven feature selection methods for predicting boiler efficiency and discovered that dimensionality reduction can improve the model accuracy and that ensemble learning techniques such as random forest classifiers are more robust than other methods. Hundi and Shahsavari [5] compared several supervised and unsupervised learning algorithms such as linear regression, multilayer perceptron (MLP), support vector machine, random forest regressor (RFR), and elliptic envelope for health monitoring of power plants. Their results represent the applicability of ML for health monitoring. Gu et al. [6] investigated a safety assessment of thermal power plants based on 120 Management Systems Safety Assessment of Thermal Power plants records using different ML algorithms. Ismail et al. [7] deployed an early prediction of boiler tube leak trip using an intelligent monitoring system. In this study, two models use artificial neural networks (ANNs) and hybrid techniques to predict tube leakage based on real value prediction. Allen et al. [8] showed the difference between supervised, semi-supervised, and unsupervised ML algorithms in the field of anomaly detection. Additionally, their investigations revealed the necessity of checking training data when the model fails, checking for physical changes to the machinery and non-standard configurations. Mohd Nistah et al. [9] investigated the implementation of an ANN for fault detection to help operators to identify and narrow down the operational boiler parameters that cause the fault quickly. Although, all of these papers show the use of ML techniques for anomaly detection, the results are closely related to real-time value prediction.

However, hardly any literature exists on prediction models for thermal power plants. In a comprehensive work, Tufekci at al. [10] investigated different ML algorithms to predict the power output of two gas turbines and a steam turbine in a combined cycle power plant. The author used ambient conditions and the steam turbine vacuum to show dependencies between the features and the power output. Based on these features, the power output was predicted for the nominal load. In contrast, our project partner has large variations in the steam output, and we aim to predict the live steam for various loads with high accuracy. In the second paper to consider here, the authors created a k-nearest neighbors (KNN) prediction model on power plant data to predict the temperature and differential pressure of a coal mill [11]. Their model shows good accuracy but doesn't allow for extrapolation to unknown states.
We state that the gap in literature is generic recommendations on algorithms, feature handling, and appropriate training periods for predicting performance parameters in thermal power plants. In our research, we intend to address this lack. We benchmark several ML algorithms on actual power plant data to (1) detect and predict short- and long-term changes in the power plant operational behavior, and (2) investigate the sensitivity of our predictions to different settings with respect to (a) the hyperparameters of the algorithms, (b) the feature set, and (c) the training set.

Methods

Steam power plant description

A detailed schematic of the investigated thermal power plant is given in Fig. 1.

