



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Letícia Carvalheiro Navarro

Predicting Temperature in Blast Furnaces Using Machine Learning Regression Methods

Vitória, ES

2023

Letícia Carvalheiro Navarro

Predicting Temperature in Blast Furnaces Using Machine Learning Regression Methods

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Supervisor: Prof. Dr. Thiago Oliveira dos Santos

Co-supervisor: Dr. Rodrigo Ferreira Berriel

Vitória, ES

2023

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

N322p Navarro, Letícia Carvalheiro, 1997-
Predicting Temperature in Blast Furnaces Using Machine
Learning Regression Methods / Letícia Carvalheiro Navarro. -
2023.
63 f. : il.

Orientador: Thiago Oliveira dos Santos.
Coorientador: Rodrigo Ferreira Berriel.
Dissertação (Mestrado em Informática) - Universidade
Federal do Espírito Santo, Centro Tecnológico.

1. Inteligência artificial. 2. Aprendizado do computador. I.
Santos, Thiago Oliveira dos. II. Berriel, Rodrigo Ferreira. III.
Universidade Federal do Espírito Santo. Centro Tecnológico. IV.
Título.

CDU: 004



Predicting Temperature in Blast Furnaces Using Machine Learning Regression Methods

Letícia Carvalho Navarro

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 07 de Março de 2023.

Prof. Dr. Thiago Oliveira dos Santos
Orientador

Prof. Dr. Rodrigo Ferreira Berriel
Coorientador

Prof. Dr. Alberto Ferreira De Souza
Membro Interno

Prof. Dr. George Darmiton da Cunha Cavalcanti
Membro Externo, participação remota

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória/ES, 07 de março de 2023



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA



O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
THIAGO OLIVEIRA DOS SANTOS - SIAPE 2023810
Departamento de Informática - DI/CT
Em 07/03/2023 às 23:41

Para verificar as assinaturas e visualizar o documento original acesse o link:
<https://api.lepisma.ufes.br/arquivos-assinados/663579?tipoArquivo=O>



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA



O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
ALBERTO FERREIRA DE SOUZA - PROFESSOR VOLUNTÁRIO
Departamento de Informática - DI/CT
Em 10/04/2023 às 20:37

Para verificar as assinaturas e visualizar o documento original acesse o link:
<https://api.lepisma.ufes.br/arquivos-assinados/687746?tipoArquivo=O>

*I dedicate this work to everyone who helped me directly and indirectly to this conclusion.
God, my family, friends, co-workers, teachers and my cat.*

Acknowledgements

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Brazil) - Finance Code 001; Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil); and Fundação de Amparo à Pesquisa do Espírito Santo (FAPES, Brazil) - grant 2021-07KJ2.

*“All truths are easy to understand
once they are discovered;
the point is to discover them
(Galileo Galilei)”*

Resumo

Na indústria siderúrgica, a operação estável dos altos-fornos com monitoramento e controle eficientes da temperatura do ferro gusa é uma tarefa muito importante no processo de geração de ferro gusa de alta qualidade. Em geral, a operação dos altos-fornos depende principalmente de decisões baseadas na experiência de operadores humanos, que usam as medidas mais recentes de temperatura do gusa líquido e outras variáveis operacionais para executar as decisões de controle. No entanto, devido ao grande número de variáveis e à complexa interação entre elas, a operação de tais equipamentos não é uma tarefa fácil. Este trabalho propõe um sistema de predição como primeiro passo de um sistema de controle maior e mais complexo para melhorar a eficiência da produção de ferro considerando o cenário brasileiro. Ele compara vários modelos de aprendizado de máquina (K-Nearest Neighbors, Linear Regression, Extreme Boosting Machine, Light Gradient Boosting Machine, Random Forest, Support Vector Machine, XGBoost e Multilayer Perceptron) na tarefa de previsão da temperatura do gusa. Um bom sistema de predição de temperatura permitirá planejar melhor as ações de controle a seguir, a fim de estabilizar a temperatura do forno durante a produção de gusa. O método proposto foi avaliado com dados reais de uma empresa siderúrgica. Resultados mostraram que o sistema conseguiu prever a temperatura do ferro gusa com um erro médio absoluto de 9,56 comparado ao baseline que teve um erro de 12,61.

Abstract

In the iron and steel industry, the stable operation of blast furnaces with efficient hot metal temperature monitoring and control is a very important task in the process to generate high-quality hot metal. In general, the operation of blast furnaces mostly relies on experience based decisions of human operators, which use the most recent measures of hot metal temperature and other operational variables to execute control decisions. However, due to the large number of variables and complex interaction among them, the operation of such equipment is not an easy task. This work proposes a prediction system as the first step of a larger and more complex control system for improving the efficiency of iron production considering the scenario in Brazil. It compares several machine learning models (K-Nearest Neighbors, Linear Regression, Extreme Boosting Machine, Light Gradient Boosting Machine, Random Forest, Support Vector Machine, XGBoost, and Multilayer Perceptron) in the task of hot metal temperature prediction. A good temperature prediction system will allow to better plan the control actions ahead in order to stabilize the furnace temperature during hot metal production. The proposed method was evaluated using real-world data from an steel-producing company. Results shown that the system can predict the hot metal temperature with mean absolute error of 9.56 when compared to the baselines with mean average error of 12.61.

List of Figures

Figure 1 – Tap hole drawing off hot metal - Source: Pixabay (https://pixabay.com/photos/industry-stole-iron-blast-furnace-647413/) under Pixabay license	16
Figure 2 – Example of classification and regression models	21
Figure 3 – Blast furnace components	27
Figure 4 – Tap holes distribution in a Blast Furnace with four tap holes	28
Figure 5 – Modules and flow of information. The system receives a set of control variables, temperatures and delta times which are preprocessed and used to predict a temperature in the future	31
Figure 6 – Illustrates a casting with four measurements with the maximum temperature of that casting marked	34
Figure 7 – Illustrates, in time, the input and output information of a prediction at t_p with the user at time t_0 considering information of n hours prior t_p	35
Figure 8 – The temperature difference in two tap holes (Tap Hole 2 and 4)	35
Figure 9 – The temperature difference in two tap holes (Tap Hole 1 and 3)	36
Figure 10 – Illustrates the generation of the $T_{projected}$	36
Figure 11 – Qualitative experiment: predicting t_{+1}	49
Figure 12 – Qualitative experiment: predicting t_{+2}	50
Figure 13 – Qualitative experiment: predicting t_{+2}	51
Figure 14 – Qualitative experiment: error of predicting t_{+1} , t_{+2} and t_{+3} (top to bottom) between 2021-01-21 and 2021-01-31.	52
Figure 15 – Mask Experiment: Increased PCI variable t_{+1} – Absolute Temperature	53
Figure 16 – Mask Experiment: Increased Oxygen Rate variable t_{+1} – Absolute Temperature	54
Figure 17 – Qualitative experiment: predicting t_{+1} with projected temperature	55
Figure 18 – Qualitative experiment: predicting t_{+1} with projected temperature	55
Figure 19 – Mask Experiment: Increased PCI variable t_{+1} – Projected Temperature	56
Figure 20 – Mask Experiment: Increased Oxygen Rate variable t_{+1} – Projected Temperature	56

List of Tables

Table 1 – Expected Time to affect the Temperature	29
Table 2 – Control variables used as input in the system and description of each of them	32
Table 3 – Sensor variables used as input in the system and description of each of them	33
Table 4 – History Length Experiment - Validation Result	45
Table 5 – Main experiment – Predicting t_{+1} – 8-Hours History Length *Baseline methods	46
Table 6 – Main experiment – Predicting t_{+2} . – 8-Hours History Length *Baseline methods.	47
Table 7 – Main experiment – Predicting t_{+3} . – 8-Hours History Length *Baseline methods.	47
Table 8 – Tap Hole Experiment – Absolute Temperature – Predicting t_{+1} – 8-Hours History Length	53
Table 9 – Projected Temperatures – Predicting t_{+1} – 8-Hours History Length . . .	54
Table 10 – Tap Hole Experiment – Projected Temperature – Predicting t_{+1} – 8-Hours History Length	57
Table 11 – Sensors variables experiment - Absolute Temperature – Predicting t_{+1} – 8-Hours History Length * Baseline methods.	58
Table 12 – Sensors variables experiment - Projected Temperature – Predicting t_{+1} – 8-Hours History Length * Baseline methods.	58

List of abbreviations and acronyms

EBM	Explainable Boosting Machine
EFB	Exclusive Features Bundling
GOSS	Gradient-Based One-Side Sampling
HMT	Hot Metal Temperature
LGBM	Light Gradient-Boosting Machine
LR	Linear Regression
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MSE	Mean Squared Error
MLP	Multi Layer Perceptron
PLS	Partial Least Square
RMSE	Root Mean Square Error
RF	Random Forest
SVR	Support Vector Regression
TM	Trend Metric

Contents

1	INTRODUCTION	16
1.1	Objectives	18
1.1.1	Specific objectives	18
1.2	Proposal	18
1.3	Contributions	18
1.3.1	Paper	19
1.3.2	Offline System	19
1.4	Structure	19
2	THEORETICAL BASIS	20
2.1	Supervised Learning	20
2.2	The Regression Problem	21
2.3	Regressors	22
2.3.1	Linear Regression	22
2.3.2	Partial Least Square	23
2.3.3	Random Forest	23
2.3.4	EBM	24
2.3.5	LGBM	24
2.3.6	XGBoost	24
2.3.7	KNN	25
2.3.8	SVR	25
2.3.9	MLP	26
2.4	Blast Furnace Description	26
2.4.1	Temperature Measurement	28
2.4.2	Blast Furnace Variables	28
2.5	Literature Revision	29
3	TEMPERATURE PREDICTION SYSTEM	31
3.1	Overview	31
3.2	Input Variables	32
3.2.1	Control Variables	32
3.2.2	Sensors Variables	33
3.2.3	Summarization	33
3.3	Model with Absolutes Temperatures	34
3.4	Model with Projected Temperatures	35
3.5	Preprocessing	37

3.5.1	Noise Removal	37
3.5.2	Normalization	37
3.6	Prediction	38
4	EXPERIMENTAL METHODOLOGY	39
4.1	Dataset	39
4.2	Hyper-Parameter Search	39
4.3	Evaluation Metrics	41
4.4	Experiments	42
4.4.1	Absolute Temperature	42
4.4.1.1	History Length	42
4.4.1.2	Main Experiment - Predictions Ahead	42
4.4.1.3	Custom History Length	43
4.4.2	Projected Temperature	43
4.4.2.1	Main Experiment	43
4.4.2.2	Custom History Length	43
4.4.3	Sensor Variables	43
4.5	Baselines	44
4.6	Computational performance	44
5	RESULTS AND DISCUSSION	45
5.1	Absolute Temperature	45
5.1.1	History Length	45
5.1.2	Predictions Ahead	46
5.1.3	Analysis by Tap Hole	48
5.1.4	Custom History Length	52
5.2	Projected Temperature	53
5.2.1	Main Experiment	54
5.2.2	Custom History Length	55
5.2.3	Analysis by Tap Hole	56
5.3	Sensors Variables Analysis	57
5.4	Limitations	58
6	CONCLUSION	60
	BIBLIOGRAPHY	61

1 Introduction

Iron and steel industries constantly optimize their processes to achieve high-quality products with greater efficiency and, consequently, greater profits. The steel industry plays an important role in the world economy by providing steelmaking raw materials for other industries, such as automotive, aviation, and civil construction (GONZÁLEZ; KAMIŃSKI, 2011). The main component in the ironmaking process is the blast furnace, a huge metallurgical reactor used to produce hot metal from iron ore. The furnace is fed from the top with raw metal materials and fuel. Hot air is also injected from tuyeres at the bottom of the furnace. These and several other variables are involved in this complex process which results in hot metal and slag that runs off from tap holes at the lower part of the furnace, as can be seen in Figure 1 and flows through the casting channels to their next step in the ironmaking process (ZHANG; KANO; MATSUZAKI, 2019a).



Figure 1 – Tap hole drawing off hot metal - Source: Pixabay (<https://pixabay.com/photos/industry-stole-iron-blast-furnace-647413/>) under Pixabay license

A stable operation of blast furnaces with efficient hot metal temperature (HMT) monitoring and control is very important to generate high-quality hot metal. To achieve this, the temperature of the hot metal that runs off from the tap holes should be kept close to 1500°C. Monitoring the HMT is essential for measuring the quality of the final product as well as the thermal state of the furnace. Lower HMT values can lead to the obstruction of the tap holes and difficulties in separating the hot metal from the slag. In contrast, high HMT values might indicate elevated fuel consumption and increased CO₂ emission. Moreover, the variation of HMT out of the desired operational range may

increase costs in the following steps of the ironmaking process due to a larger amount of impurities in the hot metal (such as silicon), increase maintenance costs and cause productivity losses. (WANG; HU; TANG, 2021; ZHANG; KANO; MATSUZAKI, 2019b; HASHIMOTO; SAWA; KANO, 2019).

The thermal control and operation of blast furnaces mostly rely on decisions of experienced human operators, which use the most recent measures of hot metal temperature and other operational variables to execute control decisions. However, due to the large number of variables and complex interaction among them, the operation of such equipment is not an easy task. Furthermore, due to the limitations of temperature sensors, it is difficult to directly measure temperatures inside the furnace. Therefore, in practice, the usual way is periodically measuring it from the hot metal on the casting channels with thermocouple sensors (ZHANG; KANO; MATSUZAKI, 2019a; HASHIMOTO; SAWA; KANO, 2019; AZADI et al., 2022).

Machine learning has been widely used in forecasting tasks in industrial environments. Increasingly inserted in practical environments, regression methods are gaining more and more space in the market with the application of real-world data, not just in academia. The blast furnace companies have been using machine learning techniques to better predict and control their processes, which are very complex.

Works in the literature address prediction in blast furnaces as a time series analysis. The pioneering work of Pandit et al. (PANDIT; CLUM, 1975) in the 1970s, for instance, applies multivariate linear analysis of times series for modeling, prediction, and control of blast furnaces. On that occasion, they addressed the prediction of quality parameters of hot metal based on the sulfur and silicon content, and on the temperature of the hot metal.

