Optimizing Boiler Control in Real-Time with Machine Learning for Sustainability

Yukun Ding
University of Notre Dame
Notre Dame, Indiana
yding5@nd.edu

Jinglan Liu
University of Notre Dame
Notre Dame, Indiana
jliu16@nd.edu

Jinjun Xiong
IBM Thomas J. Watson Research Center
Yorktown Heights, New York
jinjun@us.ibm.com

Meng Jiang
University of Notre Dame
Notre Dame, Indiana
mjiang2@nd.edu

Yiyu Shi
University of Notre Dame
Notre Dame, Indiana
yshi4@nd.edu

ABSTRACT

In coal-fired power plants, it is critical to improve the operational efficiency of boilers for sustainability. In this work, we formulate real-time boiler control as an optimization problem that looks for the best distribution of temperature in different zones and oxygen content from the flue to improve the boiler's stability and energy efficiency. We employ an efficient algorithm by integrating appropriate machine learning and optimization techniques. We obtain a large dataset collected from a real boiler for more than two months from our industry partner, and conduct extensive experiments to demonstrate the effectiveness and efficiency of the proposed algorithm.

CCS CONCEPTS

• **Hardware** → **Temperature optimization**; • **Information systems** → *Process control systems*;

KEYWORDS

Boiler Control; Combustion Optimization; Sustainability

ACM Reference Format:

Yukun Ding, Jinglan Liu, Jinjun Xiong, Meng Jiang, and Yiyu Shi. 2018. Optimizing Boiler Control in Real-Time with Machine Learning for Sustainability. In The 27th ACM International Conference on Information and Knowledge Management (CIKMâĂŹ18), October 22–26, 2018, Torino, Italy. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3269206.3272024

1 INTRODUCTION

As coal-fired power plants currently produce 41% of global electricity [27], proper control of coal-fired boilers in producing electricity is not only essential to the safety of power plant operation, but also directly affects boilers stability, energy efficiency, and sustainability, thus having huge socioeconomic and environmental impacts [9]. How to optimally control boilers' operating condition in real-time is, however, difficult. The combustion process inside a boiler is highly complex and nonlinear with strong-coupling and time-delayed influences. It is well understood in literature that it is not easy to achieve high efficiency in operating large utility boilers and most existing practices in the industry are highly sub-optimal [4].

Nonuniform temperature distribution inside a boiler is known to cause tube rupture, a frequent failure mechanism for boiler operations. But to maintain a uniform temperature distribution inside a boiler is difficult even for domain expert in practice due to the dynamic air flow inside the boiler. One of the most frequently used practices today to deal with nonuniform temperature distribution is spraying water inside a boiler, which introduces unnecessary efficiency loss and additional operating cost [9, 19]. Another practice is to remold a boiler by re-arranging super-heater panels to alleviate the uneven temperature distribution [29], which requires to shut down the boiler and cannot be done in real-time. Temperature distribution inside a boiler has been studied using various computational fluid dynamics (CFD) methods under steady-state conditions [9, 19]. However, employing the CFD methods for real-time boiler control is not feasible because of the extremely high computational cost of solving the CFD models [2].

To avoid solving the physical-based CFD models, researchers have proposed to use machine learning-based methods to predict boiler temperature or other related parameters in order to control boiler combustion process [14, 15, 34]. However, such formulations have mainly focused on reduction of pollutant emission, not for the uniform distribution of temperature inside boilers. There are some works focusing on the boiler efficiency optimization [8, 13] where external measurements (e.g. exhaust gas temperature) are used to estimate the boiler efficiency through some models due to the difficulties in obtaining temperatures within the boiler. Instead, in this work we have collected the temperature distribution data within the boiler from our industry partner, which allows precise and accurate observation of the combustion efficiency and stability. Moreover, due to the strong-coupling, nonlinear and large time delay characteristics of boilers, existing solutions often result in a complicated black-box optimization problem, which requires a significant amount of time to obtain the solution, and thus can

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3272024

hardly be used in real-time optimization of boilers [4, 14, 28, 34]. The major contribution of this paper is three-fold.

- First, we introduce a novel formulation to the boiler control problem based on inputs from industry, i.e., maintaining a uniform distribution of temperature in different zones and a balanced oxygen (O_2) content from the flue in a coal-fired power plant. The importance of this objective is well understood in the industry for improving boiler stability and energy efficiency. However, such a formulation has never been addressed in the literature to the best of our knowledge.
- Second, we validate the formulation and show high solution quality using a real industry boiler dataset. This dataset is unique as it contains real-time measurements of temperature and O_2 content distribution within a boiler. It was this unique characteristics that enables us to formulate such a novel and realistic formulation for the boiler control problem. The dataset and our implementation are available at https://github.com/yding5/BoilerOptimization.
- Third, we develop a new but practical algorithm to solve the proposed real-time boiler control problem by combining machine learning and optimization techniques. Though none of those techniques are new by themselves, we integrate them in a way such that it affords us to solve the proposed boiler control problem in real-time. Moreover, we introduce an error compensation technique to address the local bias issues that are often difficult for machine learning techniques.

The rest of the paper is organized as follows. We first give a background about the boiler control problem, and then propose our new formulation with its mathematical structure. Based on the understanding of such a mathematical structure, we further propose a solution based on proper choice of both machine learning and optimization techniques. We then report the experimental results and make conclusions.