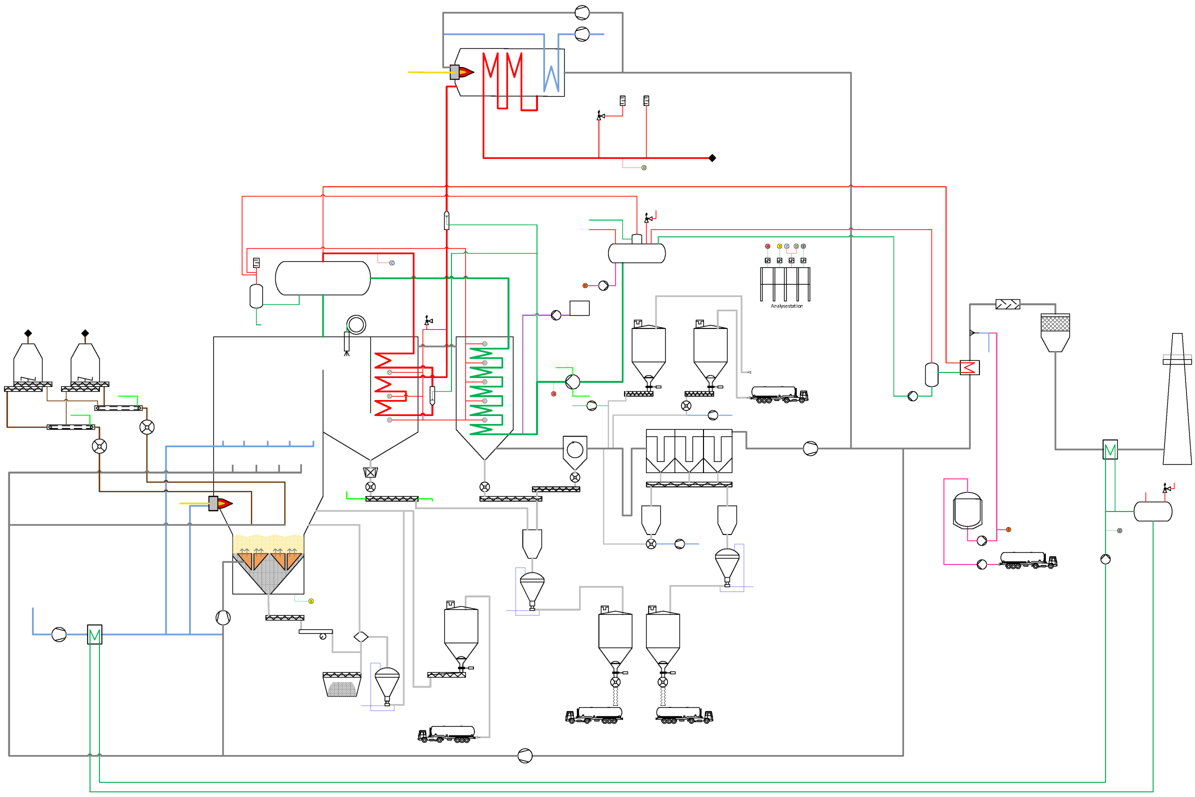

Figure 1. Scheme of the power plant with auxiliary equipment

The steam power plant investigated includes a water treatment system, a deaerator, a boiler with superheaters, and an external superheater on the water side. The fuel side contains solid fuel storages, transporters, feeders, and gas lines with gas burners. Finally, the air side includes heaters, ash removers, air purifiers, and a flue-gas stack. The boiler unit has several ash removers and collecting systems to clean the airflow from impurities. It is evident that a steam turbine is not installed as the subsequent pulp and paper production uses the live steam directly. Therefore, the thermal power plant produces live steam with a pressure of $73\ \mathrm{bar}$ and a temperature of $460\ °\mathrm{C}$.

Raw data and preprocessing

Our predictions are conducted using a period of half a year between January 1st and July 5th in 2022. The raw data samples have a resolution of $1\ \mathrm{Hz}$. In the first step, we remove redundant, unnecessary, and incomplete sensor signals, simplifying the dataset to 900 operational features from more than 3500 signals. The features are then resampled to a quarter hour resolution by aggregation or calculation of the arithmetic mean.

Prediction algorithms

Python version 3.8.5 [12] and scikit-learn package [13] version 1.1.1. are used for ML predictions. We benchmark four ML algorithms for predicting the target value of live steam flow.

- **K-nearest neighbors**: When assigning a continuous value to a new sample, the KNN algorithm compares the features of the sample and the training set. It then calculates a target value by averaging the target values of the n_n values that are in the closest neighborhood with respect to the features. The model complexity is varied by a parameter representing the number of neighbors n_n to consider.
- **Random forest regressor**: RFR is an example of an ensemble learning algorithm. It uses a group of decision trees and averages the prediction of the individual trees to predict a continuous value. The model complexity can be varied using the number of decision trees n_t.
- **Multilayer perceptron**: MLP is a representer of ANNs, which utilize a supervised learning technique called backpropagation for training. The ANN consists of at least three layers ($n_\mathrm{l} = 3$), and adding additional layers increases the model complexity.
- **Lasso regression**: Lasso is an adaptation of a simple linear regression algorithm. It uses an additional regularization parameter (α) to vary the model complexity. The feature weights are reduced for large values of α, lowering the algorithm's complexity.

Internal parameters, feature and training sets

Most ML algorithms have parameters for varying the complexity. A high complexity indicates that the algorithm tries to extract as much information as possible from the training set, to fit the target most accurately. This can lead to overfitting on the training set and to a specified solution, which doesn't generalize very well. A suitable model complexity, on the other hand, prevents overfitting of the data and leads to a more generic solution. In Tab. 1 the complexity parameters of the individual algorithms are listed with their respective minimum and maximum values. Variations of these hyperparameters have an impact on the accuracy and on the time consumption of the algorithm.

Table 1. Hyperparameters of all algorithms

Algorithm	Complexity parameter	Minimum	Maximum
KNN	n_n	1	100
RFR	n_t	1	200
MLP	n_l	1	500
Lasso	α	0.001	500

We investigate the impact of different training periods and how these periods affect individual predictions by comparing two training sets: a static and a dynamic training set (Fig. 2).

