Online Mass Flow Prediction in CFB Boilers with Explicit Detection of Sudden Concept Drift

M. Pechenizkiy, J. Bakker, I. Žliobaitė
Dept. of CS, TU Eindhoven
P.O. Box 513, NL-5600MB
the Netherlands
{m.pechenizkiy,j.bakker,i.zliobaite}@tue.nl

A. Ivannikov, T. Kärkkäinen
Dept. of MIT, U. Jyväskylä
P.O. Box 35, FIN-40014
Finland
aivanni@cc.jyu.fi,tka@mit.jyu.fi

ABSTRACT

Fuel feeding and inhomogeneity of fuel typically cause fluctuations in the circulating fluidized bed (CFB) process. If control systems fail to compensate the fluctuations, the whole plant will suffer from dynamics that is reinforced by the closed-loop controls. This phenomenon causes reducing efficiency and the lifetime of process components. In this paper we address the problem of online mass flow prediction, which is a part of control. Particularly, we consider the problem of learning an accurate predictor with explicit detection of abrupt concept drift and noise handling mechanisms. We emphasize the importance of having domain knowledge concerning the considered case and constructing *the ground truth* for facilitating the quantitative evaluation of different approaches. We demonstrate the performance of change detection methods and show their effect on the accuracy of the online mass flow prediction with real datasets collected from the experimental laboratory-scale CFB boiler.

1. INTRODUCTION

Continuous growth and increase of variance in electricity consumption can lead to frequent load changes. This calls for novel control concepts in order to minimize emissions and to sustain high efficiency during load changes.

From combustion point of view the main challenges for the existing boilers are caused by a wide fuel selection, increasing share of low quality and bio fuels, and co-combustion. In steady operation, combustion is affected by the disturbances in the feed rate of the fuel and by the incomplete mixing of the fuel in the bed. It may cause changes in the burning rate, oxygen level and increase CO_2 emissions. This is especially relevant for the new biomass fuels, which have increasingly been used to replace coal. The bio-fuels are rather inhomogeneous and very reactive in comparison with coal.

Traditionally, mathematical models of CFB boiler operation, have been developed [13; 10; 14], incorporating operational parameters in the models. More recently, data mining approaches were considered for developing better understanding of the underlying processes in CFB boilers, or learning a model to optimize its efficiency [9].

In our work we focus on a data driven approach for online mass flow estimation. It is required for efficient control of the boiler. Online estimation of fuel consumption in mechanical devices is a challenging task due to noise, presence of outliers and non-stationarity of the signal. Mechanical devices typically comprise of a number of moving parts. The movements of these parts cause interference in the observed sensor signal. The challenge is to filter out the *true* signal from the measured noise. In this study we investigate the online estimation of the fuel mass inside a CFB boiler.

Different amounts of fuel can be added to the boiler at irregular time intervals resulting in sudden drifts in a signal. Since the fuel feeding is done mechanically, the feeding start and end time is not necessarily (as in our case) available from the sensors as a direct measurement. Hence, in order to estimate accurately the amount of fuel in the container at each moment in time we should be able explicitly or implicitly handle these changes.

There is a lot of work on change detection and outlier detection, see e.g. a recent review [3]. However, the boiler problem exposes specific combination of change points and outliers at which existing change detection methods may fail. Statistical change detection methods, which are based on comparing chunks of raw data (e.g. [2]) do not take signal trends into account. The noise and outliers are not evenly distributed making it hard to use statistical methods that assume a particular distribution of the data [1]. Change detection (e.g. [4]) based on the streaming learner accuracy is not directly suitable for this problem. Due to the nature of the signal (noise, trends and specific outliers) the streaming accuracy is hard to benchmark. Burning and feeding stages, observed in the fuel mass signal, are very different in nature and timing.

In this paper we present the revision of our recent work [1] where the performance of change detection methods was compared. We consider also a tailor made online method for the signal prediction presented in [15] and do a thorough quantitative evaluation and comparison of alternative prediction methods on real datasets[1].

