Heat demand forecasting in District Heating Network using XGBoost algorithm

Maciej Bujalski[1,2*], *Paweł* Madejski[1], and *Krzysztof Fuzowski*[3]

[1]AGH University of Science and Technology, Mickiewicza 30, 30-059 Kraków
[2]PGE Energia Ciepła S.A., Złota 59, 00-120 Warszawa
[3]PGE Baltica sp. z o.o., Mokotowska 49, 00-542 Warszawa

Abstract. Forecasting an hourly heat demand during different periods of district heating network operation is essential to optimize heat production in the CHP plant. The paper presents the heat demand model in the real district heating system with a peak load of 200 MW. The predictive model was developed with the use of the machine learning method based on the historical data. The XGBoost (Extreme Gradient Boosting) algorithm was applied to find the relation between actual heat demand and predictors such as weather data and behavioral parameters like an hour of the day, day of week, and month. The method of model training and evaluating was discussed. The results were assessed by comparing hourly heat demand forecasts with actual values from a measuring system located in the CHP plant. The RMSE and MAPE error for the analysed time period were calculated and then benchmarked with an exponential regression model supplied with ambient air temperature. It was found that the machine learning method allows to obtain more accurate results due to the incorporation of additional predictors. The MAPE and RMSE for the XGBoost model in the day-ahead horizon were 6.9% and 8.7MW, respectively.

1 Introduction

District heating systems are commonly used for useful heat production and distribution in central end eastern Europe. Typical system consist of heat generation units supplying district heating network (DHN). The system's effective operation depends on the accuracy and reliability of heat demand predictions in the short and long term horizon [1]. Hourly heat demand in the day-ahead is needed for short-term planning and optimization of energy production in cogeneration heat and power plants (CHP) [2]. In district heating systems, the production of electricity is dependent on the actual heat load. Thus, a precise heat load forecast is also relevant to estimate electricity production for trading in the day-ahead market.

Actual heat demand in DHN depends mainly on the weather data and end-user behavior. Non-stationary effects associated with heat accumulation and losses in DHN [3] and thermal inertia in buildings [4] should also be considered.

Heat demand models are mainly based on the data-driven approach where historical data from the operation of the system are used. Primarily, weather data such as ambient air temperature, wind speed, and solar irradiation are needed. In the literature, the application of the predictive model in real district heating systems can be found. Dotzdauer [5] developed a simple model based on linear regression of ambient air temperature. Baltputnis et al. [6] used a polynomial regression of outdoor temperature. Fang et al. [7] applied multiple regression of air temperature and wind speed to improve heat demand forecasts. It was found by Bianchi et al. [8] that the heat demand model is more accurate if it includes social components such as holidays, the day of the week and the hour of the day. The use of multiple predictors over a long period of time often requires the application of advanced algorithms such as machine learning (ML) methods. In this case, the regression supervised learning technique is used. It aims to model the relationship between a certain number of predictors and a continuous target variable.

Recently, new solutions using ML are exploring in the field of heat load prediction for a large DHN [9] or individual building [10]. Idowu [11] et al. tested machine learning-based approaches (support vector regression, decision tree, feed-forward neural network) based on the data coming from residential and commercial buildings. Kurek et al. [12] successfully applied an artificial neural network in a real large-scale DHN. Saloux et al. [13] compared ML models with linear regression of ambient temperature and proved that ML leads to obtain more accurate and reliable results. Dahl et al. [14] demonstrated a support vector regression model supplied with the weather, calendar, and holiday data. It was shown that calendar data could significantly improve the accuracy of the model due to the capture of social patterns.

In this paper, the XGBoost algorithm was applied to develop the heat demand model in a case study DHN. The algorithm provides an implementation of the gradient boosted trees algorithm that was proposed by

* Corresponding author: bujalski@agh.edu.pl

Chen et al. [15] in 2016. It has become a popular method used in various data mining scenarios and algorithm competitions. In the following chapters, the method of training and evaluation is presented.

2 Heat demand model for case study DHN

The heat demand model was developed for a real case study district heating system comprised of a gas-fired CHP plant supplying DHN with approximately 100,000 end-users. A peak heat load during winter is around 200 MW. During the summer period, the heat load for domestic hot water ranges from 20 to 30 MW. The heat output at the CHP is regulated by changing the supply temperature as well as flow rate of hot water taking into account current demand. Figure 1 presents the scatter plot of ambient air temperature and heat demand over a calendar year of the operation. It can be noticed a large spread for the same temperature values. Thus, in order to increase the accuracy of the heat load forecast, it is necessary to consider additional weather and other parameters. Depending on the district heating system, solar irradiation and wind speed can also be significant predictors for the heat load.