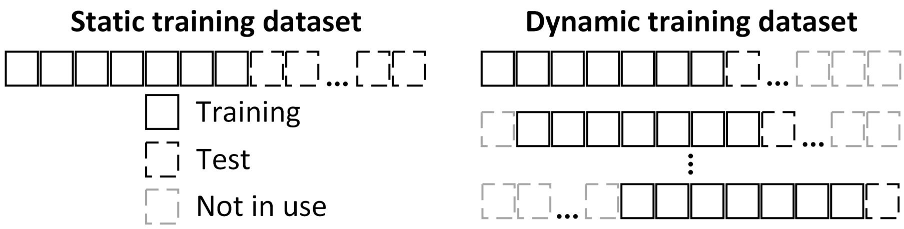

Figure 2. Different types of training sets (each square represents one day of operation)

- **Static training set:** The first seven days of the investigated period of 185 days are selected as the training period. The live steam flow is then predicted for the entire remaining dataset.
- **Dynamic training set:** To predict a certain day of the 185 days, the previous seven days are used for training the algorithm. Retraining the dataset every day is computationally extensive but allows for an adaption of the algorithm to occurring changes in the power plant operation.

The period of seven days for training the algorithm is not chosen arbitrarily. We conducted preliminary test runs and concluded that a training period between three and ten days gives the best prediction results. As a power plant has weekly patterns regarding the live steam flow we select seven days as the training period. In addition to using two training sets, the impact of the features for training the algorithm is of interest. Two different feature sets are used:

- **Basic feature set:** We use the natural gas and solid fuel flows as features and consider this the basic feature set.
- **Extended feature set:** For the extended feature set, 13 signals are selected based on expert knowledge. All the selected features are related to the target value and influence it to some extent.

Fig. 3 shows the natural gas and solid fuel flows (which comprise the basic feature set) and the target value live steam flow for an exemplary period. The left y-axis represents the fuel flows, while the right y-axis shows the live steam flow.

According to Fig. 3 the boiler works either on gas or solid fuel and sometimes on both to sustain the burning stability. It is also evident that solid fuel usage results in a higher live steam flow. When solid fuel only powers the boiler, the live steam flow is not constant and hasn't reached its maximum.

Fig. 4 represents the correlations between the 13 signals that are chosen based on expert knowledge and the target value. Trivial features like the feed water flow are not considered for the feature selection process. While most of the selected signals are highly correlated, some don't correlate at all. We still included these low-correlated signals in the extended feature set. The reason for this is that while they are not beneficial to the accuracy of the prediction, they are indicators of typical anomalies in thermal power plants. For example, the level in the steam drum is usually constant but will decrease for tube leakages, resulting in a deviation between the prediction and the actual values.

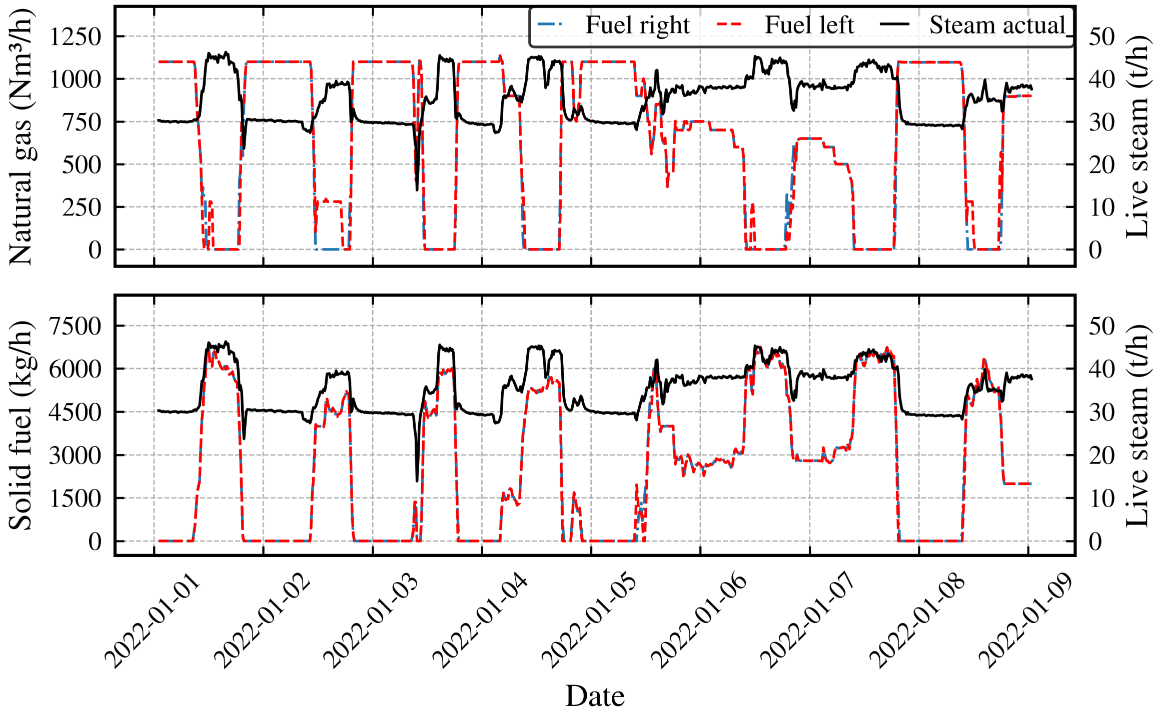

Figure 3. Basic feature set containing natural gas (top) and solid fuel (bottom), and the target value live steam flow with respect to time for nine days