Despite the potential of utilizing time series methods for predicting hot metal temperature, the requirement for uniformly spaced input data posed a challenge. Given the noisy nature of the data, as will be elucidated later, it was decided not to use a time series approach in this study and to make this problem a regression problem.

Throughout the years, other regression models were incorporated into furnace-related parameters prediction, such as VARMAX (ÖSTERMARK; SAXÉN, 1996) and, in particular, Partial Least Square (PLS) (LIN et al., 2011), a technique that addresses the two main limitations of the standard least squares: non-linearity issues and correlation among independent variables. Nonetheless, the evolution of the Machine Learning field, with the Support Vector Machines (SVMs) and the Artificial Neural Networks – particularly the Multi-Layer Perceptron (MLP) –, has enabled more robust models to treat the inherent complexity of industrial processes (BAG, 2007; GAO; JIAN; LUO, 2011; PETTERSSON; CHAKRABORTI; SAXÉN, 2007). Such processes are subjected to several exogenous (natural or human-induced) factors characterized by noise and data loss.

Although there are works in the field, they are usually not very accessible in terms of code and data, hindering comparison and further use. This added to the fact that each blast-furnace impose different particularities to the machine learning algorithms, the problem still deserves investigation.

1.1 Objectives

The main objective of this work is the development of a system for predicting the Hot Metal Temperature of a Blast Furnace.

1.1.1 Specific objectives

To achieve this objective, the following procedures were carried out:

- Investigation of different modeling approaches for the HMT prediction problem, focusing on regression-based models;
- Evaluation of classical and well-known machine learning based regression methods applied to the HMT prediction problem;
- Assessing of the qualitative performance of the investigated methods applied to various modeling approaches on real-world data from an operating blast furnace.

1.2 Proposal

This work proposes a machine-learning based prediction system as the first step of a larger and more complex control system for improving the efficiency of iron production considering the scenario of a company in Brazil. It compares several machine learning models (e.g., K-Nearest Neighbors (KNN), PLS, and MLP) for HMT prediction, while presenting a new modeling problem based on control variables and temperatures as input without using time series. A good temperature prediction system will allow to better plan the control actions ahead in order to stabilize the furnace temperature during hot metal production.

1.3 Contributions

This work brings two contributions - A paper and an offline system - described below.

1.3.1 Paper

As a result of the work developed in this dissertation, an article (NAVARRO et al., 2022) was submitted and accepted for publication at the SIBGRAPI 2022 - 35th Conference on Graphics, Patterns and Images. This publication, was written in collaboration with members of the Instituto de Inteligência Computacional Aplicada and of the industrial partner.

1.3.2 Offline System

Using the HMT forecasting system proposed in this dissertation, an offline system was created to perform the forecast in offline mode using data from recent years. Based on that offline system, an online system was developed and deployed by our team at the our industrial partner, where it remains operational.

1.4 Structure

The remainder of this dissertation is structured as follows: Section. 2 describes the theoretical basis; Section. 3 presents the proposed system; Section. 4 defines the experimental methodology; Section. 5 shows the results and analysis, and, finally, the conclusions are presented in Section. 6.

2 Theoretical Basis

This section describes the concepts needed to understand the proposed method. Section 2.1 describes Supervised Learning. Section 2.2 talks about the regression problem. In Section 2.3 are some popular regressors in the literature. Section 2.4 brings information about the operation of blast furnaces and Section 2.5 a review of the literature on the subject.

2.1 Supervised Learning

Supervised learning is a machine learning technique that uses labeled data to train a model to predict an output label given an input feature. Labels are the desired output of the input. The labeled data is a set of pairs, formed by a instance x and a label y in: $\{(x_i, y_i)\}_{i=1}^n$, this way each input has a respective output. The term supervised learning comes from the idea of having an instructor or teacher guiding the model through what to do. (GOODFELLOW; BENGIO; COURVILLE, 2016; ZHU; GOLDBERG, 2009; ZHANG et al., 2021)

In supervised learning, the labeled data is used to training a function f seen in Equation 2.1 to discover a pattern. After training, input data that was not seen by the model will be inserted, and the goal of this algorithm is to find correct labels for this unknown input.

$$f : X \rightarrow Y \quad (2.1)$$

The label can have many domains, it can have a finite set like $\{hot, cold\}$, or it can belong to a domain of real numbers. The domain label defines the type of problem to be solved. When the domain is a finite set of discrete values the problem to be solved is the **Classification**, and when the Y is a continuous value the problem is the **Regression**. The Figure 2 illustrates the visual difference of the classification and regression problems.

From the model, one needs to know whether the predicted value is good enough as the true label. For this, a loss function is needed, which measures the error size of the predicted value and the actual value. The error of a supervised learning model refers to the difference between the predicted output and the true output labels for a given input example. This error is typically measured using a loss function, which calculates a numeric value representing the magnitude of the error.

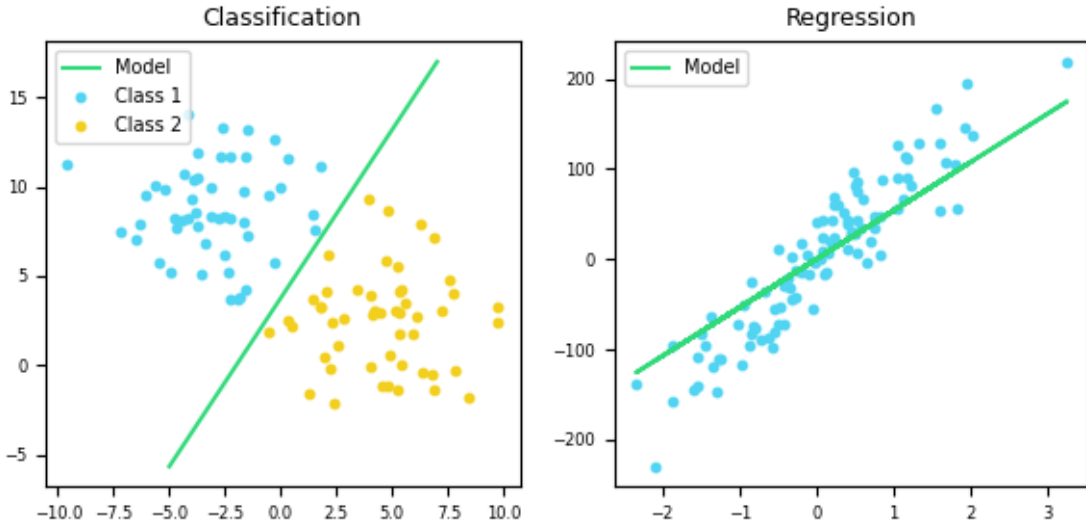


Figure 2 – Example of classification and regression models

2.2 The Regression Problem

Regression is about using input data to predict an actual continuous value. It is a common task in machine learning that can be applied in many cases. Suppose it is needed to calculate the rent of a house from the size of house and number of bathrooms. This is a regression problem because the output, or the rent, is a continuous value. In order to solve this problem, we could gather a dataset of houses with their corresponding sizes, number of bathrooms, and rents. Using the previous equation 2.1, X would be the size and number of bathrooms of the house, and Y would be the predicted rent.

Since the regression model predicts a real value, it is unrealistic to expect the model to predict exactly with a real number label, as it is expected from classification. Thus, the error measure between these two tasks is different. In regression, the error measure is based on the difference between the predicted actual value and a correct actual value. (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018)

One of the error functions used in regression problems is the Mean Absolute Error (MAE) seen in Equation 2.2, where y is the true label, \hat{y} is the predicted label, and n is the number of samples in the training data set. It calculates the average absolute difference between the predicted output and the ground-truth.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.2)$$

Another error function often used in regression problems is the Root Mean Squared Error (RMSE), which calculates the squared root of another metric Mean Squared Error (MSE). The MSE is calculated as the average squared difference between the predicted values and the true values. The Equation 2.3 shows the RMSE, where y is the true label,

\hat{y} is the predicted label, and n is the number of samples in the training dataset.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.3)$$

The RMSE is more sensitive to outliers than the MAE. This means that the RMSE will be larger than the MAE if there are outliers present in the data. Furthermore the RMSE is more affected by the magnitude of the error, while the MAE is more affected by the frequency of the error. This means that the RMSE will be larger than the MAE if there are a few large errors, even if the number of errors is small.

The Mean Absolute Percentage Error (MAPE) is a metric used to evaluate the performance of a machine learning model when predicting a continuous value, which the Equation is seen in 2.4, where n is the number of data points, y_i is the true value, in the i th points, and \hat{y} is the predicted value. The first step of MAPE is calculate the absolute percentage error for each data point, which is done taking the absolute value of the difference between the true value and the predicted value, and dividing this value by the true value. The MAPE is then calculated summing the absolute percentage errors for all data points, and dividing by the total number of data points.

$$MAPE = \frac{1}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.4)$$

2.3 Regressors

To solve regression problems, several methods have been proposed in the literature over time. This section will describe some of these methods that are widely used.

2.3.1 Linear Regression

One of the simplest and most common regression methods is Linear Regression (LR). It assumes that the relationship between the features and the target is almost linear, and has records that go back to the beginning of the 19th century ([LEGENDRE, 1805](#)). The model is responsible for describing how features can be transformed into an output forecast.

$$\hat{y} = Xw + b \quad (2.5)$$

A basic linear model can be described in the Equation 2.5. The X is the feature input matrix, the w the model weights vector, which determines the influence of each feature on the forecast. The b is the bias and determines the value of the prediction when

all features are equal to 0, and the \hat{y} is the output of the prediction, belonging to the set of real numbers (ZHANG et al., 2021). With this model, the objective of linear regression is to find the vector of weights and biases that best predicts the output vector \hat{y} , with the smallest possible error compared to the actual output of the dataset.

2.3.2 Partial Least Square

Partial Least Square is a regression method that uses dimensionality reduction technique. The aim is predicting a set of dependent variables from a set of independent variables, which is performed as follows: PLS create a set of “latent” variables that are linear combinations of the original predictor variables, and then regress the response variable onto these latent variables. PLS has similarities to PCA in dimensionality reduction. In PLS, we have the decomposition of input X and output Y . The difference is that in PCA only the input is decomposed. (ABDI, 2010; BHATTACHARYA, 2005)

$$X = TP + E \quad (2.6)$$

$$Y = TP + F \quad (2.7)$$

As can be seen in the Equations 2.6 and 2.7, both input and output are decomposed, where the E and F are residual matrices, the T is the variable latent matrix, correspondent to the principal component in the PCA, and the P means the loading matrix. (ZHANG; KANO; MATSUZAKI, 2019a)

2.3.3 Random Forest

Random Forest (RF) is a machine learning model proposed by Breiman (2001). This method is a ensemble of decision trees that works training this decision trees on random subsets of the training data and then averaging the prediction of all the trees.

The steps for the random forest training are as follows:

1. A random sample of the training data is selected. This sample is used to train each individual decision tree.
2. For each decision tree, a random subset of the features is selected. This ensures that each decision tree is trained on a different subset of the features, which helps to reduce overfitting.
3. A decision tree is trained on the selected sample of training data and features.
4. Steps 1-3 are repeated for the desired number of decision trees.

5. To make a prediction for a new sample, the sample is passed through each of the trained decision trees and their predictions are averaged.

2.3.4 EBM

Explainable Boosting Machine is a machine learning technique created by [Nori et al. \(2019\)](#), used for regression and classification tasks and based on a Generalized Additive Model. The EBM are designed to be interpretable, providing a clear explanation of how the model is making its predictions.

$$g(E[y]) = \beta_0 + \sum f_j(x_j) + \sum f_{i,j}(x_j, x_i) \quad (2.8)$$

The method equation can be seen in the Equation 2.8, where the g is the link function what says if is a classification or a regression, β is the interception, f_j is a feature function that operates in each one of the features, and $f_{i,j}$ is the function that uses iteration terms, two variables at once. ([NORI et al., 2021](#)).

2.3.5 LGBM

The Light Gradient Boosting Machine (LGBM) is a Gradient Boosting Machine (GBM) method based on decision trees, which use histogram methods and the tree lead-wise growth approach. This method was proposed by [Ke et al. \(2017\)](#) and is an tool to regression and classification, using fewer memory. This method has two alterations from GBM, which are the exclusive features bundling (EFB) that can use a big number of input features without overfitting the model, and the gradient-based one-side sampling (GOSS) which works well with big datasets.

$$\hat{V}_j(d) = \frac{1}{n} \left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i \right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i \right)^2}{n_r^j(d)} \right) \quad (2.9)$$

The equation of the GOSS can be seen in Equation 2.9, where the $A_l = x_i \in A : x_{ij} \leq d$ and $A_r = x_i \in A : x_{ij} > d$, and $B_l = x_i \in B : x_{ij} \leq d$, and $B_r = x_i \in B : x_{ij} > d$.

2.3.6 XGBoost

XGBoost is a machine learning method based on Gradient Boosting Machine proposed by [Chen e Guestrin \(2016\)](#). The method has a reinforcement technique, combining the prediction of weak learners to create a strong learner. This is done by minimizing the objective functions, allowing the prediction and regularization terms to be combined to make the process as fast as possible ([NORI et al., 2019](#)). To correct weak learners, the first

learner receives the entire data set input, in the next the second learner receives the residuals. That is, it is a set that adds new learners to adjust the existing error (OGUNLEYE; WANG, 2020).

$$\mathcal{L}_{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2.10)$$

The equation of the objective function of XGBoost, which contains the regularization and the loss function can be seen in Equation 2.10, where the $\Omega(f_t)$ is the penalty and l is a function that measures the difference between y_i and \hat{y}_i .

2.3.7 KNN

K-Nearest Neighbors is a simple machine learning method of classification and regression. The algorithm was firstly introduced in 1951 by Evelyn Fix and Hodges. To find the next neighbors, KNN uses a measure of distance, such as Euclidean distance. Subsequently, the method makes a prediction with the average value of the neighbors. KNN is called lazy learning because it has no explicit training, storing all training data and making predictions only when it receives a new sample.

$$\hat{y} = \frac{1}{k} \sum y_i \quad (2.11)$$

The Equation for KNN used in regression problems can be seen in 2.11, where y is the output predicted, k is the number of nearest neighbors, and y_i is the output for each of the k nearest neighbors.