2 BACKGROUND AND RELATED WORK

We give a brief introduction of the operation of a power plant boiler. As shown in Figure 1, pulverized coal is fed into the furnace from different coal feeders with a proper volume of airflow, both of which are controlled by their respective throttle openings to maintain a desired air-to-fuel ratio for combustion. The water circulates in a water-steam system and absorbs the radiation energy from the furnace continuously until it becomes high-pressure superheated steam in the superheater. Through the steam turbines, the thermal energy is transferred to mechanical work and finally becomes electricity through generators. In a power plant, a central controller determines the desired setpoints for various subsystem controllers, a critical one of which is the combustion controller that determines the feed rates of coal and airflows [16].

Since combustion quality ultimately determines the production efficiency, we focus on combustion control in this paper. In general, higher temperature inside the furnace and lower O_2 content in the flue indicate higher efficiency. But to ensure sustainably high combustion efficiency and stability, it is desirable to also maintain a balanced high temperature distribution and low O_2 content in the flue, as a balanced distribution of both temperature and O_2 content indicates that both flames and pressures are uniformly distributed, and thus promising the stability and safety of the boiler. However, existing formulations [12, 14, 28, 34] have not considered the temperature distribution, and not mentioned the distribution of O_2 content.

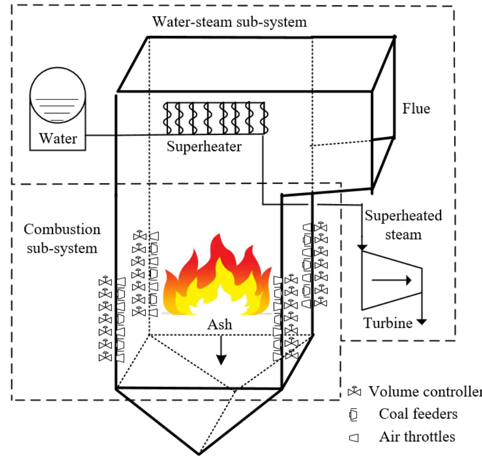

Figure 1: Simplified structure of coal-fired boiler

In current industry practice, the combustion controller consists of a set of PI/PID controllers and pre-computed set-pints (computed theoretically and fine-tuned empirically). The well-established control system based on PI/PID was improved by advanced PI/PID controller such as auto-tuning PID [31]. In recent years, machine learning-based prediction control has been studied widely and included in commercial boiler control solutions [11, 13]. The prediction model was used for steady state optimization initially and then gradually for real-time optimization [5, 16, 20, 23]. From the algorithm perspective, the modeling approaches are dominated by neural networks and their variants, e.g. vanilla feed-forward network, RBF network, double linear fast learning network [4, 14, 15]. The optimization problems are solved by various heuristic search algorithms, e.g. GA, PSO, DE, ACO [21, 33, 36]. However, due to the considerable number of variables, the computational requirement remains a challenge which degrades the real-time performance [35]. Instead of using all related data, models with fewer inputs are investigated to reduce the runtime of optimization [26]. However, none of the existing works considered the uniform temperature and oxygen distribution.

The modeling of temperature and O_2 content are more challenging than the modeling of NOx emission in a steady state, as their dynamics are affected by a number of variables through an extremely complex and time-delayed process especially when they are not average values in the boiler over a period of time. In addition, it is difficult to obtain the temperature data in a large furnace even with special sensors because of the heat. To provide a practical real-time control solution for boiler combustion, the solution needs to be solved in real-time, i.e., the prediction of temperature distribution and O_2 content and the optimization of finding the proper control inputs all need to be solved as soon as obtaining measurements at current time step. This poses a strict constraint on the choice of the modeling techniques and computational methods to solve the real-time boiler control problem.

Table 1: Summary of important model parameters

Parameter	Definition	Method
N	Number of past data point(s) used in prediction models as part of input.	Hyper-parameter chosen by a grid search.
S	Window size in error compensation, which is the number of previous samples used as input for error compensation.	Hyper-parameter chosen by a grid search.
$W_1, ..., W_4$	Weighting constants for trade-off among four objective terms.	Chosen by users according to their respective importance.
$T_t(i), \overline{T_t}$	Variables. Temperatures in the boiler at zone $i \in \{1, \cdots, 6\}$ and their average at time step t.	Solved.
$O_t(j), \overline{O_t}$	Variables. O_2 contents from left side and right side of the flue and their average at time step t. $j \in \{1, 2\}$	Solved
$\gamma_1, \gamma_2, \gamma_3, \gamma_4, \alpha_T, \beta_O$	Constants for standardization.	Determined from the data.
$x_t(k)$	Variables. Controllable parameters including 12 coal feed rates (for k=1 to 12) and 16 airflow throttle openings (for k=13 to 28) at time step t.	Solved
$x_{t,min}(k), x_{t,max}(k)$	Bounds of every controllable variable at time step t.	Given by users.
$L^A_{min}, L^A_{max}, L^C_{min}, L^C_{max}$	Bounds for total air flows and total coal feeding rates.	Given by users.
M_{t-1}	Variables. All uncontrollable input for the prediction model, including historical operations and real-time measurements of boilers at time step t , such as the generation load, etc.	Determined from the data.
$\beta^T_{t,i}, \beta^O_{t,j}, b^T_{t,i}, b^O_{t,j}, H, f^{\mathrm{T}}, c$	Intermediate variables	Derived from models.