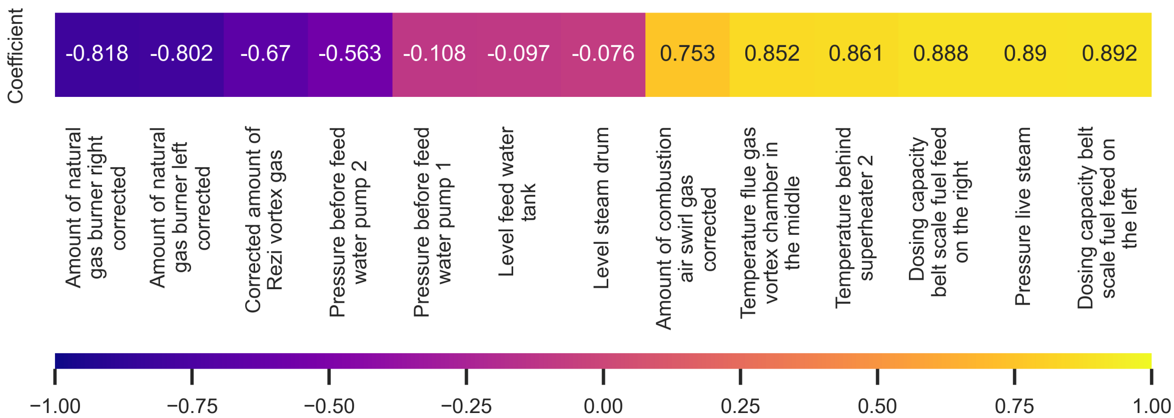

Figure 4. Correlation between selected sensor signals and the target value live steam flow

Flow scheme and quality estimators

Fig. 5 depicts a summary of the conducted methods in a flow chart. The raw dataset consisting of 3518 signals is first pre-processed, then resampled to a quarter hour resolution and finally used for various predictions of the live steam flow. KNN, RFR, MLP and lasso regression are used as ML algorithms. For each of the tested algorithms, the prediction is computed multiple times to find optimized values for the respective hyperparameters. This is done for two different feature sets (the basic feature set consisting of four features and the extended feature set consisting of 13 features) and two different training sets (a static and a dynamic training set).

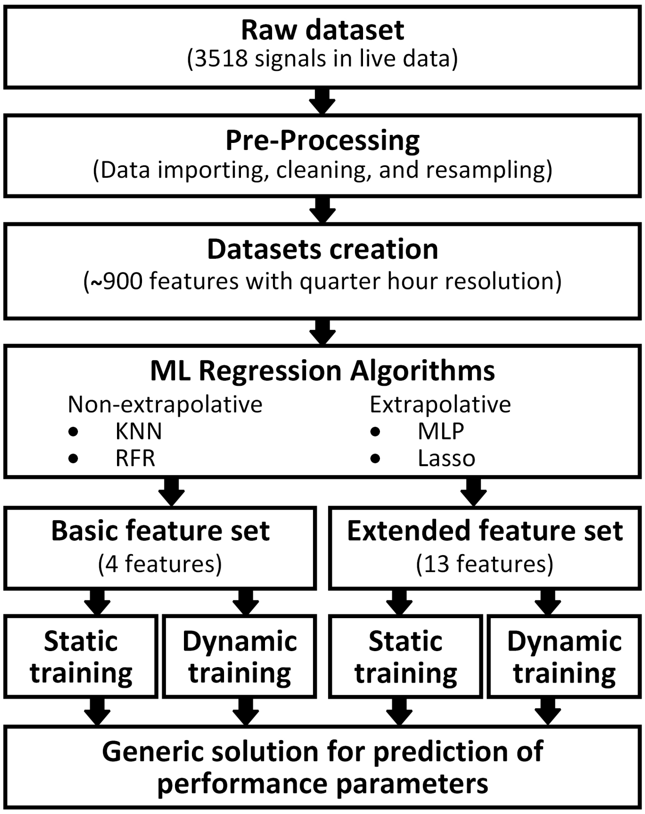

Figure 5. Flow chart of the prediction process

The coefficient of determination R^2 and the mean absolute error (*MAE*) are calculated to estimate the quality of the individual predictions. R^2, as given by Eq. 1, explains how well the variance in the target variable can be explained by the variances of the features.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \tag{1}$$

In addition, we compute *MAE*, a model evaluation metric for regression models. The mean absolute error is computed as the arithmetic mean of the absolute prediction errors of all individual samples in the test set. It is given in Eq. 2:

$$MAE = \frac{\sum_i |\hat{y}_i - y_i|}{n} \tag{2}$$

Although, *MAE* and R^2 typically correlate, mean absolute error is helpful as a quality estimator, as absolute numbers are often more intuitive and aid in interpreting the data.