2.3.8 SVR

Support Vector Regression is a famous machine learning method. The goal of the method is to find a hyperplane to separate the data into classes, maximizing the distance between the closest points. The distance between the hyperplane and the first point of each class is called the margin. This method doesn't work very well with noisy data.

$$\text{minimize} \quad \frac{1}{2} \|w\|^2 \quad (2.12)$$

$$\text{subject to} \quad y_i - wx_i - b \leq \epsilon \quad (2.13)$$

The point of SVR is realize a dual optimization problem with the Equation 2.12 and 2.13, where the w is the parameter, b is the bias term of the hyperplane equation, $\|w\|^2$ is the Euclidean norm for the w length, and the ϵ is the maximum deviation from the target data y_i . (SMOLA; SCHÖLKOPF, 2004)

2.3.9 MLP

Neural networks are a type of machine learning algorithm that are modeled after the structure and function of the human brain. They consist of interconnected nodes, which are similar to neurons, that process and transmit information. The first neural network was called a perceptron and was introduced by [Rosenblatt \(1958\)](#). There are several types of neural networks, including feedforward networks, which only process information in one direction, from input to output. The Multilayer Perceptron is a type of feedforward network ([CINAR, 2020](#)) that is made up of multiple layers, including at least one input layer, one output layer, and one or more hidden layers.

$$y = f(W_2 * f(W_1 * x + b_1) + b_2) \quad (2.14)$$

For example, given a MLP with an input layer x , a hidden layer h , and an output layer y , this neural network can be written as in the Equation 2.14, where W_1 and W_2 are weight matrices that control the connections between the layers, b_1 and b_2 are bias vectors of the hidden layer and output layer, respectively, and f is the activation function.

2.4 Blast Furnace Description

The process of producing hot metal in a blast furnace can be described as follows. In the beginning, raw materials such as iron ore, sinter (a type of agglomerated ore), and a fuel named metallurgical coke (a fuel made from coal that is used to provide the necessary heat and carbon for the smelting process) are loaded into the upper part of the furnace, the throat. At the bottom, hot air enriched with oxygen alongside with fuels - such as pulverized coal, oil, and natural gas - is injected into the furnace by multiple tuyeres. The tuyeres are used to blow the hot blast into the furnace, providing the necessary heat for the smelting process. The preheated blast at temperatures above 1000°C chemically reacts with the descending loads of coke and injected fuels, producing a large amount of reducing gases (mainly carbon monoxide), reaching flame temperatures of the order of 2000°C. This high temperature is necessary to melt the iron ore and produce hot metal. The carbon monoxide is a reducing agent that removes the oxygen from the ore. The hot metal is then poured out of the furnace through the tap hole at the bottom. The slag (a byproduct of the smelting process) is poured out through a separate hole. The hot metal and slag are cooled and solidified, and the hot metal is then processed further to produce various steel products.

The Blast Furnace is a component responsible to make the hot metal from the iron ore. The blast furnace components include the throat, stack, belly, bosh and hearth, which can be seen in the Figure 3 ([GEERDES; CHAIGNEAU; LINGIARDI, 2020](#)). The

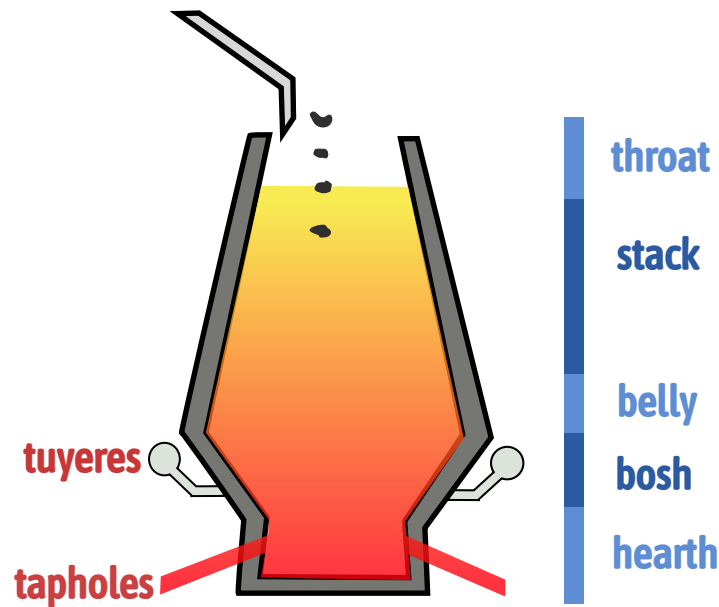


Figure 3 – Blast furnace components

throat is the upper part of the blast furnace where raw materials are loaded. It is located above the stack and below the furnace top. The stack is the vertical cylinder that forms the main body of the blast furnace. It is lined with refractory bricks to protect it from the high temperatures and chemical reactions inside the furnace. The belly is the intermediate part of the blast furnace, located between the stack and the bosh. It is where the most chemical reactions between the hot blast, injected fuels, and raw materials take place, and where the hot metal is produced. The bosh is the lower part of the blast furnace, located below the belly. It is where the slag is produced and separated from the hot metal. The hearth is the lowest part of the blast furnace, located below the bosh. It is where the hot metal is collected and poured out of the furnace through the slag hole. The hearth is also where the furnace tap hole is located, through which the hot metal and slag are removed from the furnace.

The result of this process is the reduction of oxidized iron and melting of slag and metallic iron, which accumulates in the hearth of the blast furnace. The slag is generated by unreduced impurities and it is on top of the hot metal. The hot metal removal process is done along with the slag removal. The hearth of the blast furnace has tap holes used to periodically draw off the hot metal and the slag for the following steps of the iron-making process (CARDOSO; FELICE; BAPTISTA, 2022; WANG; HU; TANG, 2021; ZHOU et al., 2017). In a blast furnace with four tap holes (1, 2, 3, and 4), such in the case that was used in this dissertation, they would usually be circularly arranged at the base of the furnace, with the pairs (1,3) and (2,4) orthogonal to each other, such as can be seen at the Figure 4. The next subsections describe the blast furnace that was used in this

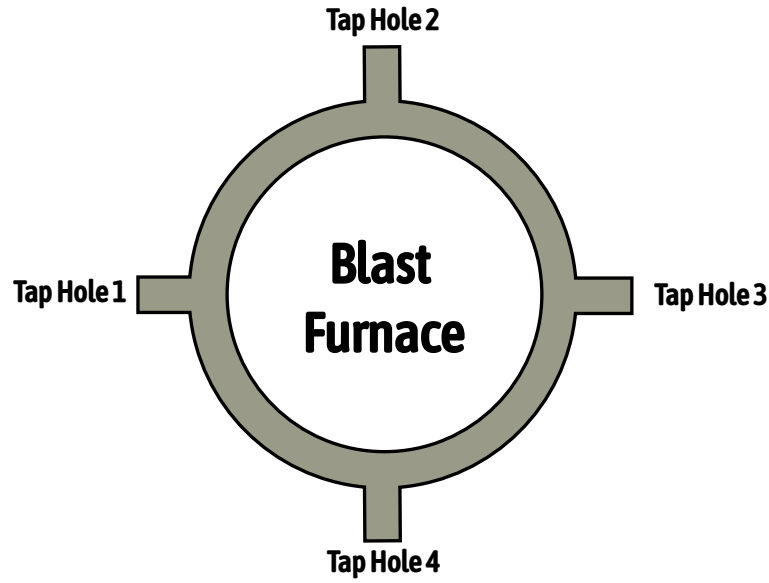


Figure 4 – Tap holes distribution in a Blast Furnace with four tap holes

dissertation, and may not describe a generic furnace.

2.4.1 Temperature Measurement

The hot metal is extracted in a process called *casting*, in which one of the tap holes is opened so that the hot metal flows through the casting channels. The castings are performed alternating between opposite tap holes, meaning that several castings are performed alternating between 1 and 3, and later several castings are performed alternating between 2 and 4. For example, in our dataset, a single casting lasts on average $02:49 \pm 00:31$ hours, with an average interval between the last measurement of a casting and the first measurement of the next one of $01:14 \pm 00:23$ hours.

Hot metal temperature is usually measured through a manual process, in which an operator puts a sensor inside the flowing hot metal every 30 minutes. The first temperature measurement of the casting is performed after a stabilization period that lasts for around one hour from the tap hole opening. As a result, a single casting outputs several temperature measurements (on average 5.87 ± 1.87 measurements) which are stored in the blast furnace system. These temperatures are analyzed by a specialist in order to change the action plan of the blast furnace parameters to keep the temperature between the desired range of 1480°C to 1520°C .

2.4.2 Blast Furnace Variables

Several variables may describe or affect the HMT in a blast furnace. They can be categorized into two types: control variables, those that can be changed by the operator

(e.g., a load of material to be added to the furnace); and sensors variables, whose values come from sensors installed in the blast furnace. The data associated with these variables is stored in the blast furnace system with sampling frequency ranging from 1 to 10 minutes.

The control variables can be load variables, which are related to the charge injected into the top of the furnace, such as dry coke, which is the charge in tons of dry coke that is placed in the furnace, silicon charge, ore and others. And the variables related to the actions of the furnace, such as water flow in the stove, temporal variation of skin flow temperature, top pressure, pulverized coal injection, among others.

Each of the control variables has a different expected time to affect the blast furnace temperature. That is, when increasing the load at a given moment, it is only possible to see the change referring to it after a certain time. In Table 1 it is possible to see the time that each control variable takes to change the temperature, with the data received by the furnace operator.

Table 1 – Expected Time to affect the Temperature

Variables	Interference Time
Blast Flow Rate	30 min
Blast Moisture	30 min
Blast Temperature	30 min
PCI	1 hour
Top Pressure	0 hour
Top Gas Analysis (H^2)	30 min to 1 hour
Top Gas Analysis (N^2)	30 min to 1 hour
Top Gas Analysis (CO)	30 min to 1 hour
Blast flow Rate	30 min
Direct reduction	1 hour
HO Thermal Index	1 hour

The sensors variables describe the actual state of the furnace, giving the operator information about how the furnace is at that moment. Some examples of sensors variables are the CO/CO₂ ratio, the permeability index, direct reduction, the thermal loss of the stove, among others. These variables are used by the operator to take actions.

2.5 Literature Revision

In the last decades, many works have been proposed using machine learning methods to improve the control of the furnace and produce hot metal with high quality and low production cost. These methods include regressive models, support vector regression

methods, artificial neural networks, and other artificial intelligence models. In these data-driven models, unlike in physics-based simulations, little to no prior information about the process is required, given that it is their goal to detect patterns in the data that could potentially describe the dynamic behavior of the studied systems. The work of Lin et al. (LIN et al., 2011) used models based on partial least-squares (PLS) and principal component analysis (PCA) regression for predicting silicon in the hot metal. Ding et al. (DING et al., 2018) used a long short term memory network (LSTM) and a recurrent neural network to solve the same task of silicon prediction in the hot metal. Zhou et al. (ZHOU et al., 2017), proposed a nonlinear autoregressive exogenous model that uses multioutput least squares support vector regression for the task of multivariable prediction of the hot metal quality indexes, namely the hot metal temperature, silicon content, sulfur content and phosphorus content of the hot metal. In Azadi et al. (AZADI et al., 2022), a hybrid modeling framework is proposed for the prediction of hot metal silicon content and slag basicity in industrial-scale blast furnaces.

Other works (ZHANG; KANO; MATSUZAKI, 2019b) proposed an ensemble pattern trees model for the task of hot metal temperature prediction. For the same HMT prediction task, Su et al. (SU et al., 2019) proposed an ensemble model based on multi layer extreme learning machine and strategies using the adaptive particle swarm optimization. Hashimoto et al. (HASHIMOTO; SAWA; KANO, 2019) developed a model to predict HMT by adopting moving horizon estimation based on a one-dimensional transient model. Zhang et al. (ZHANG; KANO; MATSUZAKI, 2019a) provide a comparative study between some shallow and deep methods for prediction of hot metal temperature, including, among others, PLS, SVR, and neural networks. Hashimoto et al. (HASHIMOTO et al., 2022) proposed a non-linear model predictive control (NMPC) to predict the hot metal temperature 10 hours ahead and indicate the controlling action for a variable called pulverized coal rate (PCR).

In general, times series analysis assumes regularly-sampled (over time) data. Considering the irregularity of our data, traditional methods for times series analysis were not included in the analysis. Moreover, there is a relevant discussion on the use of Deep Learning for hot metal temperature prediction (ZHANG; KANO; MATSUZAKI, 2019a). The authors suggest that deep models are more suitable for short-term prediction, whereas shallow models yield better performance on multi-steps ahead (long term) prediction. Since we are also interested in the latter scenario, deep models were not considered in this work.

3 Temperature Prediction System

This section presents the description of the temperature prediction system and its components, and organized as follows: Section 3.1 presents the overview of the system, Section 3.2 presents the input variables used in the models, Section 3.3 the modeling using absolute temperatures and Section 3.4 the modeling of the system using projected temperatures. At the end, the pre-processing of the system is described in Section 3.5 and the prediction process in Section 3.6.

3.1 Overview

The objective of the system is to predict the hot metal temperature with several input variables coming from the blast furnace and make the result available to the operator. The inputs for the prediction model comprise the control variables, sensors variables, and additional information related to the temperature measurements and the elapsed time between the prediction time and the measurement times. An overview of the proposed system considering is shown in Figure 5.