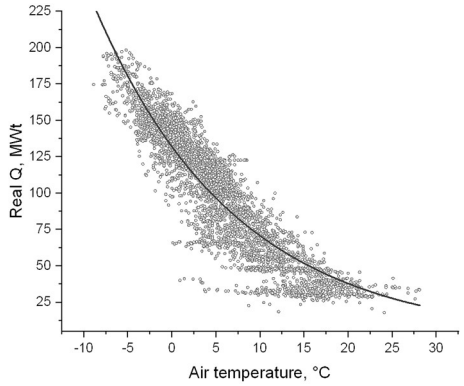

Fig. 1. Hourly heat demand with ambient air temperature

2.1 XGBoost algorithm

XGBoost is an implementation of gradient tree boosting technique, which combines multiple weak classifiers into a strong classifier. The main idea is to sequentially add decision trees to the ensemble model to improve the accuracy. The target variable is predicted using additive functions as in Equation 1 [15].

$$\bar{y}_i = y_i^0 + \eta \sum_{k=1}^{M} f_k(X_i) \quad (1)$$

where $\bar{y}_i$ is the predicted result based on features X_i, y_i^0 is the initial guess and η is the learning rate that helps to improve smoothly the model while adding new trees and avoid overfitting [15].

The estimation f_k of the additional k-th estimators is presented in Equation 2 [16].

$$\bar{y}_i{}^k = \bar{y}_i{}^{(k-1)} + \eta f_k \quad (2)$$

where $\bar{y}_i{}^k$ is the k-th predicted result and f_k is defined by the leaves weights. To learn the functions used in model above, the following regularized objective $L(\phi)$ is minimized (Eq. 3) [16].

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (3)$$

Here, $l(\hat{y}_i, y_i)$ is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$and the target y_i. The second term Ω penalizes the complexity of the model and it acts as an additional regularization term helps to avoid overfitting. XGBoost uses second-order Taylor expansion to expand the loss function in the gradient boosting process in an iterative way [17].

In this work, the extreme gradient tree boosting framework in *R* programming language and open-source library named *xgboost* [18] was used.

2.2 Evaluation metrics

The model forecasts were examined with RMSE (Root Mean Square Error) as in Equation 4. MAPE (Mean Absolute Percentage Error) as in Equation 5. was also used as a evaluation metric.

$$RMSE = \sqrt{\frac{1}{N}\sum_t \left(Q_{pred,i} - Q_{real,i}\right)^2} \quad (4)$$

$$MAPE = \frac{100\%}{N}\sum_t \left|\frac{Q_{pred,i} - Q_{real,i}}{Q_{real,i}}\right| \quad (5)$$

where $(Q_{pred} - Q_{real})$ is the difference between the predicted (in the day-ahead) and real heat load, t corresponds to hour and N is a total number of hours in the analysed period.

The ML model with the XGBoost algorithm was compared with a simple exponential regression with one predictor (Equation 6).

$$Q_{pred,i} = Ae^{(t_{am,i} \cdot B)} \quad (6)$$

Where t_{am} is the ambient air temperature, A and B coefficients were determined for each month separately for the learning dataset using nonlinear fitting.

2.3 Input variables

The target variable for the predictive model was actual heat demand, aggregated in an hourly resolution by the heat meter located at the outlet of CHP to DHN. The following weather data was used to train the model:

- t_{am} – ambient air temperature (°C),
- v_{wind} – wind speed (m/s),
- I_{rad} – solar irradiation (W/m^2),
- φ – humidity (%).

The algorithm was trained with the real weather data while weather forecasts in the day-ahead horizon were used to generate predictions. Additional parameters were included using categorical variables representing social behaviour of the end-users:

- *hour* – hour of the day,
- *day* – day of the week,

- *month* – month of the year.

The historical data aggregated at 1 hour time interval from two subsequent heating seasons were used (from September 2016 to December 2018). The learning dataset was divided – 75% was used as a training set and 25% for validation. The validation part of the data was used to check the model accuracy on the set that was not used for learning the algorithm. A model hyper-parameters was tuned to obtain high accuracy during validation. Then, the previously developed model was used to generate predictions for the data from January – April 2019.