Discussion and Results

Variation of algorithms

Fig. 5 shows that we use two non-extrapolative (KNN, RFR) and two extrapolative (MLP, Lasso) algorithms. We test the general accuracy and the capability of these algorithms to extrapolate by selecting a designated period of our dataset, where the test period includes values that do not occur in the training period. In Fig. 6 the comparison between the actual and the predicted values are visualized for the training and the testing period. Fig. 6 shows that the ability to extrapolate has a crucial impact on the algorithm's precision. While KNN and RFR suffer from the lack of information in training data, MLP and lasso can still give a decent prediction quality for values that are outside the range of the target in the training period.

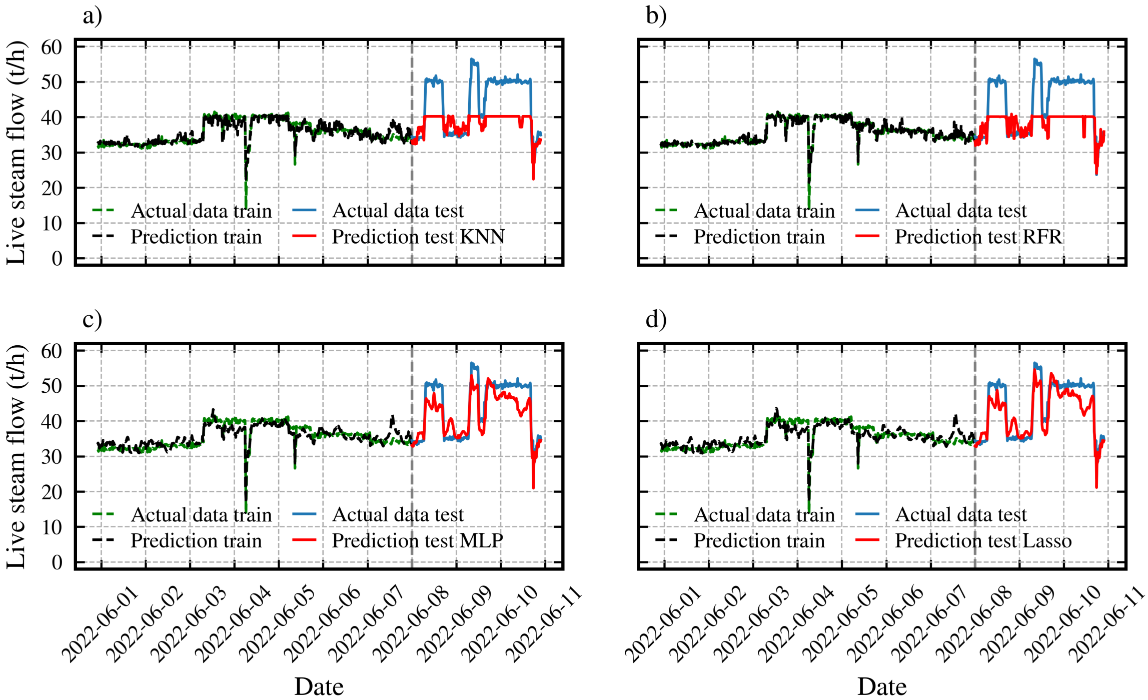

Figure 6. The showcase of algorithms constraints (a) KNN (b) RFR (c) MLP (d) Lasso

Variation of training sets

Fig. 7 visualizes the impact of using a static or a dynamic dataset for training. The average live steam flow is $35.1\ \mathrm{t/h}$ in the investigated period. Additionally, a planned shutdown, which is excluded from the prediction, is shown as greyed-out area. We use daily average values to make the figure more presentable. In general, training the algorithms on a dynamic dataset increases the accuracy of the prediction compared to training on the static dataset. Moreover, before the shutdown, the predictions exclusively overestimate the actual live steam flow, whereas, after the shutdown, they underestimate it. This behavior indicates that some critical changes happened during the shutdown period. Due to the higher prediction accuracy, the dynamic training set can be of great use to power plant operators for detecting changes on a much shorter time scale.