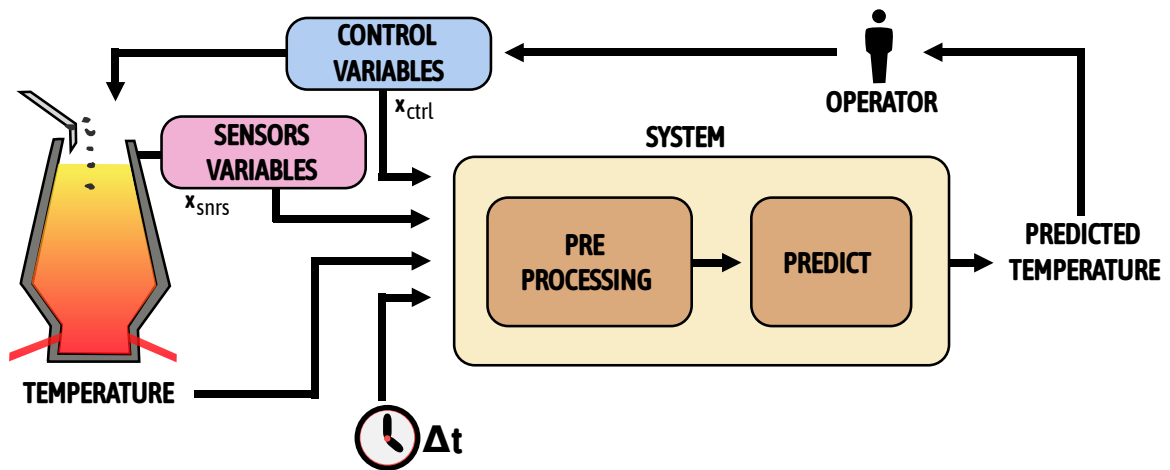


Figure 5 – Modules and flow of information. The system receives a set of control variables, temperatures and delta times which are preprocessed and used to predict a temperature in the future

The current application of the system takes place at the present time (designated as t_0) to predict a temperature in the future (prediction time, t_p) based on recent history and a action plan on the control variables from t_0 to t_p . The input data, comprising past and future values of the control variables (in some experiments, the past values from sensors variables) and information about the last measured temperatures, are firstly preprocessed

and then used by a machine learning algorithm (regressor) to predict the temperature. These variables are better explained later in Figure 7.

With the prediction available, the operator can properly update the action plan of the furnace (control variables). It is worth noting that this work focuses on studying the potential of the prediction system, therefore, the action plan is assumed to be known.

3.2 Input Variables

The input variables of the model comprise a subset of furnace control variables, sensors variables, temperature measurements, and elapsed times related to relevant furnace events (Figure 7) which are described below.

3.2.1 Control Variables

Ten control variables were initially identified by an experienced operator as those with greater potential to change the furnace temperature, which can be seen in the Table 2 with the description of each variable.

Table 2 – Control variables used as input in the system and description of each of them

Variables	Description
Blast volume	Volume of air + enriched O^2 injected inside the furnace through the tuyeres.
Oxygen enrichment	Amount of oxygen added to the air on the blast
Blast moisture	Total moisture on the blast (added steam + atmospheric)
Blast temperature	Air + O^2 mixture temperature before the tuyeres
Pulverized coal injection	Amount of coal being injected at the tuyeres
Top pressure	Regulated pressure at the top of the furnace
Dry coke	Amount of coke charged by the top disregarding its moisture content
Coke rate	Amount of consumed coke per ton of produced hot metal (kg/tonHMT)
Ore rate	Amount of ore used per ton of produced hot metal (kg/tonHMT)
Ore	Amount of ore charged by the top

Recall that the proposed system receives updates about the state of the blast furnace variables every 30 minutes. The state of the chosen control variables is synchronized but the temperature measurements are not. The time it takes for each control variable to have an impact on the hot metal temperature is different. Often, some variables only start influencing the temperature after an hour or two, while others take effect instantly. To predict the change in the temperature from the last measurement, \mathbf{x}_{ctrl} carries a window of n hours of the ten control variables in time. Moreover, other input variable related to control variables, Δt_{ctrl} , is used to inform the system of the elapsed time from the last supposed update of the control variables to the time of prediction.

The control variables are used as input in all modelings. They are essential and allow a more flexible system because, as intended for real/online operation of the system, the control variables can be replaced by a action plan ranging from the current time to the time of prediction.

3.2.2 Sensors Variables

Three sensors variables were chosen by the furnace operator who uses them to control the blast furnace, which can be seen in the Table 3 with the description of each variable. Like the control variables, these variables are reported to the system every 30 minutes.

Table 3 – Sensor variables used as input in the system and description of each of them

Variables	Description
Direct reduction	Calculated direct reduction percentual via heat, oxygen, carbon and iron balances
Thermal Index $H0$	Proprietary thermal control index
CO^2 composition at the top	Continuous CO^2 measurement via chromatography at the top

Sensors variables were not included as input variables throughout the modeling, being used as a separate experiment, in a different approach. This happens because unlike the control variables, in an online operation of the system, the sensor variables cannot be replaced by a control plane, as it is not possible to assume what the state of the furnace will be like.

3.2.3 Summarization

Since the temperature measurement process is noisy and the sampling rate of the blast furnace variables is higher than the expected temperature response, the data is summarized to enable the later online use of the system. The data summarization process is described in the following.

For the hot metal temperature, the several measurements per casting are summarized into a single measurement to help removing noise. In the beginning, the first measurement of each casting is removed since it is usually noisier than the others, because many times the channel that runs the hot metal is still cold in these first measurements. Subsequently, for each casting, the highest temperature is chosen as the representative of the respective casting. Then, the data associated with the blast furnace variables is summarized in intervals of 30 minutes considering the average value of the past 30 minutes.

The summary of a casting can be seen in Figure 6, in a run with four measurements, where T_{max} is the maximum temperature of the run, and t_{max} is the respective time of that measured. The first measurement in that casting is a temperature higher than the chosen

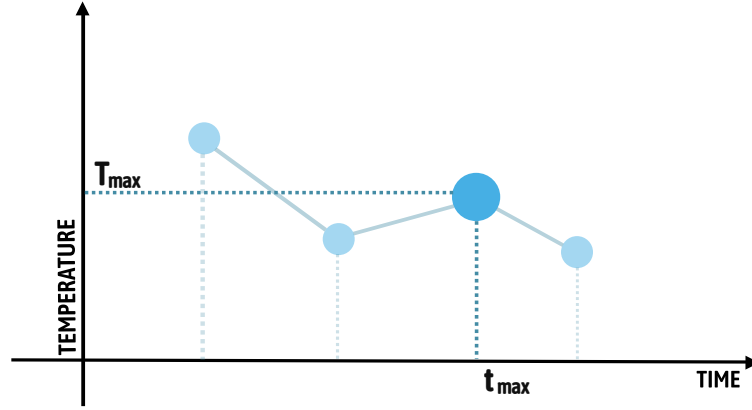


Figure 6 – Illustrates a casting with four measurements with the maximum temperature of that casting marked

maximum temperature; However because it is the first of a casting, it is disregarded.

3.3 Model with Absolute Temperatures

The modeling with absolute temperatures uses measurement temperature variables later than current. As the tap holes behave independently, when predicting a temperature from a certain X tap hole, if the last temperature measurement comes from a different tap hole, an Y tap hole (which occurs in 99,5% of the times) information about the last recorded temperature coming from that same tap hole is important.

The elapsed time of the last measurement is also crucial information. Without this elapsed time, the input having a temperature from 25 minutes ago and 3 hours ago would be erroneously the same. Without knowing the elapsed time of these measurements, it would be impossible to know which measurement is more recent, and, thus, which one would be more relevant for the prediction. Therefore, these temperatures and elapsed times are needed for the prediction.

Therefore, the inputs in the absolute temperatures model are: T_{last} (*last* denotes the last temperature measurement considering the current time) and T_{same} (*same* denotes the last temperature measurement of the same tap hole considering the current time), and by Δt_{last} and Δt_{same} , the respective elapsed times from such measurements to the time of desired prediction. The measurements are illustrated in Figure 7.

Thus, for a particular temperature prediction at t_p , the model receives as input an updated vector $\mathbf{x} = (\mathbf{x}_{ctrl}, \mathbf{T}, \Delta \mathbf{t})$, where \mathbf{x}_{ctrl} is the current state of the control variables window, $\mathbf{T} = (T_{last}, T_{same})$ and $\Delta \mathbf{t} = (\Delta t_{ctrl}, \Delta t_{last}, \Delta t_{same})$. Using the sensors variables, the vector \mathbf{x} also contains \mathbf{x}_{sns} , which are the current state of the sensors variables window.

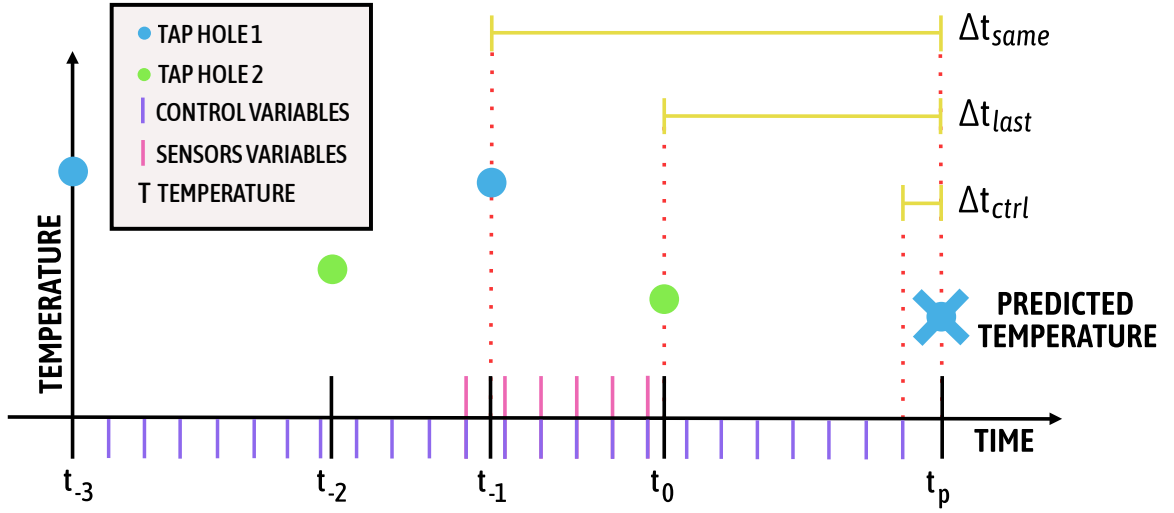


Figure 7 – Illustrates, in time, the input and output information of a prediction at t_p with the user at time t_0 considering information of n hours prior t_p .

As in the approach using sensors variables it is not possible to replace values in online mode as control variables, only variables prior to t_0 were used.

3.4 Model with Projected Temperatures

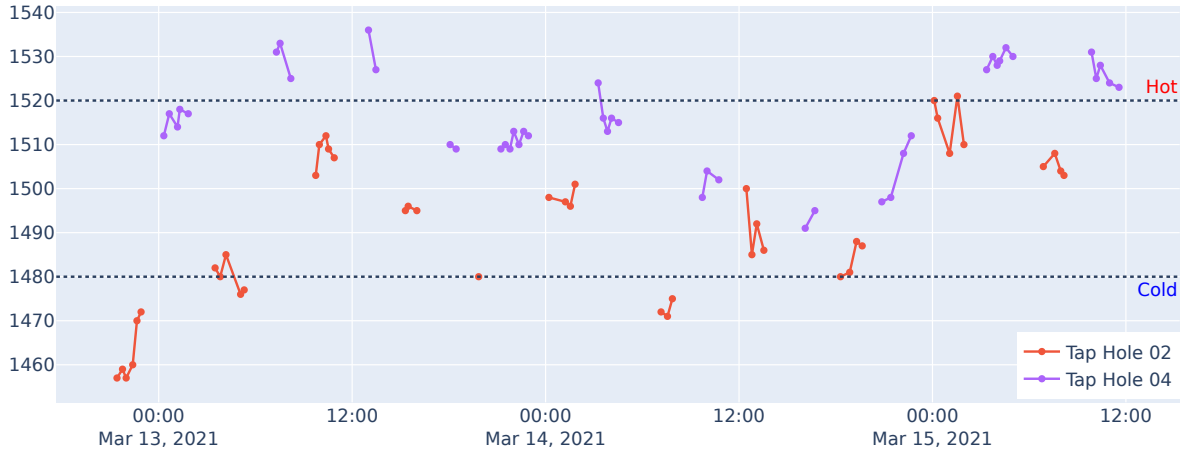


Figure 8 – The temperature difference in two tap holes (Tap Hole 2 and 4)

Despite belonging to the same blast furnace, the tap holes behave differently within the same time interval. As can be seen in Figures 8 and 9, there is a gap in measurement temperatures between two interleaved tap holes. Given this difference, predicting the temperature of a given hole using the last measured temperature, which belongs to another hole, can negatively influence the forecast.

Aiming at this problem, this model was built, which generates an artificially

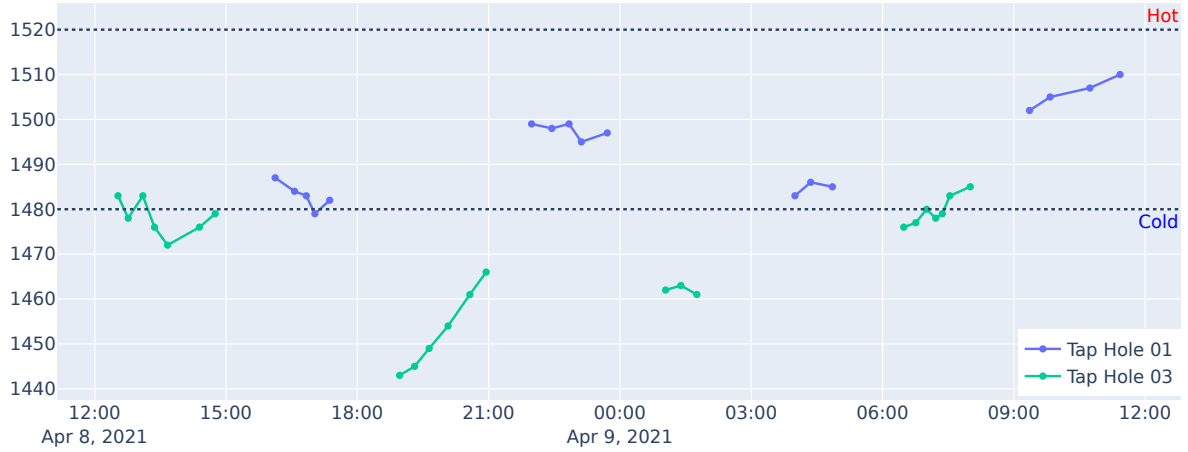
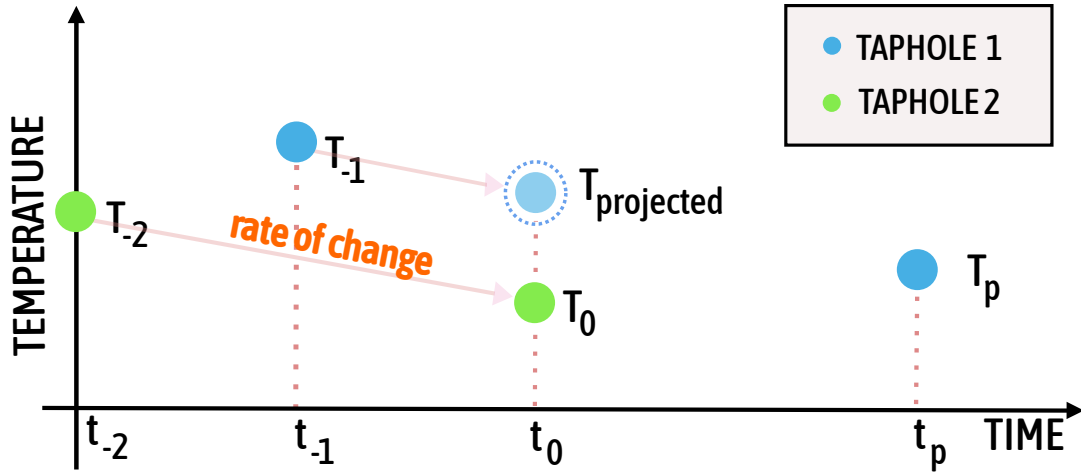