3 PROBLEM FORMULATION

We formulate a new boiler control problem in this section by not only maintaining a high temperature and low O_2 content, but also maintaining a uniform distribution of temperature in different zones and O_2 content inside the flue in a coal-fired power plant. For clarity, we summarize the meaning of important model parameters and how their values are determined in Table 1. Note that although the number of model parameters is quite large, only a small number of them are hyper-parameters that need to be tuned by users. To capture the time-dependent effects of controllable variables, subscripts are added to indicate time step.

The goal of achieving a balanced distribution of temperatures and O_2 content can be captured by a quadratic penalty function of the deviation of temperatures from the average value and the difference between O_2 content from two sides in the flue. Certainly there are other options but quadratic penalty function is employed, because it is differentiable, suitable for capturing the deviation from the desired value, and relatively simple for optimization. For effective combustion, we also want to maintain a high temperature and low O_2 content inside a boiler. Together, we can use a weighted sum of these components as our objective, and formulate the boiler control problem as the following constrained optimization problem in Equation (1) according to constraints under real operation, where the polynomial objective function has four terms, indicating the variance of temperature in different zones, the difference of O_2 content in two sides of flue, the average temperature, and the average O_2 content, respectively. The problem needs to be solved continuously for every time stamp t based on data including operations from $t-1$ and prior in order to achieve the goal of real-time control for the boiler. f_i^T and f_j^O define prediction models for temperature and O_2 respectively, which will be discussed in detail in the next section.

$$
\begin{aligned}
\min_{x_{t-1}(i)} V_t = & W_1 \times \gamma_1 \times \frac{\sum_{i=1}^{6} (T_t(i) - \overline{T_t})^2}{6} + \\
& W_2 \times \gamma_2 \times (O_t(1) - O_t(2))^2 - \\
& W_3 \times \gamma_3 \times (\overline{T_t} - \alpha_T) + \\
& W_4 \times \gamma_4 \times (\overline{O_t} - \beta_O) \\
s.t. \quad & T_t(i) = f_i^T(x_{t-1}, M_{t-1}), i = 1, \dots, 6 \\
& O_t(j) = f_j^O(x_{t-1}, M_{t-1}), j = 1, 2 \\
& x_{t-1,min}(k) \le x_{t-1}(k) \le x_{t-1,max}(k) \\
& k = 1, 2, ..., 28 \\
& L^C_{t-1,min} \le \sum_{k=1}^{12} x_{t-1}(k) \le L^C_{t-1,max} \\
& L^A_{t-1,min} \le \sum_{k=13}^{28} x_{t-1}(k) \le L^A_{t-1,max}
\end{aligned}
\tag{1}
$$

4 SOLUTION FRAMEWORK

The structure of proposed real-time boiler control framework is shown in Figure 2. The prediction models, f_i^T and f_j^O trained on historical data, provide the symbolic expression of temperature and O_2 content based on control variables and other measured uncontrollable variables, which are denoted as x_{t-1} and M_{t-1} respectively. In every time step, M_{t-1} in the symbolic expression will be replaced by the latest observed values and only the controllable variables x_{t-1} and V_t remain. Then the optimization model takes the resulted expression and solves the nonlinear programming problem to give the optimal combination of controllable variables, which is the control input to the boiler. An error compensation module, which will be discussed later, is employed to further improve the prediction accuracy.

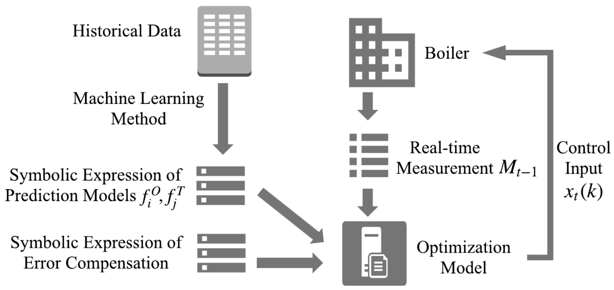

Figure 2: Structure of the solution framework

The time cost for solving the optimization model at every time step depends on the choice of optimization algorithm and the complexity of the problem determined by the prediction model. Since the control loop needs to be continuously solved for every time step as soon as possible, the runtime performance of the optimization model is the critical consideration.

We employ machine learning-based approaches for predicting both temperature and O_2 content. We notice that there is a special mathematical structure of Equation (1), where the constraints are linear with respect to controllable variables $x(i)$ and the objective is quadratic with respect to the predicted values $T(i)$ and $O(i)$. Therefore, among many possible choices of machine learning techniques, we propose to use the epsilon-support vector regression (ε-SVR) with linear kernel [25] as the prediction model. As it shall become clear in a moment, such a choice will render a nice mathematical structure for the optimization model, which in turn enables us to choose an effective optimization technique to solve the problem efficiently.