Tab. 2 lists the calculation time, the coefficient of performance R^2, the *MAE*, and the relative difference in MAE between the static and the dynamic training set for all investigated algorithms. Calculation time does not include the time needed for the tuning of the hyperparameters. For the static dataset, the fastest algorithm is more than 30 times faster than the slowest algorithm. As for the dynamic dataset, KNN (fastest) is more than 700 times faster than MLP (slowest). Even though we predict half a year of data, it is still possible to use MLP for daily predictions, but not on live data, because training the algorithm is linked to a high computational effort. With the same training dataset, all algorithms perform similarly in terms of *MAE*, while R^2 is approximately 20% lower for KNN and RFR than for MLP and lasso. The usage of a dynamic training set leads to a 33% improvement on average in both *MAE* and R^2, with small variations for the individual algorithms. Although *MAE* never exceeds $3.5\ \mathrm{t/h}$ with a $35.1\ \mathrm{t/h}$ average, $R^2 \leq 0.73$ and must be improved.

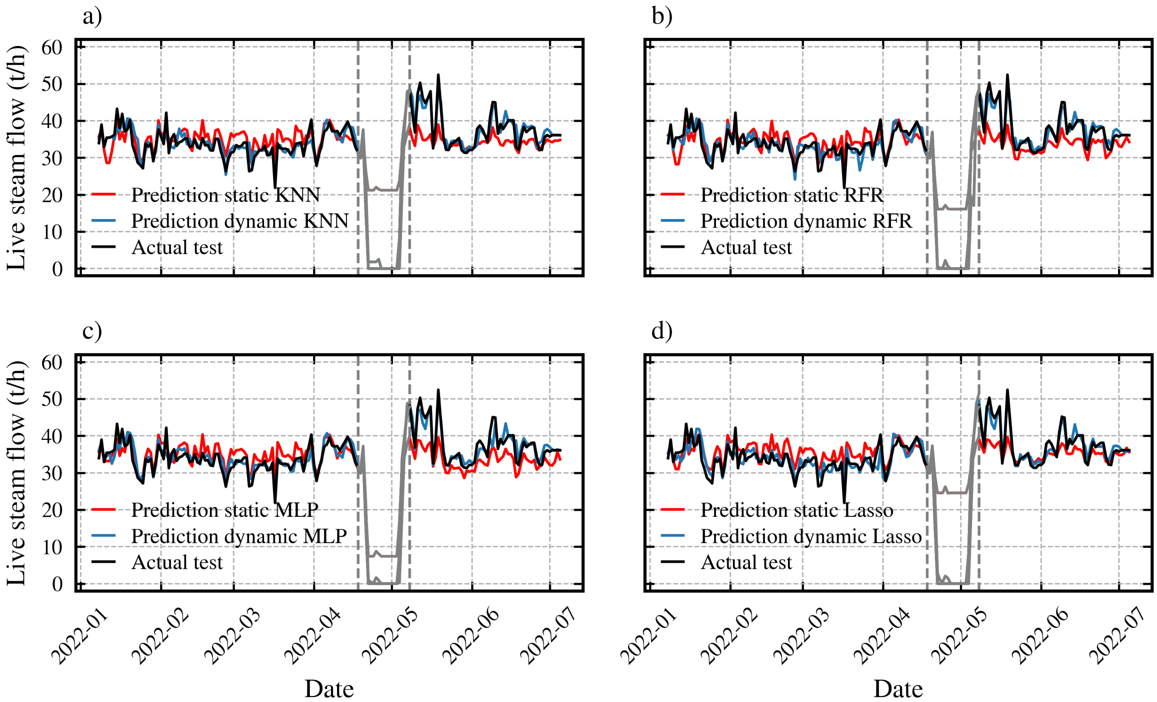

Figure 7. Prediction results in comparison between static and dynamic datasets (a) KNN (b) RFR (c) MLP (d) Lasso (the grey period represents a planned shutdown which was excluded for the prediction)

Table 2. Calculation time, coefficient of performance *MAE* and the relative difference in MAE between the static and the dynamic training set for the basic feature set

Algorithm	Training Set	t_{calc} (s)	R^2 $(-)$	MAE (t/h)	ΔMAE (%)
KNN	Static	1	0.37	3.19	35.6
	Dynamic	1	0.65	2.05	
RFR	Static	1	0.35	3.25	32.6
	Dynamic	8	0.61	2.19	
MLP	Static	22	0.46	3.06	37.6
	Dynamic	718	0.73	1.91	
Lasso	Static	11	0.46	3.06	28.4
	Dynamic	37	0.66	2.19	

Variation of feature sets

Tab. 3 lists the calculation time, the coefficient of performance R^2, and the *MAE* for the extended feature set. For RFR and MLP, the calculation time for the dynamic dataset with an extended feature set is more than 90 times and more than 20 times higher than for the static dataset. KNN and lasso did not notice such a difference. The deviation of *MAE* between the algorithms among training datasets is minor, and only the results of KNN differ from the other three. Directly comparing two training datasets, we see the major *MAE* improvement from $40\ \%$ in the case of KNN up to $68\ \%$ in the case of RFR, the results of MLP and lasso are in between. In turn, R^2 of algorithms increased up to three times.