Figure 9 – The temperature difference in two tap holes (Tap Hole 1 and 3)

Figure 10 – Illustrates the generation of the $T_{projected}$

projected temperature to be used as input, $T_{projected}$. To generate this temperature, which can be seen in Figure 10, we obtain the rate of change between measurements T_0 and the previous temperature of the same tap hole (in this case, T_{-2}), and their respective time lapses. From this rate of change, a new T_0 value is projected from T_{-1} , the same hole as T_p . In 95.7% of the time, T_0 and T_{-2} correspond to the same tap hole. The equation that describes this process is:

$$T_{projected} = \frac{T_0 - T_{-2}}{t_0 - t_{-2}} * (t_0 - t_{-1}) + T_{-1} \quad (3.1)$$

This value is projected at the same position as the last measurement t_0 . In this way, the value of the last measurement stops feeding the model, which starts to receive the projected value. As with absolute temperature modeling, the elapsed time between the last temperature measurement and the current temperature is also used as input. That is, the $\Delta t_{projected}$ is equals to Δt_{last} .

To predict a certain point, the model with the projected temperature receives the $\mathbf{x} = (\mathbf{x}_{ctrl}, \mathbf{T}, \Delta t)$ vector as input in the same way as in the absolute temperature model (with the \mathbf{x}_{sns} in the sensors variables approach), but with the $\mathbf{T} = (T_{projected}, T_{same})$ and $\Delta t = (\Delta t_{ctrl}, \Delta t_{projected}, \Delta t_{same})$ vector in this way. As in the approach using sensors variables it is not possible to replace values in online mode as control variables, only variables prior to $t_{projected}$ were used.

3.5 Preprocessing

The system was evaluated with real-world data collected from a blast furnace of a large steelmaking company. Prior to use, the data was cleaned and preprocessed as follows next.

3.5.1 Noise Removal

Noisy temperatures with clearly low values (below $100^{\circ}C$) were removed. Castings having temperatures with same timestamp were removed. Then, we removed some runs that had all of their measurements associated with same timestamp in their system. This situation should never happen and it is probably caused by a problem in their system, which makes these runs unreliable.

The shorter casting was removed in case of overlapping castings. Although this overlap can happen in practice (albeit most of them were due to problems with the logging system), the shorter one was removed because it was deemed less reliable and would yield a lower impact on the raw data. Moreover, as the HMT measurement process is noisy in itself, sometimes an operator notices that the temperature measurement is likely to have been wrongly taken, and retakes it. Usually, this second measurement happens shortly after (in an interval of less than 5 minutes). Therefore, measurements that had a subsequent one within this interval were also removed. Besides, the first HMT measurement of each casting was removed because it usually presents a lower signal-to-noise ratio.

Continuing the cleaning process, the data collected close to a maintenance downtime ($\pm 8h$) were disregarded because the blast furnace may be in an unusual state preparing to or recovering from the maintenance. Furthermore, control variables with values outside their operational range (provided by an expert) were considered outliers and had their values truncated.

3.5.2 Normalization

After the cleaning, the data was preprocessed in a two-fold manner: data normalization and data imputation. To normalize the data, we employed an IQR-based normalization,

which is basically the subtraction of the median and division by the interquartile range (IQR).

Control variables were normalized by mean and IQR of each variable separately. As for the temperature and delta variables, these were normalized differently. The deltas have been normalized together. Temperatures were normalized using the median and IQR of one variable, the run maximum temperature. All data was normalized using the normalization parameters from training data. There was no contamination of the test set with the training data.

Finally, some control and sensors variables may have missing values due to various reasons (e.g., sensor malfunctioning, system and networking issues, database failures). As most of the models evaluated in this work do not support working with missing data, we imputed values according to the moving average using a fixed window of the past 4 hours. After the data imputation, the data is ready for use.

3.6 Prediction

In order to perform a prediction with a system, the user defines the interval (in the future) when the casting is meant to happen. Within this interval, predictions are performed every 30 minutes approximately, and the maximum predicted temperature among them is used for that casting.

In the online system, we do not have the control variables until the prediction, the values are replaced by a control plan, provided by the furnace operator with their next steps. But, in an offline approach, all control variables are available; Therefore, the substitution of future variables to a control plan is not required.

4 Experimental Methodology

In order to evaluate the system, a set of experiments were performed with real-world data from a blast furnace. The evaluating of the experiments was focused on two modelings, with absolute temperatures and projected temperatures, and with and without the sensors variables. In Section 4.1, the dataset is described, while Section 4.2 covers the hyper-parameter search. The evaluation metrics are discussed in Section 4.3, followed by a detailed description of the experiments themselves in Section 4.4. Finally, the baselines used are presented in Section 4.5.

4.1 Dataset

The proposed system was evaluated with real-world data from the industry. The preprocessed dataset comprises 79,764 records of sensor data collected along the years 2017 to 2021. We split data into training (47,416 records), validation (16,968 records), and testing (15,380 records) sets. The training set comprises data collected from the years 2017, 2018, and 2019, the validation set includes only the year 2020, and the test set includes the year 2021 (more recent data). The data was actually collected and stored in the company systems already considering the synchronization in time (as described in Section 2).

4.2 Hyper-Parameter Search

To provide a fair comparison study, for each of the evaluated regressors, we performed an extensive hyper-parameter search using the Tree-structured Parzen Estimator (BERGSTRA et al., 2011), and the best model was chosen as that which maximizes the MAE on the validation set. For each regressor, a hyper-parameter search was performed to find the best combination in each case. The methods were optimized using the MAE metric as a guide for the best result on the validation set. The search for hyperparameters took place around the default values of each library. The default value and chosen range are listed below. Each method’s default value listed in the library was considered when selecting the intervals and ranges. The investigated parameters and their respective domains are mentioned in the following (real ranges are expressed as $[\cdot]_{\mathbb{R}}$):

- EBM¹:
 - Max Bins - Default: 256 - Range: $\in [128, 512]$;

¹ Library used: <https://interpret.ml/docs/ebm.html>

- Max Interactions - Default: 32 - Range: $\in [16, 64]$;
- Number of Interactions - Default: 10 - Range: $\in [5, 15]$;
- External Pockets - Range: $\in [4, 16]$;
- Number of Trials: 30.
- KNN²
 - Number of neighbors - Default: 5 - Range: $\in [3, 100]$;
 - Number of Trials: 25.
- LGBM³
 - Boosting Type - Default: gbdt - Range: $\in \{\text{gbdt}, \text{dart}, \text{goss}\}$;
 - Number of Estimators - Default: 100 - Range: $\in [3, 1000]$;
 - Number of Leaves - Default: 31 - Range: $\in [25, 100]$;
 - Number of Trials: 15.
- PLS²
 - Number of Components - Default: 2 - Range: $\in [1, 100]$;
 - Number of Trials: 100.
- Linear Regression²
 - Intercept Adjustment - Default: True - Range: $\in \{\text{True}, \text{False}\}$;
 - Number of Trials: 2
- MLP²
 - Number of Neurons - Default: 100 - Range: $\in [2^5, 2^7]$;
 - Number of Layers - Default: 1 - Range: $\in [2, 5]$;
 - Initial Learning Rate - Default: 0.0001 - Range: $\in [0.0001, 0.01]_{\mathbb{R}}$;
 - Activation Function - Default: ReLU - Range: $\in \{\text{ReLU}, \text{tanh}\}$;
 - Maximum Number of Iterations - Default: 200 - Range: $\in [100, 1000]$;
 - Number of Trials: 15.
- Random Forest²
 - Number of Estimators - Default: 100 - Range: $\in [3, 1000]$;

² Library used: <https://scikit-learn.org/>

³ Library used: <https://lightgbm.readthedocs.io/>

- Function to measure the quality of a split - Default: MSE - Range: $\in \{\text{MSE}, \text{Friedman MSE}\}$
- Function that limits the Number of features in split decisions - Default: None - Range: $\in \{\text{sqrt}, \text{log}\}$;
- Number of Trials: 100.
- SVM²
 - Number of Estimators - Default: 10 - Range: $\in [10, 15]$;
 - Number of Trials: 10.
- XGBoost⁴
 - Number of Estimators $\in [3, 1000]$;
 - Loss function - Default: MSE - Range: $\in \{\text{MSE}, \text{pseudo-huber}\}$;
 - Number of Trials: 30.

4.3 Evaluation Metrics

Five metrics were used to evaluate the performance of the model on real-world data. Three of them are very common in prediction analysis: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The other two address the particularities of this application: accuracy at range (AAR) and a trend metric (TM). AAR quantifies the ratio of predictions that lie within a margin of tolerance ϵ in relation to the ground-truth. Given n_{pred} predictions,

$$\text{AAR}(\epsilon) = \frac{1}{n_{pred}} |\{t : |\hat{T}_t - T_t| \leq \epsilon\}|, \quad (4.1)$$

where \hat{T}_t stands for the prediction at time t and T_t for the respective ground-truth temperature.

While AAR can be seen as a two-class (“hit or not hit”) classification metric, there is the trend metric (TM) that regards three classes: constant, increased, or decreased. These classes (labels) are given by the following function that verifies the trend behavior when moving from a temperature T_{curr} to T_{next} :

$$\mathcal{B}(T_{curr}, T_{next}) = \begin{cases} \text{“increased”}, & \text{if } T_{next} - T_{curr} > \epsilon \\ \text{“decreased”}, & \text{if } T_{next} - T_{curr} < -\epsilon \\ \text{“constant”}, & \text{otherwise.} \end{cases} \quad (4.2)$$

⁴ Library used: <https://xgboost.readthedocs.io/>

Based on this definition, the trend measure is defined as

$$\text{TM}(\epsilon) = \frac{1}{n_{\text{pred}} - 1} |\{t : \mathcal{B}(T_t, \hat{T}_{t+1}) \text{ matches } \mathcal{B}(T_t, T_{t+1})\}|. \quad (4.3)$$

The metric chosen to be used as a standard was the Mean Absolute Error, as it is widely used in regression problems, as in this work. It was used in most analyzes and to optimize parameters in parameter search.

4.4 Experiments

In this work, we perform a comprehensive comparative study of several machine learning regressors: KNN, LR, PLS, EBM, LGBM, RF, SVM, XGBoost, and MLP. Overall, our study comprises tuning these regressors (via a hyper-parameter search on the validation set) in multiple configurations (combinations of the length of the time history for the control variables and which previous temperatures to provide for the model) for three scenarios of interest (predicting one, two and three castings ahead), then evaluating them on a large-scale real-world test set in all these scenarios. Following, we provide details on each of the steps we performed for this study.

4.4.1 Absolute Temperature

This experiment uses absolute temperature modeling, taking as temperature inputs only actual measurements. Three specific experiments was performed in this approach, which can be seen below.

4.4.1.1 History Length

In order to verify the maximum influence of the control variables, and to cover a period of time in which there is no temperature measurement, lagged control variables were given as input, in two possibilities, with 8 hours of delay and 6 hours of delay. In this experiment, the prediction is one (t_{+1}) casting ahead. This analysis will verify which of the two possibilities performed better, and define this parameter for the next experiments.

4.4.1.2 Main Experiment - Predictions Ahead

The models are finally evaluated on three different use cases: HMT prediction one (t_{+1}), two (t_{+2}), and three (t_{+3}) castings ahead, after exhaustively searching the best hyper-parameter for each method in the validation set. As already mentioned, given that the evaluation is on historical data, we assume that the operator followed the only control plan available (i.e., the one present in the data) and predictions were performed synchronously with the real measurements. The best models on the validation set are

evaluated on the test set and their results are reported. For the AAR and TM metrics, we used $\epsilon = 10^\circ\text{C}$. After this experiment, the next ones predict only t_{+1} casting ahead, as this is the central problem to be explored.

4.4.1.3 Custom History Length

Each control variable has a different time of influence on the temperature. The introduction of these control variables at a time when there is no interference in the hot metal temperature could allow the model to learn a nonexistent correlation at that point. That way, a mask was made on the input control variables so that the model includes only from the moment when the beginning of the load affects the temperatures. This experiment will verify the effectiveness of the mask in affecting the result in the right way, compared to the best result without the mask of the T_{t_p-1} experiment.

To carry out this test, at a certain point in the data set an artificial value was inserted in one of the variables. From this, it is analyzed whether the behavior of the custom history length system and contrasted with the full history approach.

4.4.2 Projected Temperature

This experiment uses projected temperature modeling, taking as temperature inputs actual measurements and projected measurements. Two specific experiments was performed in this approach, which can be seen below.

4.4.2.1 Main Experiment

The idea of this experiment is to analyze the predicted temperature modeling, performing a comparative study of each regression model in the same way as the absolute temperature, searching the best parameters. However, in the projected temperature experiments, only the predicted HMT prediction one casting ahead (t_{+1}) is analyzed.

4.4.2.2 Custom History Length

Implemented in the same way as modeling absolute temperatures, this experiment checks whether, with the custom history length, the variables influence the predicted temperature in the expected way.