In ε-SVR, the goal is to find a function so that there has at most ε deviation from the actually obtained target for all training data, and at the same time, it is as flat as possible. Different from previous work [4, 28, 34] that only uses measurements in a boiler at a steady state to build the prediction model, we add a varying number of immediate past time data as part of the inputs to capture the time-delayed impacts for prediction. We obtain the following linear prediction models through the ε-SVR linear kernel method as Equation (2), where f_i^T is defined by the selected prediction model, and $\beta_{t-1,i}^T, b_{t-1,i}^T$ are obtained from the trained model with all known data substituted. The formulation for f_j^O is similar and thus omitted for the sake of brevity.

$$\begin{aligned} T_t(i) &= f_i^T(x_{t-1}, M_{t-1}) \\ &= \beta_{t-1,i}^T x_{t-1} + b_{t-1,i}^T \quad i \in \{1, ..., 6\} \end{aligned} \tag{2}$$

Plugging $T_t(i), \overline{T_t}, O_t(i), \overline{O_t}$ back into Equation (1) with some rearrangement, we obtain a quadratic programming model as follows (by dropping the subscript t for simplicity):

$$\begin{aligned} \min_x \quad & V(x) = \frac{1}{2}x^{\mathrm{T}}Hx + f^{\mathrm{T}}x + c \\ s.t \quad & A_q \cdot x \le b_q, \\ & A_e \cdot x = b_e, \\ & B_l \le x \le B_u. \end{aligned} \tag{3}$$

where H is a real symmetric matrix of coefficients, f is a vector of coefficients, and c is a constant term, all of which can be easily constructed based on values from the prediction model. A_q, A_e, b_q, b_e, B_l, and B_u are compact representation of known constraints.

Therefore, at each time step, we end up with a quadratic programming problem for the optimization model. We adopt an efficient algorithm for quadratic programming, the interior point convex (IPC) method [6, 7, 17]. It uses a presolve procedure to remove redundancies and simplify constraints. It then tries to find a point where the Karush-Kuhn-Tucker (KKT) conditions hold, and use multiple corrections to improve the centrality of the current iteration.

As discussed before, the working mechanism of a coal-fired boiler is extremely complex and time-varying, and not all factors are observable through measurements. Therefore, a machine learning-based prediction model of this kind may produce time-related local bias because of the change of underlying hidden factors, such as the fluctuation of coal quality and the restart of boilers. By borrowing an idea from the field of control, we add an error compensation part to further improve the prediction accuracy by compensating the local bias, which is estimated by computing an average difference between the actual output and predicted output for a prior window size of time steps, and adding this value to the future predicted output to decrease residuals [3, 30]. At every time step, the latest prediction error is obtained and the new compensation value is added to $T_t(i)$ and $O_t(j)$ as constants. The window size S is another algorithmic tuning parameter of this method. Note that the error compensation can be effective because the input and output are sampled from a physically continuous system, thus the adjacent prediction errors may implicitly stores contextual information and can be used for better prediction. This work is also an example of how a well-trained machine learning model can be improved by leveraging its physical meaning. This approach can be extended to other applications, where prediction is made about a continuous system and prediction error is available in online operation.

The workflow of the solution is shown in Algorithm 1. The prediction model is trained offline and does not need to be re-trained any more. The time to solve the optimization problem dominates the latency in the control loop which is the lag from the observation to the corresponding control operations. Because the lag is not taken into consideration when building the prediction model according the dataset, the closer the lag is to zero the better. It is also the reason to simplify the complex highly nonlinear optimization problem to a quadratic programming problem, which is very important for a real-time solution.

5 EXPERIMENTAL RESULTS

5.1 Experiment Setup

We conduct experiments using the real dataset collected by our industry partner from a production power plant boiler as discussed

Algorithm 1 Boiler control solution

Input: Historical data T, O, x, M
Output: Control signal x_t at every time step t
Train prediction functions f_i^T, f_i^O
Select the windows size S
for every time step t **do**
  Obtain measurement M_t
  Compute the error compensation value at t
    using previous S samples
  Substitute M_t into Equation (1) and Equation (2)
    to obtain H, f, c in QP
  Solve QP using IPC
  Output control signal for current time step x_t
end for

before. It contains more than 13,000 samples collected in a span of more than two months at a sample rate of 432 seconds. Each sample corresponds to a time stamp with 49 features including temperatures in six zones, O_2 content in two sides of flue, generation load, Nitric oxide in two sides of flue, twelve coal feed rates, and sixteen throttle openings, etc. All samples are numbered according to their time order. For the predicted variables of interests, temperatures and O_2 contents, their respective range are from 978℃ to 1403℃ and from 0.9824 to 11.43. The value of O_2 content indicates the percentage of O_2 in the flue gas. Their units are omitted for brevity.

For comparison purpose, we also implement different algorithms to show the effectiveness of our proposed algorithm. The alternative options used for the prediction model include the ε-SVR with a radial basis function (RBF) kernel, the classic three layer feed forward neural network (NN) with tangent-sigmoid activation function for the hidden layer, the vanilla recurrent neural network (RNN) [32], and the LSTM model [10]. The alternative options for the optimization model include some popular heuristic search algorithms, including genetic algorithm (GA) [1], differential evolution (DE) [17], particle swarm optimization (PSO) [24], and Sequential Quadratic Programming (SQP) [18]. All tuning parameters are selected by Bayesian optimization [22] or grid search. We select two continuous parts from the dataset to avoid the shut-down period of the boiler. The first part contains 7501 samples and is used for training and validation, while the second part contains 3801 samples and is used only for testing.