If we compare Tab. 2 and Tab. 3, we see that all algorithms become more computationally demanding, but still six out of eight combinations have calculation times less than a minute. For the static dataset, swap from the basic to the extended feature set results in a higher *MAE*.

A possible reason for this is that the additional sensors increase the complexity of the model, resulting in overfitting. Removing some of the less correlated features might improve the prediction in these instances. Meanwhile, the use of the extended feature set in combination with the dynamic training set reduces the *MAE* from $2.05\ \mathrm{t/h}$ to $1.92\ \mathrm{t/h}$ in the case of KNN, from $2.19\ \mathrm{t/h}$ to $1.19\ \mathrm{t/h}$ for RFR, from $1.91\ \mathrm{t/h}$ to $1.33\ \mathrm{t/h}$ for MLP, and from $2.19\ \mathrm{t/h}$ to $1.18\ \mathrm{t/h}$ for lasso. We also see that KNN isn't profiting from the extended feature set. That is because of its computational ease and the non-complex model. Ultimately, the extended feature set based on expert knowledge increases R^2 in some cases by more than $50\ \%$ from 0.61 to 0.93 (RFR).

Table 3. Calculation time, coefficient of performance and *MAE* for the extended feature set

Algorithm	Training Set	t_{calc} (s)	R^2 (−)	*MAE* (t/h)
KNN	Static	3	0.31	3.39
	Dynamic	4	0.89	1.92
RFR	Static	2	0.31	3.73
	Dynamic	186	0.93	1.19
MLP	Static	57	0.41	3.84
	Dynamic	1120	0.94	1.33
Lasso	Static	20	0.52	3.41
	Dynamic	25	0.91	1.18

The results show that detecting long-term changes is possible using an ANN-based MLP algorithm. Fig. 8 visualizes the best configurations for long-term and short-term state detection in the thermal power plant.

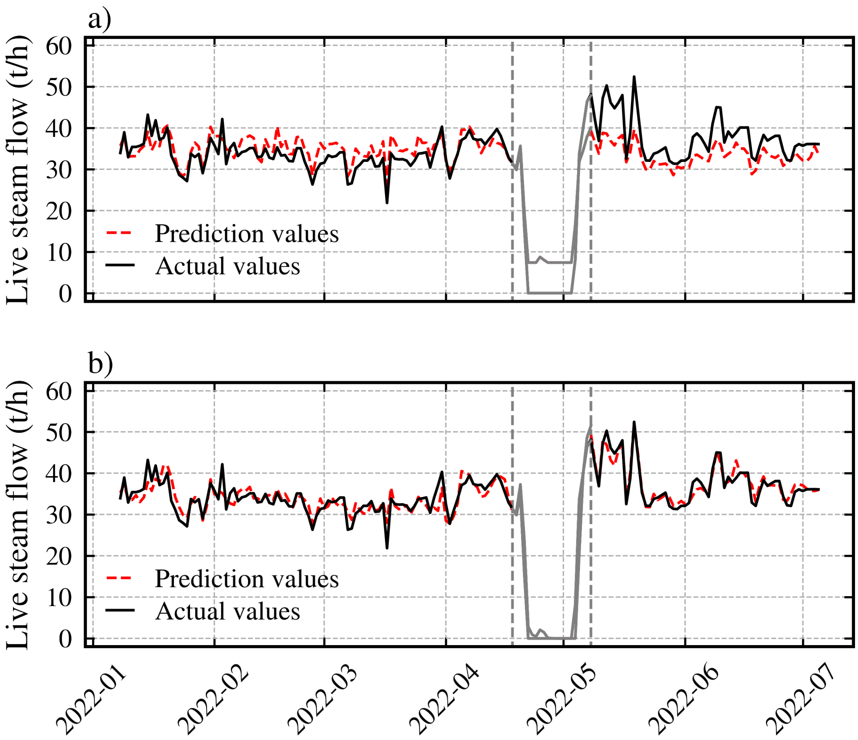

Figure 8. The best results for (a) long-term prediction based on the static dataset, MLP algorithm, and basic feature set (b) short-term prediction based on the dynamic dataset, lasso algorithm, and extended feature set

Fig. 8 (a) visualizes the most appropriate setting for a long-term prediction using the MLP algorithm and four sensors. In our specific case, we can use one week of the data, where we know that the equipment operates in a normal state, and then predict several months. The computational effort is not a crucial criterion, as the algorithm only needs to be computed every couple of months. With the resulting predictions, long-term deterioration or degradation of the equipment is detected.