2.4 Model training and validation

An important aspect during the implementation of ML models is to avoid overfitting. A model can be extremely trained to the learning datasets while it could give significant error on the new dataset. An optimal set of input variables and hyper-parameters were found in order to minimize RMSE on the validation dataset. In Figure 2, RMSE of the XGBoost model is presented for each dataset. The analysis includes two cases. In the first, only weather data were used to build the model. In the second, additional parameters such as an hour of the day, day of week, month were additionally included. It can be found that the inclusion of social components by incorporating calendar data, results in increased accuracy of the model on both datasets. Since the model is supplied by weather forecast during operation, relevant predictors are affected by forecast error. The average difference between predicted and actual value for the analyzed dataset was 1.51 °C for air temperature, 0.68 m/s for wind speed and 45.9 W/m^2 for solar irradiation. RMSE error on the validation dataset was calculated both using real and forecasted weather data in the day-ahead horizon (Figure 2). The analysis shows that RMSE error is greater by approximately 2.3 when forecast weather data is used.

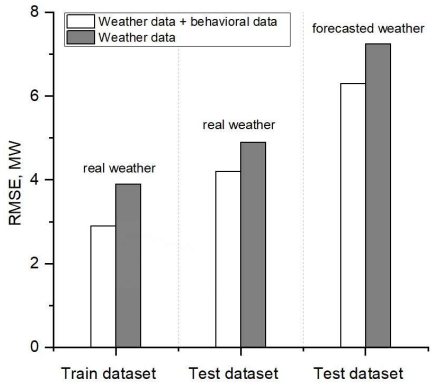

Fig. 2. RMSE error on the train and test dataset depending on the set of input variables.

During training the model, three hyper-parameter were adjusted in order to minimize error. These are: *n_estimators* (the number of iterations in training), *max_depth* (maximum depth of a tree), *learning_rate* (reduce the weight of each step to make the model more robust) and *min_split_loss* (defines the sum of sample weight of the smallest leaf nodes to prevent overfitting) [19]. Other parameters in the package were left with their default values. In Figure 3, the RMSE error during iteration process of learning the model is presented, respectively for two sample sets of hyper-parameters. The problem of overfitting was encountered as was presented in the Figure 3a. As the iteration of the model increases, the error increases on the test set (that was not used for training). Reduction in depth of the tree and learning rate led to an improvement in accuracy. The selected values of parameters are as follows: *n_estimators* = 300, *max_depth* = 8, *learning_rate* = 0.1, *min_split_loss* = 3.

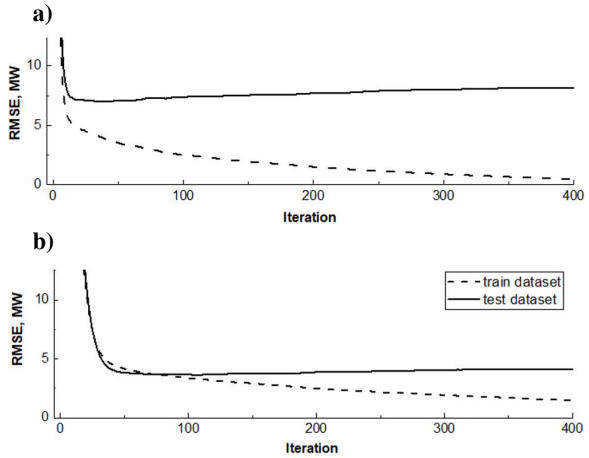

Fig. 3. RMSE error on the train and test dataset depending on the set of hyperparameters values. (a) *max_depth = 12, learning_rate = 0.3, min_split_loss = 1* (b) *max_depth = 8, learning_rate = 0.1, min_split_loss = 3*

3 Results and discussion

The paragraph discusses the accuracy of the developed heat demand model in a case study DHN during the heating season. The analysis covers the period from January to April 2019. This dataset was excluded from the training and validation process of the model. In Figure 4, the time course of hourly absolute percentage error is presented together with real values of heat demand. It can be noticed that the model gives a larger error during March and April, where the air temperature is relatively higher, and significant fluctuations of heat load over the day occur, both in magnitude and variance. Moreover, the weather forecast is more inaccurate during this period which affects additional model errors. The aggregated error metrics such as MAPE varies over the months, but it is also important to look at instantaneous relative residuals. Maximum errors of the model can be relevant for production planning in CHP plant, particularly for trading in the electricity market. A vast majority (about 90%) of relative errors are in the range of -10% to 10%. The maximum error value is more than 30%.