5.2 Comparison of Prediction Models

We first compare the prediction accuracy among the five prediction models. During the tuning of hyper-parameters, hold-out validation (6501 for traning and 1000 for validation) is used for RNN-based models and 10-fold cross-validation is used for others.

For each model, there are also different ways of organizing the input data (or feature selection for ε-SVR based methods). Three variants are considered: (A) non predicting data from the current time stamp, (B) all data from the current time stamp, and (C) all data from both the current time stamp and a varying number of past time steps. Most existing work on boiler control uses the type (A) data [4, 28, 34] as they assume a steady state model. Type (B) data is a special case of type (C) data with zero previous time step data.

To show the importance of organizing input data properly, we apply all the three types of data to the first three methods, and only type (B) data to RNN and LSTM models. The reason for the latter is that RNN and LSTM needs time-dependent data and the models themselves can be trained to capture the time-delayed effect through internal memories. The accuracy metrics used are averaged Mean Squared Error (MSE) and mean absolute percentage error (MAPE) of the six zones for temperature and two side for O_2.

The prediction accuracy for temperature is reported in Table 2. As it can be seen, for the first three methods, results from type (B) and (C) data are significantly better than those from type (A) data and results from type (C) data are the best. In terms of methods, the proposed ε-SVR linear model performs the best with the least average MSE value. Even when comparing results from type (B) data for all five methods, ε-SVR linear is still better than RNN and LSTM. It turns out that the ε-SVR linear model achieve good prediction accuracy and the temperature prediction result on test dataset is illustrated in Figure 3.

Table 2: Temperature prediction models

Model	Data Type	MSE	MAPE
SVR (linear)	(A)	975.3	2.06%
	(B)	289.2	1.12%
	(C)	**164.8**	**0.82%**
SVR (RBF)	(A)	1860.1	2.88%
	(B)	1199.8	2.24%
	(C)	246.6	1.01%
NN	(A)	1268.8	2.55%
	(B)	344.4	1.23%
	(C)	181.0	0.87%
RNN	(B)	635.6	1.89%
LSTM	(B)	841.7	1.98%

Although RNN and LSTM seem to be the most suitable models for time series data, we suspect the limited data size (albeit the largest in the literature) and the peculiarity of the system dynamics have prevented RNN and LSTM from achieving a stable solution such that the hidden states memorizing past information are weaker than raw data from past steps when supporting the subsequent predictions.

We also offer the following reasons to explain why our proposed ε-SVR with linear kernel performs the best. First, the measurements of temperature and O_2 content contain some outliers because of the unstable airflow inside the boiler. The ε-SVR with ℓ_1 loss is less sensitive to such outliers when compared to the ℓ_2 loss in other methods. Second, ε-SVR treats errors less than ε as zero, and is thus less sensitive to sensor noises than others, which further helps to reduce unnecessary updates in the training process. Third, the simple linear regression is less likely to be overfitting compared to those more complex nonlinear models.

To further illustrate the robustness of the prediction, we report the 10-fold cross-validation results for ε-SVR linear. The mean and standard deviation of temperature MSE in cross-validation for the