4.4.3 Sensor Variables

In these experiments, the sensor variables are used as input along with the control variables. As the furnace operator uses these variables to guide the blast furnace, the objective of the experiment is to verify if the system is able to capture this information to aid in prediction. Such as the main experiment of the absolute temperature modeling

and projected temperature modeling, is done a search of the best parameters for each regression model.

This experiment is done for both absolute and projected modeling.

4.5 Baselines

As, to the best of our knowledge, the models published in the literature are not publicly available. Therefore, in addition to the comparison between the several evaluated models, we also provide common baselines: a naive forecast that repeats the last values (in our case, T_{t_p-1} , T_{t_p-2} and T_{t_p-3} being one for each type of prediction t_{+1} , t_{+2} and t_{+3} respectively) as well as moving averages with window size $w = 2$ for each of the scenarios t_{+1} , t_{+2} and t_{+3} . The moving averages $M_{t_p-1}^w$, $M_{t_p-2}^w$ and $M_{t_p-3}^w$ are calculated considering the w past temperature measurements starting from T_{t_p-1} , T_{t_p-2} and T_{t_p-3} for each of the considered scenarios respectively.

4.6 Computational performance

The experiments were executed in a machine with 256GB of memory RAM, with Ubuntu 16.04.5 LTS and CPU Intel(R) Xeon(R) CPU E7-4850 v4 @ 2.10GHz with 128 cores.

5 Results and Discussion

This section presents the results to the experiments described in Section 4 and the study limitations at the end.

5.1 Absolute Temperature

In this section, the results referring to the experiments and analysis using absolute temperatures are presented.

5.1.1 History Length

In this experiment, the results referring to the duration of the history used were reported. As can be seen in Table 4, the experiments that used the 8-hour history achieved a better validation MAE than the experiments that used the 6-hour history. For this experiment, we predict only t_{+1} castings ahead.

Table 4 – History Length Experiment - Validation Result

Method	MAE 6h	MAE 8h
PLS	10.415	10.142
Linear	10.422	10.159
LGBM	11.289	11.275
MLP	12.088	11.804
EBM	12.160	12.219
XGBoost	11.863	12.102
SVM	14.343	14.067
RF	12.029	12.134
KNN	16.784	16.882

The first four best-performing methods, PLS, Linear, LGBM, and MLP, achieved a lower MAE using the 8-hour history, together with the SVM. The other methods, XGBoost, RF, EBM and KNN also worked well, and achieved better results using a 6-hour history.

Unfortunately, it is only possible to conjecture that these methods do not give as much importance to the variables referring to the last lags as to the first methods.

This study aims to define the best value for the parameter history length. The experiment that obtained the best validation result was the one with a history length of

8-hours. Therefore, this value (8-hours) is chosen as the default for the parameter history length in the next experiments.

5.1.2 Predictions Ahead

In this comparative study, we report results for several machine learning methods on three use case scenarios, where the HMT prediction is made one (t_{+1}), two (t_{+2}), and three (t_{+3}) castings ahead. The Linear Regression and PLS methods were the ones that obtained the best results in all scenarios.

The results of this experiment can be seen in the next tables, which each comprise 12 rows, one per method (described in the first column), along with 8 columns showing the results of each metric: MAE, MAPE, and RMSE, and the AAR and TM metrics with 5°C and 10°C tolerances.

In t_{+1} the PLS and Linear tied with MAE of 9.66, as can be seen in Table 5. The AAR(10) shows that in approximately 60% of the cases of t_{+1} scenario the system predicts the correct temperature within 10°C. The TM(10) shows the system follows the trend in approximately 62% of the cases.

Table 5 – Main experiment – Predicting t_{+1} – 8-Hours History Length *Baseline methods

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
PLS	9.66	0.64	12.55	34.17	60.14	59.20	62.49
Linear	9.66	0.65	12.56	34.20	60.03	59.23	62.59
MLP	10.22	0.68	13.22	32.35	58.11	57.55	60.60
LGBM	10.36	0.69	13.35	32.66	57.12	56.07	60.40
EBM	10.76	0.72	13.83	29.57	55.71	55.22	58.79
XGBoost	10.93	0.73	14.05	29.06	54.99	54.53	57.52
SVM	10.97	0.73	14.16	30.40	55.58	57.21	58.65
RF	11.52	0.77	14.78	28.03	52.66	49.76	53.78
$M_{t_p-1}^2$ *	12.61	0.84	16.06	27.38	49.57	44.41	50.03
T_{t-1} *	13.83	0.92	17.55	26.07	45.80	26.15	45.81
KNN	17.03	1.14	21.42	18.73	36.47	47.32	45.16

In the scenario t_{+2} , which can be seen in Table 6, Linear was a little better with MAE of 10.12, followed by PLS with MAE of 10.14. And again, at t_{+3} , PLS was the best with a MAE of 12.33 behind Linear with a MAE of 12.34 (Table 7).

The other regression methods also obtained satisfactory results with values similar to Linear and PLS, with an average MAE of 10.648 in t_{+1} , with the exception of Random

Table 6 – Main experiment – Predicting t_{+2} . – 8-Hours History Length *Baseline methods.

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
Linear	10.12	0.68	13.01	31.25	57.87	58.68	61.08
PLS	10.14	0.68	13.01	30.84	57.74	58.44	60.09
LGBM	11.04	0.74	13.98	27.93	53.38	56.52	57.28
EBM	11.17	0.75	14.24	28.27	53.48	56.21	56.35
SVM	11.52	0.77	14.75	28.41	52.62	57.31	58.07
XGBoost	11.61	0.77	14.79	27.41	51.49	54.98	56.52
MLP	11.79	0.79	15.21	27.79	52.11	55.29	57.48
RF	12.36	0.82	15.68	25.21	49.06	51.85	52.75
T_{t-2}^*	14.91	0.99	19.05	24.29	42.47	47.80	47.98
$M_{t_p-2}^2^*$	16.31	1.09	20.82	21.24	39.45	40.63	42.69
KNN	17.08	1.14	21.46	18.66	35.99	47.22	45.06

Forest and KNN. KNN itself was the worst method in all use cases, the only worst-than-baseline regression model.

Table 7 – Main experiment – Predicting t_{+3} . – 8-Hours History Length *Baseline methods.

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
PLS	12.33	0.82	15.85	25.73	49.91	54.08	54.63
Linear	12.34	0.82	15.86	25.76	49.78	54.26	54.56
SVM	13.20	0.88	16.83	25.25	46.62	52.57	52.88
EBM	13.63	0.91	17.32	23.40	45.35	52.09	50.69
LGBM	13.68	0.91	17.30	23.19	45.01	51.89	51.17
XGBoost	14.50	0.97	18.37	22.16	42.71	49.59	48.80
RF	14.92	0.99	18.84	21.85	41.20	49.14	47.77
MLP	15.39	1.03	19.41	20.21	39.97	49.45	48.94
KNN	17.36	1.16	21.81	18.29	35.30	47.12	44.82
$M_{t_p-3}^2^*$	19.17	1.28	24.65	18.83	35.06	40.84	41.11
T_{t-3}^*	20.10	1.34	25.68	17.74	34.07	35.11	36.41

For the experiments, qualitative results were generated covering four time lapses of ten days, to cover all tap holes and not present biased results of only time lapse. The date interval are: 21 Jan 2021 to 31 Jan 2021, 21 Mar 2021 to 31 Mar 2021, 10 Feb 2021 to 20 Feb 2021 and 15 Apr 2021 to 25 Apr 2021. The results are charts with the X axis is the time, and the Y axis is the hot metal temperature. Qualitative results are graphs where the X-axis is time and the Y-axis is the hot metal temperature. You can see the

ground-truth in blue, the forecast with the Linear model in red, and the best baseline for each use case in green. In addition, two lines that guide whether the temperature is hot (1520°C) and cold (1480°C) are marked on the graph.

The qualitative results of t_{+1} can be seen in the Figure 11. The prediction model trend, in this case the Linear model that achieves the smallest error, follows the the ground-truth trend for almost the entire time interval shown, being more close to ground-truth than the baseline. Close to the January 25, 2021 it is possible to see a large discrepancy. On closer inspection, an experienced operator informed that the ground-truth temperature data was inconsistent with the other measurements, showing, as previously said, that the data set has a lot of noise.

Figure 12 shows the result of the experiment with t_{+2} , using the same time interval shown in Figure 11 above. The model used is Linear, still with the best result in t_{+2} , and follows the trend of the ground-truth and obtains at certain points a temperature closer to the ground-truth than t_{+1} . This is because the model input comprises not the last run measurement, but its predecessor, which has the same tap hole as the predict.

In Figure 13, which shows the experiment t_{+3} , the model errs more when moving away from the ground-truth than in t_{+1} and t_{+2} . But the predict is still inside the guide lines that inform if the temperature is hot or cold, such as can be seen in the figure.

The Figure 14 show the absolute error of the model and baseline compared to the ground-truth, in the same time interval of the Figures 11, 12 and 13. It can be seen that the error is smaller in the t_{+1} but closer to the baseline, whereas in the t_{+3} the error is bigger but the baseline is worst.

The next analysis and experiments maintain a pattern by predicting only t_{+1} , the central problem investigated.

5.1.3 Analysis by Tap Hole

Considering the mismatch between the tap holes, an analysis was performed using a specific test set for each of the tap holes, in order to verify the performance of each of them separately. The Table 8 shows the results with the MAE metric of each of the tap holes. The method used in the analysis was Linear Regression, which achieved a lower MAE in the experiment t_{+1} .

It is possible to notice that the error of experiment using tap hole 1 and using tap hole 2, with approximately 8.7 together, is relatively smaller than the error of the experiments using tap hole 3 and tap hole 4, which is around 10.5. This indicates that the temperature measured in the first tap holes is better explained by the control variables than in the later tap holes. The MAE using all holes (9.54) is approximately the average MAE of the separate tap holes (9.6).

Figure 11 – Qualitative experiment: predicting t_{+1}

Figure 12 – Qualitative experiment: predicting t_{+2}

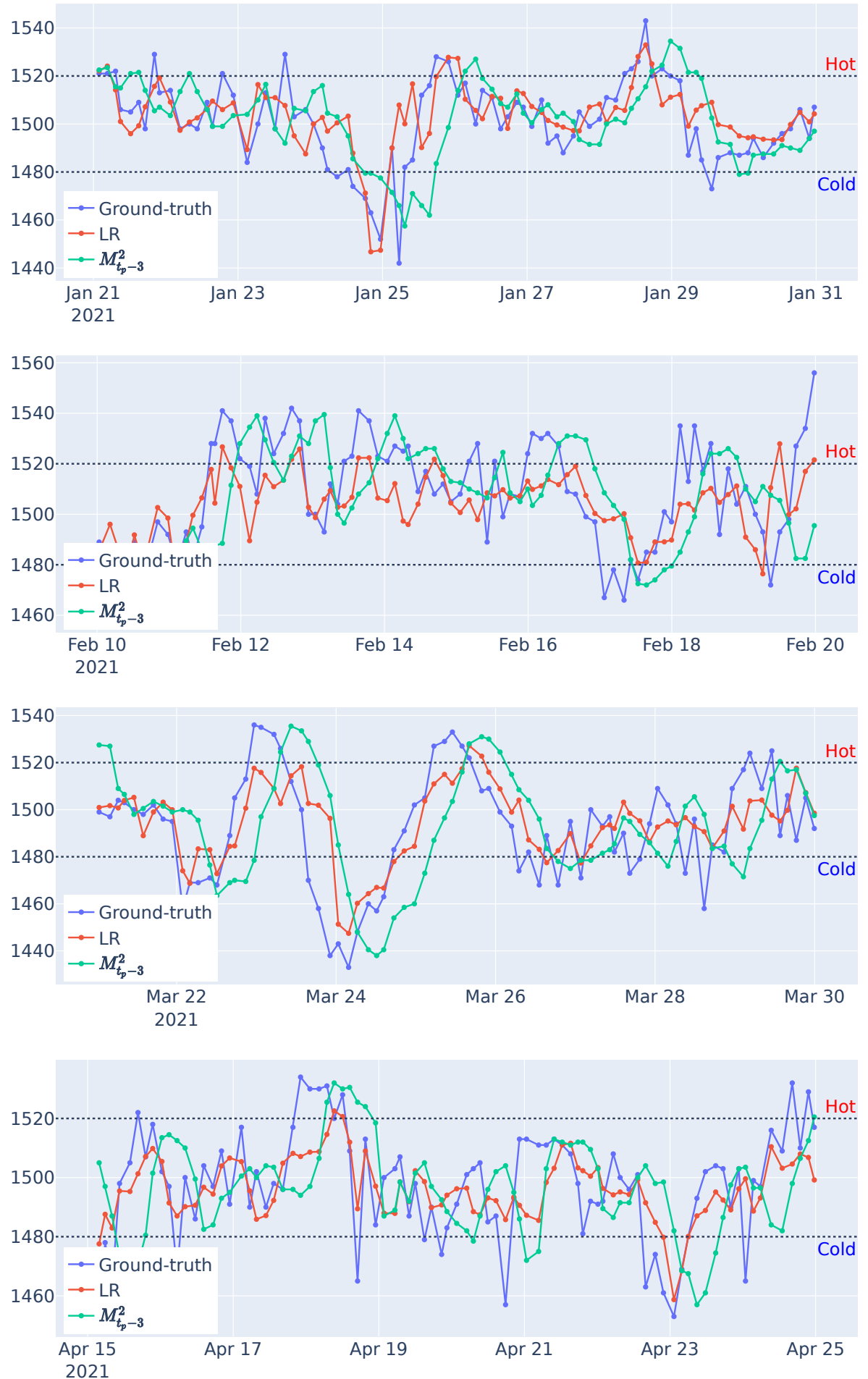
Figure 13 – Qualitative experiment: predicting t_{+2}



Figure 14 – Qualitative experiment: error of predicting t_{+1} , t_{+2} and t_{+3} (top to bottom) between 2021-01-21 and 2021-01-31.