Table 1 summarizes the outcome from the analysis where evaluation metrics are provided and benchmarked with reference model for the whole analysed period. It can be noticed that a simple model-driven by only one parameter (ambient air temperature) gives greater

inaccuracy. RMSE error for ML method is smaller by 5.9 MW as well as MAPE metric and distribution (1st and 3rd quartile) is also significantly slighter. MAPE and RMSE error of hourly heat load forecasts during analysed period of the heating season (from January to end of April) was 6.86 % and 8.67 MW, respectively. The obtained accuracy of the predicted heat load is worse compared with learning and validation dataset, so it is worth considering more frequent calibration of the model during its online operation.

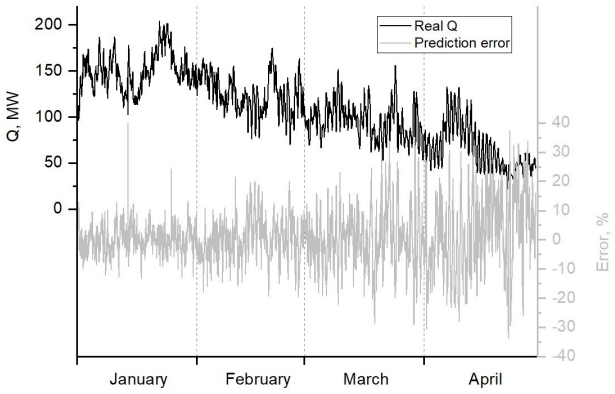

Fig. 4. Real vs. predicted heat load with XGBoost algorithm

Table 1. Comparison between machine learning and the exponential model (Jan – April)

Metric	XGBoost model	Exponential model
RMSE	8.67 MW	14.53 MW
MAPE	6.86 %	12.92 %
1st Quartile of MAPE	2.17 %	4.26 %
Median	4.85 %	9.10 %
3rd Quartile of MAPE	9.52 %	15.64 %

4 Conclusions

Predicting day-ahead heat load in the District Heating System is essential to effective optimization of heat production in CHP plants. Various machine learning approaches can be applied to deal with this problem. The extreme gradient boosting method and XGBoost library has been considered and proven to be an effective tool for heat load forecasting in case study DHN. One of its major strengths is the ability to include a lot of data and predict short term fluctuations of heat load without knowledge of the underlying physical principles.

It has been found that the proposed method can significantly improve the accuracy of prediction by capturing the additional factors related to end-users behaviour like day of the week, hour of the day, month.

References

1. H. Nielsen, H. Madsen Energy Build. **38** (1):63-71 (2006)
2. P. Żymełka, M. Szega, P. Madejski, P. J. Energy Resour. Technol. **142** (2020)
3. M. Gong, Y. Bai, J. Qin, J. Wang, P. Yang i S. Wang, J. Build. Eng. **27**, 2020.
4. L. Bellahcene1, A. Cheknane, S. Bekkouche, D. S ahel, E3S Web Conf, **22**, 00013 (2017)
5. E. Dotzauer, Appl. Energy **73**, 277–284 (2002)
6. K. Baltputnis, R. Petrichenko, D. Sobolevsky, In IEEE 6th Workshop on Advances in Information Electronic and Electrical Engineering (2018)
7. T. Fang, R. Lahdelma, R. Appl. Energy 2016, **179**, 544–552.
8. F. Bianchi, P. Tarocco, A. Castellini, A. Farinelli Lect. Notes Comput. Sci. **12565** (2020)
9. J. Liu, X. Wang, Y. Zhao, B. Dong, K. Lu and R. Wang, in *IEEE Access*, **8** (2020)
10. L. Zhang, J. Wen, Y. Li, J. Chen, Appl. Energy, **285** 254:116452 (2021)
11. S. Idowu, S. Saguna, C. Åhlund, O. Schelén, Energy Build. **133**, 478–488 (2016)
12. T. Kurek, A. Bielecki, K. Świrski, K. Wojdan, M. Guzek, M. Białek, J. Brzozowski, R. Serafin, Energy **217** (2021)
13. E. Saloux, J.A. Candanedo, Energy Procedia **149**, 59–68 (2018)
14. M. Dahl, A. Brun, O.S. Kirsebom, G.B. Andresen, Energies **11**, 1678 (2018)
15. T. Chen, C. Guestrin, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
16. Z. Wan, Y. Xu, B. Šavija, Materials **14**, 713. (2021)
17. L. Zhang, W. Bian, W. Qu, L. Tuo, Y. Wang, J. Phys.: Conf. Ser. **1873** 012067 (2021)
18. T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, *xgboost: Extreme Gradient Boosting. R package version 1.4.1.1.* (2021)
19. W. Li, Y. Yin, X. Quan, H. Zhang, Front. Genet. **10** (2019)