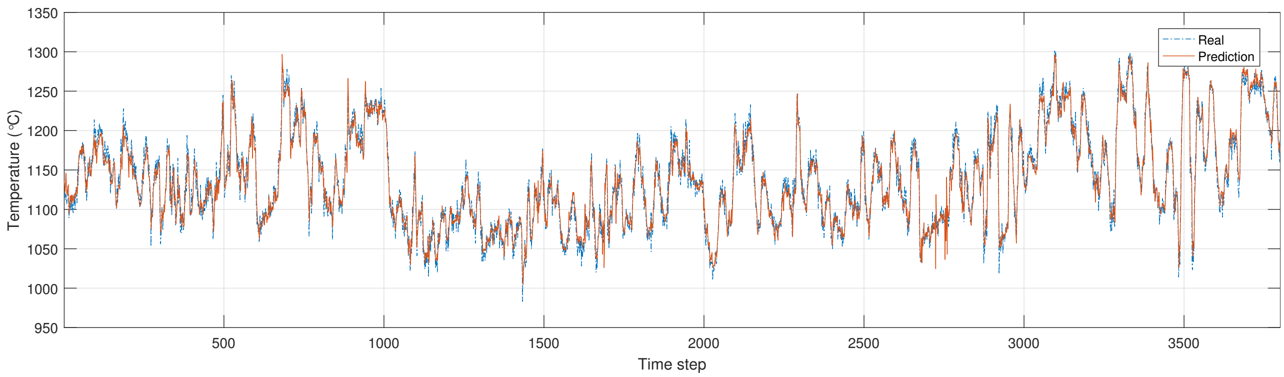

Figure 3: Comparison of real temperature and predicted temperature in zone 1

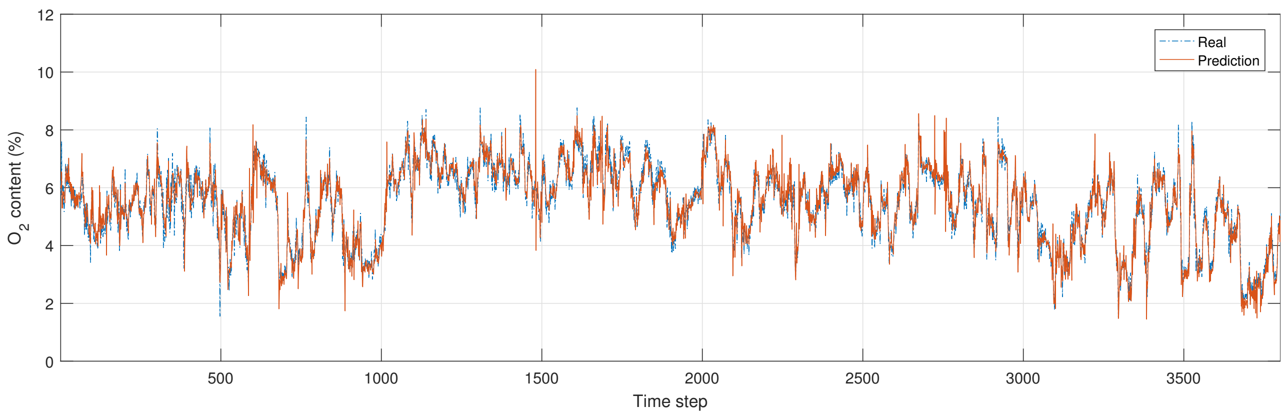

Figure 4: Comparison of real O_2 content and predicted O_2 content in right side of the flue

six zones are 126.5±8.0, 138.3±9.3, 139.3±7.8, 159.5±8.5, 127.3±7.5, and 130.4±7.7, respectively. The small standard deviations for all zones validate the robustness of the proposed modeling approach.

Bootstrapping was tried to improve the accuracy. However, it is found that bootstrapping can hardly improve the performance. After the resampling, the best average MSE increases from 164.8 to 168.1 (increases to 170.5 when normally distributed noise is added) in temperature prediction. We anticipated that the resampling reduces the diversity of operating condition.

The results of O_2 prediction is quite similar to that in the temperature prediction and shown in Table 3. Note that while the NN method with type (C) data has a low MAPE and a slightly lower MSE than that from SVR (linear) method with type (C) data, the NN method's performance is not stable. Without fixed initialization setting, the NN has a probability to result in a large MSE and must be handled separately to avoid hazarding the whole control system. In our 10,000 repeated experiments, there are 9 and 18 instances where the resulted MSE is higher than 5 times of the average MSE for two oxygen content respectively. Moreover, even though NN can provide better prediction performance, it cannot be used in the control system as it lead to a highly nonlinear optimization problem which is too complex to be solved in real-time. The prediction result of ε-SVR linear model on test dataset is illustrated in Figure 4. The average O_2 content MSE in cross-validation for the two sides of flue are 0.0734±0.010 and 0.1764±0.012, respectively.

Table 3: O_2 content prediction models

Model	Data Type	MSE	MAPE
SVR (linear)	(A)	0.606	12.37%
	(B)	0.251	7.97%
	(C)	**0.132**	**5.05%**
SVR (RBF)	(A)	0.929	16.58%
	(B)	0.679	13.71%
	(C)	0.183	13.76%
NN	(A)	0.656	12.45%
	(B)	0.298	8.30%
	(C)	0.125	5.35%
RNN	(B)	0.387	11.31%
LSTM	(B)	0.472	11.53%

5.3 Impact of Error Compensation

In this section, we report the impact of error compensation on the solution quality. Since there is no local bias in training dataset, after

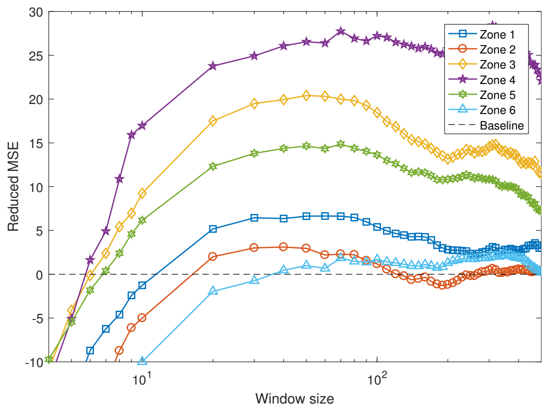

Figure 5: The reduced MSE versus window size of error compensation

training process we use the first 2000 test samples to choose a suitable window size for error compensation, and the rest 1801 samples for testing. Different window sizes can be tried on the historical data and the window size S can be selected with a trade-off between complexity and performance. Figure 5 shows the impact of different window sizes of error compensation on temperature prediction in different zones. The window size in x-axis indicates how many previous samples are used to calculate the error compensation, which is the average error for previous predictions. The y-axis stands for how much MSE are reduced by using error compensation with a given window size and thus the higher the better.

A rough rise and fall trend can be observed as expected in Figure5. When the window size is small, which means only a few latest errors are used to calculate the compensation, the error compensation makes prediction worse (negative values in Figure5) because high randomness dominates the compensation. When window size increases, the compensation helps to reduce prediction errors and these curves reach a peak since a proper window size enables us to discover a local bias covered by randomness. As the window size keeps increasing, these curves fall down as too many prediction errors from long ago are used. If the window size reaches a very large value, all curves will finally converge to a narrow range around zero because it leads compensation to a near-zero value, which is not shown within this figure. It is worth noting that predicted temperatures with lower accuracy tend to get more improvement from error compensation. We suspect those zones are more sensitive to some hidden factors and thus have more apparent local bias. Similar observations also hold for O_2 content prediction. We finally select 50 as the window size for error compensation calculation to strike a right balance. With error compensation, we further reduced the average MSE of temperatures and O_2 content by 7.4% and 3.4% respectively.