Fig. 8 (b) represents the result of the Lasso regressor using a dynamic training set and the extended feature set. Although RFR predicts approximately 3% more accurately than Lasso, it is time-consuming for retraining and suffers from an inability to extrapolate, as shown in Fig. 6. Therefore, all aspects considered we see that lasso is one of the best algorithms in every quality estimation.

Conclusion and Outlook

To predict the live steam flow of a thermal power plant with high accuracy, four different machine-learning algorithms are compared. To detect short-term and long-term developments in the power plant, (1) different training sets, (2) different feature sets, and (3) different hyper-parameter settings are investigated. With this respect, the conducted study has come up with the following findings:

- The dependency of the prediction quality is studied for different settings, and the ideal settings result in a live steam prediction with $R^2 = 0.95$.
- Long-term change in the power plant operation can be tracked by predicting with the basic feature set, which includes only fuel inputs.
- A high accuracy prediction is possible by using a lasso regressor and the extended feature set. It can be used to monitor short-term changes in the power plant operation.

Therefore, ideal configurations to monitor short-term and long-term problems in the power plant operation are recommended in our study. To increase the prediction accuracy even further and to detect typical anomalies in the power plant operation, unsupervised learning methods can be of great use. Therefore, in a following study, we plan to focus on unsupervised learning methods that help power plant operators to react effectively and on time to anomalies and state changes.

Acknowledgment

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology, and Development, and the Christian Doppler Research Association are gratefully acknowledged.

References

[1] M. Preißinger. Predictive maintenance in thermal power plants: a systematic literature survey. In *9th HPC Conference Proceedings*, pages 496–515, Bilbao, Spain, April 2022. Heat Powered Cycles (HPC).

[2] Juan José Montero Jimenez, Sébastien Schwartz, Rob Vingerhoeds, Bernard Grabot, and Michel Salaün. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, 56:539–557, July 2020.

[3] Salman Khalid, Hyunho Hwang, and Heung Soo Kim. Real-World Data-Driven Machine-Learning-Based Optimal Sensor Selection Approach for Equipment Fault Detection in a Thermal Power Plant. *Mathematics*, 9(21):2814, November 2021.

[4] Sailaja Thota and Muzamil Basha Syed. Analysis of feature selection techniques for prediction of boiler efficiency in case of coal based power plant using real time data. *Int J Syst Assur Eng Manag*, July 2022.

[5] Prabhas Hundi and Rouzbeh Shahsavari. Comparative studies among machine learning models for performance estimation and health monitoring of thermal power plants. *Applied Energy*, 265:114775, May 2020.

[6] Dong-Xiao Gu, Chang-Yong Liang, Isabelle Bichindaritz, Chun-Rong Zuo, and Jun Wang. A case-based knowledge system for safety evaluation decision making of thermal power plants. *Knowledge-Based Systems*, 26:185–195, February 2012.

[7] Firas Basim Ismail, Deshvin Singh, and Mohammad Shakir Nasif. ADOPTION OF INTELLIGENT COMPUTATIONAL TECHNIQUES FOR STEAM BOILERS TUBE LEAK TRIP. *MJCS*, 33(2):133–151, April 2020.

[8] Cody W. Allen, Chad Holcomb, and Maurício de Oliveira. Anomaly Detection for Large Fleets of Industrial Equipment: Utilizing Machine Learning With Applications to Power Plant Monitoring. In *Volume 4: Controls, Diagnostics, and Instrumentation; Cycle Innovations; Cycle Innovations: Energy Storage; Education; Electric Power*, page V004T05A016, Virtual, Online, June 2021. American Society of Mechanical Engineers.

[9] Nong Nurnie Mohd Nistah, King Hann Lim, Lenin Gopal, and Firas Basim Ismail Alnaimi. Coal-Fired Boiler Fault Prediction using Artificial Neural Networks. *IJECE*, 8(4):2486, August 2018.

[10] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140, September 2014.

[11] Vedika Agrawal, Shubham Agrawal, Sayak Nag, Debayan Chakraborty, B. K. Panigrahi, and P. M. V. Subbarao. Application of K-NN regression for predicting coal mill related variables. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pages 1–9, Nagercoil, India, March 2016. IEEE.

[12] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.