5.1.4 Custom History Length

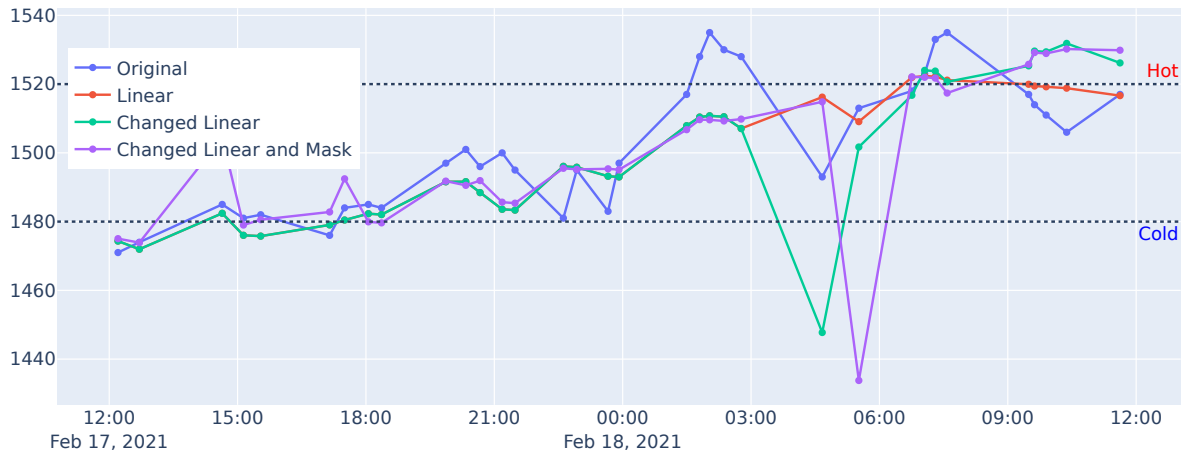
To verify the performance of the custom history length, we compared the results using the Linear regression model (the best and fastest model in the experiment) with and without the custom history length. The metric itself does not show an improvement. The custom history length achieves a MAE of 10.144 against the MAE of the original t_{+1} with

Table 8 – Tap Hole Experiment – Absolute Temperature – Predicting t_{+1} – 8-Hours History Length

Selected Tap Hole	MAE
Tap Hole 1	8.655
Tap Hole 2	8.813
Tap Hole 3	10.407
Tap Hole 4	10.757

9.66, so we created a situation to check if the custom history length was actually used or not.

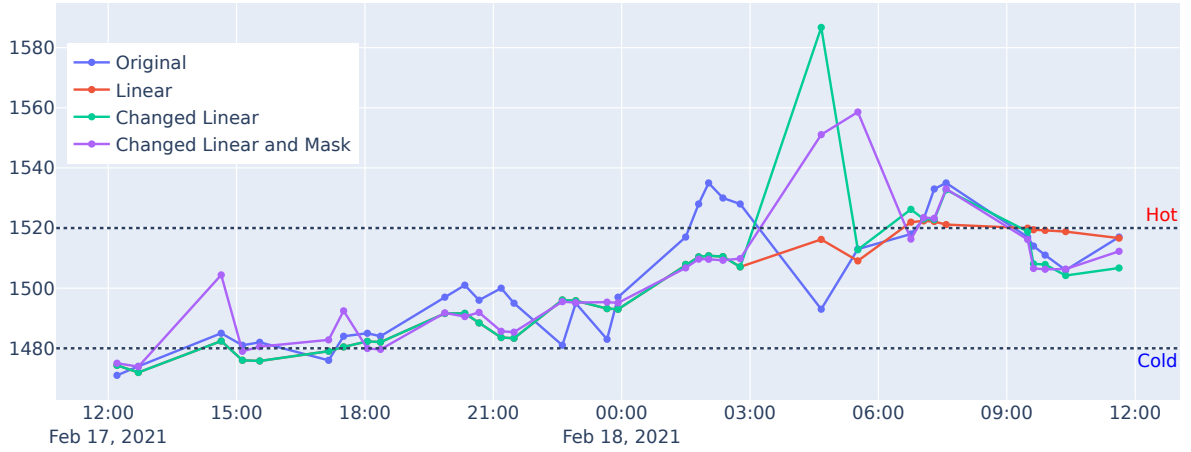
To perform this experiment, we generate artificial values in two control variables, PCI rate and Increased Oxygen Rate at the 04:30h in Feb 18, 2021. With these artificial values, is more clear to see the inference of a specify control variable in the predict temperature. As can be seen in the Figures 15 and 16, with a custom history length the model have a delay to receive the impact of the variable, showing that the purpose of the custom history length, is achieved.

Figure 15 – Mask Experiment: Increased PCI variable t_{+1} – Absolute Temperature

From these results, the custom history length shows that fulfills its purpose, influencing the temperature at the right time, despite the MAE was worse than the original experiment predicting t_{+1} one casting ahead.

5.2 Projected Temperature

In this section, the results referring to the experiments and analysis using predicted temperatures are presented.

Figure 16 – Mask Experiment: Increased Oxygen Rate variable t_{+1} – Absolute Temperature

5.2.1 Main Experiment

In this experiment we evaluated the performance of projected temperature modeling to isolate the tap holes making the prediction. As can be seen in Table 9, a better result was obtained (MAE of 9.56), compared to the absolute temperature (MAE of 9.66 in Table 5), without using the last measured temperature.

Table 9 – Projected Temperatures – Predicting t_{+1} – 8-Hours History Length

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
Linear	9.56	0.64	12.33	33.76	59.93	60.60	63.56
PLS	9.60	0.64	12.35	33.45	59.90	60.50	62.90
EBM	10.36	0.69	13.20	30.26	56.74	59.09	60.64
LGBM	10.45	0.70	13.36	30.74	56.43	58.41	59.03
XGBoost	10.59	0.71	13.59	30.15	56.60	57.96	59.68
SVM	10.89	0.73	13.96	29.71	55.13	58.37	59.03
MLP	10.99	0.73	14.17	30.15	53.65	56.38	59.16
RF	11.00	0.73	14.04	28.85	54.58	55.25	56.90
KNN	17.16	1.14	21.54	18.46	35.82	47.05	45.30

In general, all methods performed better than the absolute temperature experiment, decreasing the MAE. There was an improvement in trend (TM) but a slight decrease in accuracy (AAR).

The Figure 17 shows the qualitative result of the projected temperature, presenting a result similar to the absolute temperature, with better performance than the baseline being closer to the original temperature.

Visually comparing the absolute temperature modeling and the projected tem-

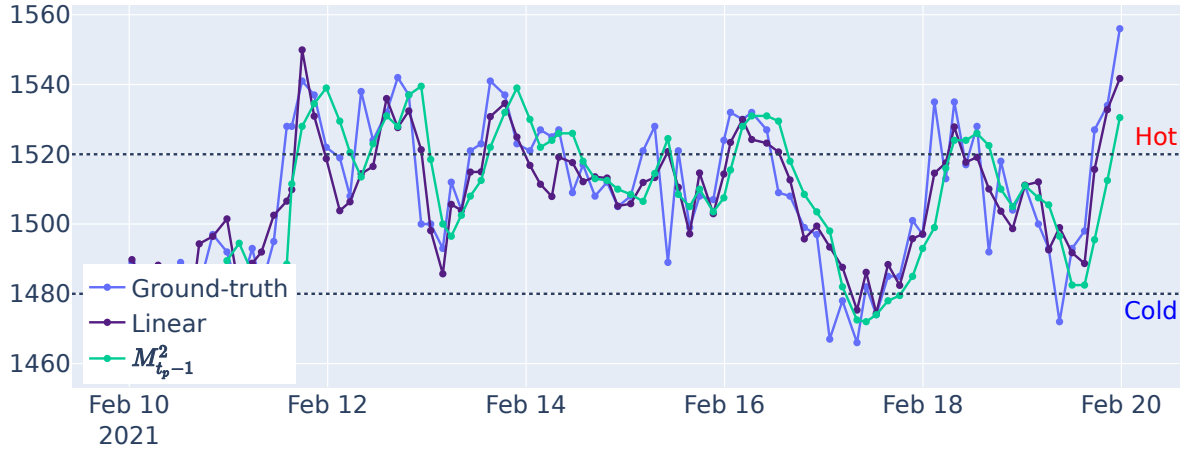


Figure 17 – Qualitative experiment: predicting t_{+1} with projected temperature

perature modeling (seen in Figure 18), the two models had a similar behavior, in some moments the absolute modeling is closer to the ground-truth, as in 23 Apr, and in other moments the projected modeling temperature works best as in 19 Apr.

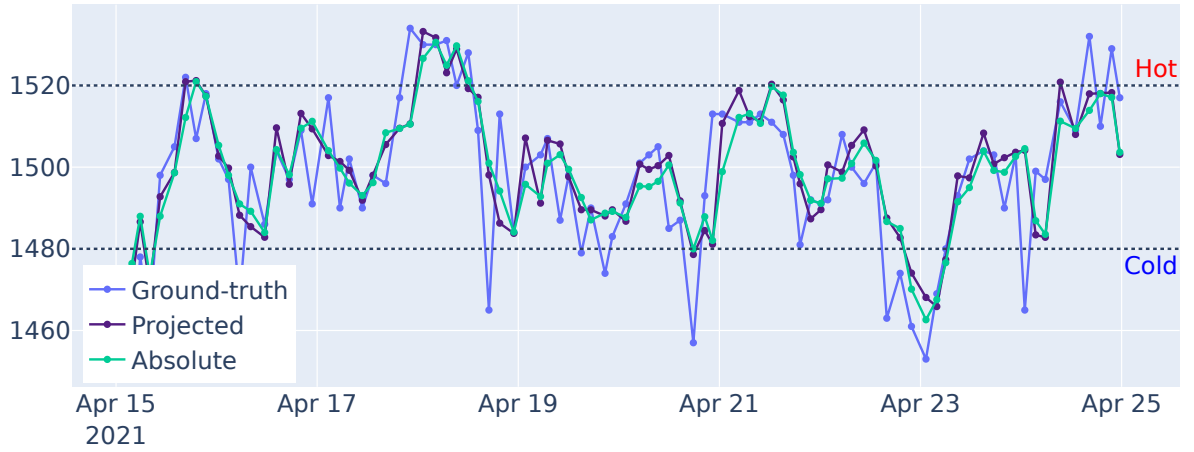


Figure 18 – Qualitative experiment: predicting t_{+1} with projected temperature

5.2.2 Custom History Length

To verify the performance of the custom history length, we compared the results using the Linear regression model (the best and fastest model in the experiment) with and without the custom history length, implemented in the same way as modeling absolute temperatures. Such as the absolute temperatures model, the MAE does not show an improvement, achieving a MAE of 9.99 against to a 9.56 from the main experiment.

To perform this experiment, we generate artificial values in two control variables, PCI rate and Increased Oxygen Rate at the 04:30h in Feb 18, 2021. With these artificial values, is more clear to see the inference of a specify control variable in the predict temperature.

The Figures 19 and 20 shows the results with a custom history length. The first figure shows a similar result to the absolute temperature, with the HMT being affected after a longer time than in the original model. In the second figure, the model is also affected after a while, but in a different way than the absolute temperature, probably due to a particularity of the projected temperature model.

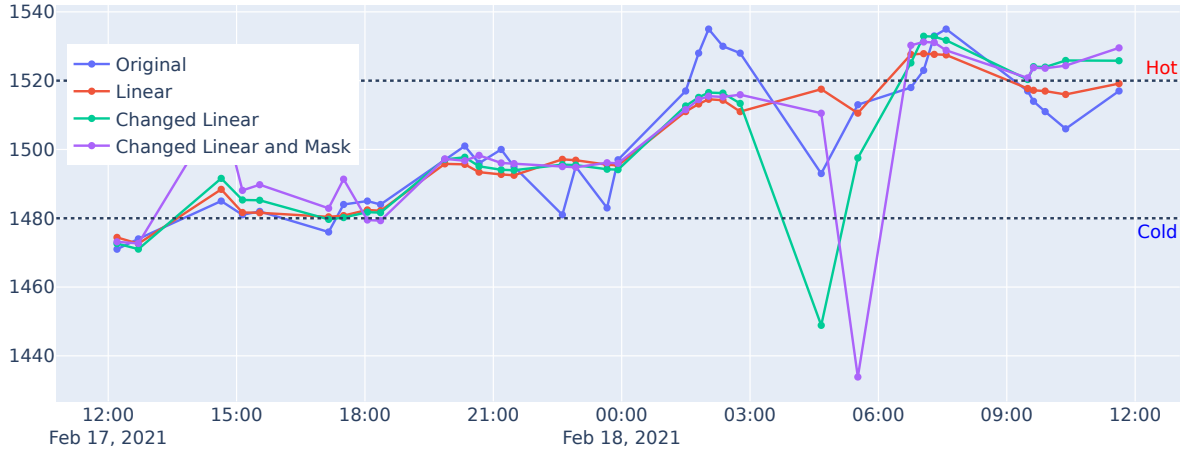


Figure 19 – Mask Experiment: Increased PCI variable t_{+1} – Projected Temperature

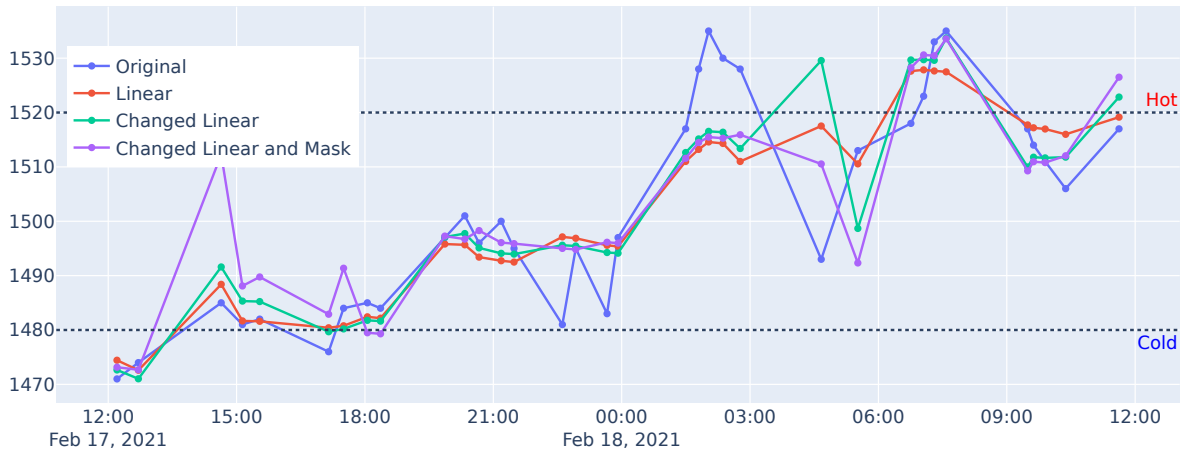


Figure 20 – Mask Experiment: Increased Oxygen Rate variable t_{+1} – Projected Temperature

Despite the difference between absolute temperature and projected temperature experiments, the custom history length also serves its purpose in projected temperature by inserting a delay to affect the hot metal temperature in the prediction.