More complicated approaches such as SVR and NN also have been tested, but surprisingly, even though they get better result under some settings, the simplest average strategy gets the best and most stable performance overall. This is probably caused by the high randomness of the boiler system and its variance along with time. So the average strategy works best under this practical circumstance.

5.4 Comparison of Optimization Algorithms

To avoid too many combinations of prediction model and optimization model, we compare various optimization algorithms on the test dataset using the best prediction model obtained in last section. The weights W_1 to W_4 in the objective function Equation (1) are chosen the same for all algorithms, and set as 0.7, 0.1, 0.1 and 0.1, respectively.

To meet the real-time constraint, all optimization algorithms need to report results within a same given time interval, which is 432 seconds at most in our case for running the algorithms on a desktop with an Intel i5-4590 3.3GHz CPU, thus the converge time of algorithms that exceeds the limit is not reported. At each time step (one test sample), the optimization algorithm will produce one objective value as defined by Equation (1), and for all test samples, the objective values are collected for each model.

Table 4: Comparison of different optimization algorithms

Solving	Time	Objectives			
Algorithm	(sec)	Mean	Min	Max	Std
IPC	**0.16**	**0.085**	**-0.207**	**0.419**	**0.140**
DE	81.5	0.127	-0.168	0.497	0.145
SQP	159	0.117	-0.189	0.434	0.138
PSO	N/C	0.235	-0.121	0.599	0.151
GA	N/C	0.586	0.158	1.093	0.234

We report the comparison results in Table 4, where the solution quality is measured by the objectives collected for all test samples, and we report their mean, minimum, maximum and standard deviation value for simplicity. The smaller the objective value, the better the solution quality. The computation time is measured by the time to converge in seconds.

We see from Table 4 that only IPC, DE, and SQP can provide a converged solution within the given time interval, while PSO and GA cannot. IPC outperforms DE and SQP on both runtime and result quality significantly. This is expected, as IPC is a most suited optimization algorithm for the special mathematical structure as derived in this work, while other algorithms are generic optimization techniques.

Based on the same prediction model, we observe that solutions from IPC based control are able to reduce the temperature standard deviation by 42.5%, and O_2 content difference by 61.5% when compared to the the original test data without optimization. At the same time, we see 32℃ higher average temperature and 38.6% lower average O_2 content, indicating that the proposed model can also improve combustion efficiency simultaneously.

6 CONCLUSIONS

Equipped with the unique and largest dataset collected from a real power plant, we introduce a new formulation for boiler control problem that focuses on maintaining not only high temperature and low O_2 content, but also a balanced distribution of temperature and O_2 content. To overcome the foremost challenge of developing a real-time solution, we propose a new algorithmic framework that incorporates a machine learning-based prediction model, an optimization model, and an error compensation model. Experimental results based on a real boiler data validate the effectiveness and efficiency of the solution. The solution framework can be extended to other applications whose online control or optimization is constrained by the complexity of prediction and its formulation.

ACKNOWLEDGMENTS

The authors would like to acknowledge Mr. Hongwei Xu of Beijing Palmary Technology Ltd for explaining the boiler temperature control problem and providing related dataset for experimentation.

REFERENCES

[1] A Conn, Nick Gould, and Ph Toint. 1997. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Mathematics of Computation of the American Mathematical Society* 66, 217 (1997), 261–288.

[2] Luis I Díez, Cristóbal Cortés, and Antonio Campo. 2005. Modelling of pulverized coal boilers: review and validation of on-line simulation techniques. *Applied Thermal Engineering* 25, 10 (2005), 1516–1533.

[3] Norman Richard Draper, Harry Smith, and Elizabeth Pownell. 1966. *Applied regression analysis*. Vol. 3. Wiley New York.

[4] Wang Dong Feng, Meng Li, Meng Li, and Han Pu. 2009. Combustion optimization based on RBF neural network and multi-objective genetic algorithms. In *Genetic and Evolutionary Computing, 2009. WGEC'09. 3rd International Conference on*. IEEE, 496–501.

[5] J Chris Foreman. 2008. *Architecture for intelligent power systems management, optimization, and storage*. University of Louisville.

[6] Jacek Gondzio. 1996. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications* 6, 2 (1996), 137–156.

[7] Nick Gould and Philippe L Toint. 2004. Preprocessing for quadratic programming. *Mathematical Programming* 100, 1 (2004), 95–132.

[8] Yanping Gu, Wenjie Zhao, and Zhansong Wu. 2011. Online adaptive least squares support vector machine and its application in utility boiler combustion optimization systems. *Journal of Process Control* 21, 7 (2011), 1040–1048.

[9] Hasril Hasini, Mohd Zuki Yusoff, Norshah Hafeez Shuaib, Mohd Hariffin Boosroh, and Muhammad Alhafiz Haniff. 2009. Analysis of flow and temperature distribution in a full scale utility boiler using CFD. In *Energy and Environment, 2009. ICEE 2009. 3rd International Conference on*. IEEE, 208–214.

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

[11] NeuCo Inc. 2017. Boiler optimization software solution. Retrieved May 22, 2018 from http://neuco.net/solutions-boileropt.asp

[12] Andrew Kusiak, Alex Burns, and Ferman Milster. 2005. Optimizing combustion efficiency of a circulating fluidized boiler: A data mining approach. *International Journal of Knowledge-based and Intelligent Engineering Systems* 9, 4 (2005), 263–274.

[13] Andrew Kusiak and Zhe Song. 2006. Combustion efficiency optimization and virtual testing: A data-mining approach. *IEEE Transactions on Industrial Informatics* 2, 3 (2006), 176–184.

[14] Guoqiang Li and Peifeng Niu. 2016. Combustion optimization of a coal-fired boiler with double linear fast learning network. *Soft Computing* 20, 1 (2016), 149–156.

[15] Guo-Qiang Li, Xiao-Bin Qi, Keith CC Chan, and Bin Chen. 2017. Deep Bidirectional Learning Machine for Predicting NO x Emissions and Boiler Efficiency from a Coal-Fired Boiler. *Energy & Fuels* 31, 10 (2017), 11471–11480.

[16] Xiangjie Liu, Ping Guan, and CW Chan. 2010. Nonlinear multivariable power plant coordinate control by constrained predictive scheme. *IEEE transactions on control systems technology* 18, 5 (2010), 1116–1125.

[17] Efrñn Mezura-Montes, Jesús Velázquez-Reyes, and Carlos A Coello Coello. 2006. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 485–492.

[18] Jorge Nocedal and Stephen J Wright. 2006. *Sequential quadratic programming*. Springer.

[19] Ho Young Park, Se Hyun Baek, Young Ju Kim, Tae Hyung Kim, Dong Soo Kang, and Dong Woong Kim. 2013. Numerical and experimental investigations on the gas temperature deviation in a large scale, advanced low NOx, tangentially fired pulverized coal boiler. *Fuel* 104 (2013), 641–646.

[20] Mykola Pechenizkiy, Jorn Bakker, I Žliobaitė, Andriy Ivannikov, and Tommi Kärkkäinen. 2010. Online mass flow prediction in CFB boilers with explicit detection of sudden concept drift. *ACM SIGKDD Explorations Newsletter* 11, 2 (2010), 109–116.

[21] Xianyong Peng and Peihong Wang. 2009. An improved multiobjective genetic algorithm in optimization and its application to high efficiency and low NOx emissions combustion. In *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*. IEEE, 1–4.

[22] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*. 2951–2959.

[23] Marcin Szega and Grzegorz Tadeusz Nowak. 2015. An optimization of redundant measurements location for thermal capacity of power unit steam boiler calculations using data reconciliation method. *Energy* 92 (2015), 135–141.

[24] Ioan Cristian Trelea. 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters* 85, 6 (2003), 317–325.

[25] Vladimir Vapnik. 2013. *The nature of statistical learning theory*. Springer Science & Business Media.

[26] Chunlin Wang, Yang Liu, Song Zheng, and Aipeng Jiang. 2018. Optimizing Combustion of Coal Fired Boilers for Reducing NOx Emission using Gaussian Process. *Energy* (2018).

[27] World Coal Association. 2016. *Coal and Electricity*. Technical Report. World Coal Association. http://www.worldcoal.org/coal/uses-coal/coal-electricity

[28] Wang Xu and Chang Taihua. 2013. The balanced model and optimization of NOx emission and boiler efficiency at a coal-fired utility boiler. In *Conference Anthology, IEEE*. IEEE, 1–4.

[29] Chungen Yin, Lasse Rosendahl, and Thomas J Condra. 2003. Further study of the gas temperature deviation in large-scale tangentially coal-fired boilersâŸĘ. *Fuel* 82, 9 (2003), 1127–1137.

[30] G Zhang, R Veale, T Charlton, B Borchardt, and R Hocken. 1985. Error compensation of coordinate measuring machines. *CIRP Annals-Manufacturing Technology* 34, 1 (1985), 445–448.

[31] Shu Zhang, Cyrus W Taft, Joseph Bentsman, Aaron Hussey, and Bryan Petrus. 2012. Simultaneous gains tuning in boiler/turbine PID-based controller clusters using iterative feedback tuning methodology. *ISA transactions* 51, 5 (2012), 609–621.

[32] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. *arXiv preprint arXiv:1404.5772* (2014).

[33] Yi Zhang, Yanjun Ding, Zhansong Wu, Liang Kong, and Tao Chou. 2007. Modeling and coordinative optimization of NO x emission and efficiency of utility boilers with neural network. *Korean Journal of Chemical Engineering* 24, 6 (2007), 1118–1123.

[34] Huan Zhao and Pei-hong Wang. 2009. Modeling and optimization of efficiency and NOx emission at a coal-fired utility boiler. In *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*. IEEE, 1–4.

[35] Wenjie Zhao, Gang Zhao, Meng Lv, and Jianjun Zhao. 2015. Fuzzy optimization control for NOx emissions from power plant boilers based on nonlinear optimization 1. *Journal of Intelligent & Fuzzy Systems* 29, 6 (2015), 2475–2481.

[36] Ligang Zheng, Hao Zhou, Chunlin Wang, and Kefa Cen. 2008. Combining support vector regression and ant colony optimization to reduce NOx emissions in coal-fired utility boilers. *Energy & Fuels* 22, 2 (2008), 1034–1040.