5.2.3 Analysis by Tap Hole

An analysis evaluating the model prediction separately by hole was also done with the projected temperature. As you can see in Table 10, tap holes 1 and 2 still had a smaller error than using all tap holes as input, but tap hole 1 obtained a worse MAE (8.737) than

the experiment with absolute temperature using only tap hole 1 (MAE of 8.655), while tap hole 4, obtained a MAE of 10.260, better than the absolute temperature experiment using only tap hole 4(10.757).

Table 10 – Tap Hole Experiment – Projected Temperature – Predicting t_{+1} – 8-Hours History Length

Selected Tap Hole	MAE
Tap Hole 1	8.737
Tap Hole 2	9.090
Tap Hole 3	10.127
Tap Hole 4	10.260

This indicates that the projected temperature modeling renders the tap holes more independent of the influence of others tap holes, as it does not pass a temperature from another hole at the entrance. This makes the prediction by tap hole using projected temperature closer to a prediction with all tap holes than the predict by tap hole using absolute temperatures.

5.3 Sensors Variables Analysis

The experiments in this sections comprises the two models, the absolute temperatures and projected temperatures, with the sensors variables.

The absolute temperature results can be seen in Table 11 . Modeling with sensor variables showed no improvement in MAE. The original absolute temperature had a MAE of 9.66 and with sensor variables a MAE of 9.77. The bests methods, are similar to the absolute temperature: PLS and Linear in the top. It is concluded that the variables of the sensors did not improve the model, despite being good for the operator, the model was not able to capture the necessary information from them for a good prediction.

The experiments using projected temperature also failed to see an improvement in a metric, as can be seen in Table 12. The main design goes from a MAE of 9.56 for Linear to a MAE of 9.72 for Linear as well, a result similar to absolute temperature modeling.

The sensor variables do not improve in the modeling with absolute temperature and projected temperature, showing that the sensor variables can have relevant and useful information for the operator, but not in the model prediction.

Table 11 – Sensors variables experiment - Absolute Temperature – Predicting t_{+1} – 8-Hours History Length * Baseline methods.

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
PLS	9.77	0.65	14.68	34.79	60.86	60.47	62.66
Linear	9.79	0.65	14.71	34.44	60.62	60.19	62.77
LGBM	10.70	0.72	13.83	30.70	56.67	55.11	57.89
SVM	11.07	0.74	14.20	29.40	54.51	57.04	57.52
XGBoost	11.41	0.76	14.68	28.92	53.89	51.27	55.11
RF	11.60	0.77	14.86	28.51	53.17	50.17	53.40
EBM	11.97	0.80	15.20	26.48	51.08	53.19	54.32
MLP	12.37	0.83	15.74	27.00	49.02	53.88	52.71
$M_{t_p-1}^2$ *	12.61	0.84	16.06	27.38	49.57	44.41	50.03
T_{t-1} *	13.83	0.92	17.55	26.07	45.80	26.15	45.81
KNN	16.81	1.12	21.15	18.63	36.16	47.70	45.33

Table 12 – Sensors variables experiment - Projected Temperature – Predicting t_{+1} – 8-Hours History Length * Baseline methods.

Method	MAE	MAPE	RMSE	AAR(5)	AAR(10)	TM(5)	TM(10)
Linear	9.72	0.65	16.38	33.93	61.34	60.78	63.59
PLS	9.83	0.66	14.11	33.07	59.79	60.36	62.63
LGBM	10.51	0.70	13.47	30.12	57.32	57.38	58.85
EBM	10.68	0.71	13.75	30.70	54.85	58.34	60.36
XGBoost	10.74	0.72	13.83	30.91	55.92	57.93	58.13
SVM	10.78	0.72	13.68	29.37	54.61	58.51	59.44
RF	11.13	0.74	14.20	28.95	53.76	55.73	56.49
MLP	12.32	0.82	20.04	28.06	50.60	56.62	57.04
$M_{t_p-1}^2$ *	12.61	0.84	16.06	27.38	49.57	44.41	50.03
T_{t-1} *	13.83	0.92	17.55	26.07	45.80	26.15	45.81
KNN	16.93	1.13	21.28	18.53	35.88	47.63	45.16

5.4 Limitations

The study had some limitations in its development. The furnace operator did not have the opportunity to effectively test the prediction system on the fly. Therefore, the whole analysis was performed offline. In addition, the data had a large amount of noise, with variations of up to 10°C in temperature measurements, which affected the accuracy

of the results. The modeling of the problem was limited to predicting temperature values only for the next casting, not being possible to predict short-term values, such as half an hour, which may be desirable within the blast furnace operation. The temperature range of hot metal temperature to ensure good quality is very small (1480°C - 1520°C), this together with the sensor measurement error of 10°C makes it difficult to make a good prediction, as the range is small and the error is large. And finally, the gap in the measurement temperature between two interleaved tap holes at the same time can affect the prediction.

6 Conclusion

In this work, we investigated the effectiveness of different machine learning regression techniques for predicting temperature in a blast furnace obtaining a perform better than naive baselines used for comparison. We used a real-world dataset consisting of temperature measurements collected over a period of time in different tap holes of a blast furnace.

We evaluated the performance of machine learning regression methods: KNN, LR, PLS, EBM, LGBM, RF, SVM, XGBoost, and MLP, in two different modeling, using absolute temperatures and projected temperatures, using two approaches: with sensors variables and without sensors variables.

The results showed that the PLS and LR techniques performed best in comparison to other methods and the baseline for all models. The model with projected temperature, without using sensors variables, proved to be the best method for the problem. To make a good prediction, it is necessary to have the temperature information of the tap hole to be predicted, as this has been proven.

Overall, our results suggest that machine learning regression techniques can be useful for predicting temperature in a blast furnace but is necessary an evaluation for the operator to guarantee the efficiency. However, it is important to consider the limitations of the data and any potential issues that may impact the accuracy of predictions. These aspects should be taken into account in future research to further improve the performance of these models. In future works, a control system is included that indicates the next steps and suggests values of control variables to the operator based on the temperature predict, and a tap hole driven prediction system, which can counteract the tap hole difference limitation.

Bibliography

ABDI, H. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley interdisciplinary reviews: computational statistics*, Wiley Online Library, v. 2, n. 1, p. 97–106, 2010. Quoted in the page 23.

AZADI, P. et al. A hybrid dynamic model for the prediction of molten iron and slag quality indices of a large-scale blast furnace. *Comput. Chem. Eng.*, Elsevier, v. 156, 2022. Quoted 2 times in the pages 17 and 30.

BAG, S. K. Ann based prediction of blast furnace parameters. *Journal–The Institution of Engineers*, v. 68, n. 1, p. 37–42, 2007. Quoted in the page 17.

BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, v. 24, 2011. Quoted in the page 39.

BHATTACHARYA, T. Prediction of silicon content in blast furnace hot metal using partial least squares (pls). *ISIJ international*, The Iron and Steel Institute of Japan, v. 45, n. 12, p. 1943–1945, 2005. Quoted in the page 23.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <<http://dx.doi.org/10.1023/A%3A1010933404324>>. Quoted in the page 23.

CARDOSO, W.; FELICE, R. di; BAPTISTA, R. C. Artificial neural network for predicting silicon content in the hot metal produced in a blast furnace fueled by metallurgical coke. *Materials Research*, SciELO Brasil, v. 25, 2022. Quoted in the page 27.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794. Quoted in the page 24.

CINAR, A. C. Training feed-forward multi-layer perceptron artificial neural networks with a tree-seed algorithm. *Arabian Journal for Science and Engineering*, Springer, v. 45, n. 12, p. 10915–10938, 2020. Quoted in the page 26.

DING, S. et al. Dynamic prediction of the silicon content in the blast furnace using lstm-rnn-based models. In: IEEE. *2018 International Computers, Signals and Systems Conference (ICOMSSC)*. [S.l.], 2018. p. 491–495. Quoted in the page 30.

GAO, C.; JIAN, L.; LUO, S. Modeling of the thermal state change of blast furnace hearth with support vector machines. *IEEE Transactions on Industrial Electronics*, IEEE, v. 59, n. 2, p. 1134–1145, 2011. Quoted in the page 17.

GEERDES, M.; CHAIGNEAU, R.; LINGIARDI, O. *Modern Blast Furnace Ironmaking: An Introduction (Fourth Edition, 2020)*. IOS Press, 2020. ISBN 9781643681238. Disponível em: <<https://books.google.com.br/books?id=cy4NEAAQBAJ>>. Quoted in the page 26.

GONZÁLEZ, I. H.; KAMIŃSKI, J. The iron and steel industry: a global market perspective. *Gospodarka Surowcami Mineralnymi-Mineral Resources Management*, Komitet Zrównoważonej Gospodarki Surowcami Mineralnymi PAN, p. 5–28, 2011. Quoted in the page 16.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Quoted in the page 20.

HASHIMOTO, Y. et al. Automatic control of hot metal temperature. *Metals*, MDPI, v. 12, n. 10, p. 1624, 2022. Quoted in the page 30.

HASHIMOTO, Y.; SAWA, Y.; KANO, M. Online prediction of hot metal temperature using transient model and moving horizon estimation. *ISIJ international*, The Iron and Steel Institute of Japan, v. 59, n. 9, p. 1534–1544, 2019. Quoted 2 times in the pages 17 and 30.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, v. 30, 2017. Quoted in the page 24.

LEGENDRE, A. *Mémoire sur les opérations trigonométriques: dont les résultats dépendent de la figure de la terre*. F. Didot, 1805. (Analyse des triangles tracés sur la surface d'un sphéroïde, N^o 1). Disponível em: <<https://books.google.at/books?id=7C9RAAAAYAAJ>>. Quoted in the page 22.

LIN, S. et al. Model of hot metal silicon content in blast furnace based on principal component analysis application and partial least square. *Journal of Iron and Steel Research, International*, Elsevier, v. 18, n. 10, p. 13–16, 2011. Quoted 2 times in the pages 17 and 30.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning, second edition*. MIT Press, 2018. (Adaptive Computation and Machine Learning series). ISBN 9780262039406. Disponível em: <<https://books.google.com.br/books?id=V2B9DwAAQBAJ>>. Quoted in the page 21.

NAVARRO, L. C. et al. Temperature prediction in blast furnaces: A machine learning comparative study. In: IEEE. *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2022. v. 1, p. 192–197. Quoted in the page 19.

NORI, H. et al. Accuracy, interpretability, and differential privacy via explainable boosting. In: MEILA, M.; ZHANG, T. (Ed.). *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021. (Proceedings of Machine Learning Research, v. 139), p. 8227–8237. Disponível em: <<https://proceedings.mlr.press/v139/nori21a.html>>. Quoted in the page 24.

NORI, H. et al. *InterpretML: A Unified Framework for Machine Learning Interpretability*. 2019. ArXiv. Disponível em: <<https://www.microsoft.com/en-us/research/publication/interpretml-a-unified-framework-for-machine-learning-interpretability/>>. Quoted in the page 24.

OGUNLEYE, A.; WANG, Q.-G. Xgboost model for chronic kidney disease diagnosis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, v. 17, n. 6, p. 2131–2140, 2020. Quoted in the page 25.

ÖSTERMARK, R.; SAXÉN, H. Varmax-modelling of blast furnace process variables. *Eur. J. Oper. Res.*, Elsevier, v. 90, n. 1, p. 85–101, 1996. Quoted in the page 17.

PANDIT, S.; CLUM, J. Modeling, prediction and control of blast furnace operation from observed data by multivariate time series.(retroactive coverage). In: *Ironmaking Proceedings*, [S.l.: s.n.], 1975. v. 34, p. 403–416. Quoted in the page 17.

PETTERSSON, F.; CHAKRABORTI, N.; SAXÉN, H. A genetic algorithms based multi-objective neural net applied to noisy blast furnace data. *Applied Soft Computing*, Elsevier, v. 7, n. 1, p. 387–397, 2007. Quoted in the page 17.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Quoted in the page 26.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004. Quoted in the page 25.

SU, X. et al. Prediction model of hot metal temperature for blast furnace based on improved multi-layer extreme learning machine. *International Journal of Machine Learning and Cybernetics*, Springer, v. 10, n. 10, p. 2739–2752, 2019. Quoted in the page 30.

WANG, X.; HU, T.; TANG, L. A multiobjective evolutionary nonlinear ensemble learning with evolutionary feature selection for silicon prediction in blast furnace. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 33, n. 5, p. 2080–2093, 2021. Quoted 2 times in the pages 17 and 27.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021. Quoted 2 times in the pages 20 and 23.

ZHANG, X.; KANO, M.; MATSUZAKI, S. A comparative study of deep and shallow predictive techniques for hot metal temperature prediction in blast furnace ironmaking. *Comput. Chem. Eng.*, Elsevier, v. 130, 2019. Quoted 4 times in the pages 16, 17, 23, and 30.

ZHANG, X.; KANO, M.; MATSUZAKI, S. Ensemble pattern trees for predicting hot metal temperature in blast furnace. *Computers & Chemical Engineering*, Elsevier, v. 121, p. 442–449, 2019. Quoted 2 times in the pages 17 and 30.

ZHOU, P. et al. Data-driven robust m-ls-svr-based narx modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. *IEEE Trans. Neural Netw. Learn. Syst.*, v. 29, n. 9, p. 4007–4021, 2017. Quoted 2 times in the pages 27 and 30.

ZHU, X.; GOLDBERG, A. *Introduction to Semi-supervised Learning*. Morgan & Claypool, 2009. (Synthesis lectures on artificial intelligence and machine learning). ISBN 9781598295474. Disponível em: <https://books.google.com.br/books?id=c_haJrQ0ScAC>. Quoted in the page 20.