The obtained results, evaluated by the domain experts, indicate that our approach is appropriate and that an acceptable level of prediction accuracy can be achieved with respect to the application requirements. Our case study illustrates however that both quantitative and qualitative evaluation is necessary, and that developing a data mining solution is a truly iterative and interactive process in these settings, including elements of unsupervised and supervised learning. The rest of the paper is organized as follows. In Section 2

[1]an extended version of this paper is accessible at `http://www.win.tue.nl/~mpechen/projects/cfb/`

we overview the problem of a mass flow prediction in CFB boiler. In Section 3 we present our solution for online mass flow prediction. In Section 4 the experimental evaluation is presented and the results are discussed. We conclude and point out open problems in Section 5.

2. PROBLEM DESCRIPTION

To better understand and control the operation of CFB boiler it is important to know how much fuel mass is in the container. Direct measurement is hardly possible in practice from the technological perspective. Therefore, fuel mass is calculated from estimates of a mass flow in the system. That is equivalent to predicting the amount of fuel in the fuel feeding system at each point in time, which we adress in this paper. We start by briefly explaining how the input signal is generated, discuss the properties of the data and available solutions.

2.1 The Input Signal

The automatically available mass signal is a noisy estimate of fuel mass at each operation time point. The mass of the fuel inside the container is measured by a scale, sampled with a sample rate of 1 Hz.

The boiler is fed with fuel from the fuel container (bunker) as depicted in Figure 1. The fuel inside the container is mixed using a mixing screw. There is a feeding screw at the outlet of the container, which transfers the fuel from the container to the boiler. During the burning stage the mass of fuel inside the container decreases (reflected by a decreasing amount of fuel in the data signal). As new fuel is added to the container (the burning process continues), the fuel feeding stage starts that is reflected by a rapid mass increase.

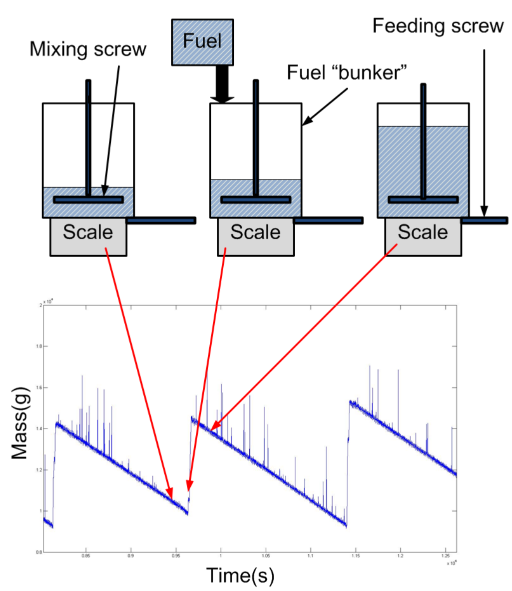

Figure 1: The origin of the input signal.

There are three main sources of changes in the signal:

1. Fuel feeding is manual and non standardized process, which is not necessarily smooth, it can have short interruptions. Each operator can have different habits. Besides, the feeding speed depends on the type of fuel used.
2. The feeding screw rotation adds noise to the measured signal. Besides, fuel particle jamming often happens, slowing down the screw for some seconds and distorting the signal estimate. Therefore, the reported mass inside the bunker is not accurate, the signal contains extreme upward outliers in the original signal, that can be seen in Figure 2.
3. There is a low amplitude rather periodic noise, which is caused by the mechanical rotation of the system parts. These amplitudes may become higher depending on the burning setup.

2.2 Data Properties

Due to the processes described above, the fuel mass signal has the following characteristics:

- There are two types of change points: an abrupt change to feeding and slower but still abrupt switch to burning.
- There are asymmetric outliers (see Figure 2 left), oriented upwards, which in online settings can be easily mixed with the changes to feeding.
- There is a symmetric high frequency signal noise.

Algorithmic change detection is not trivial as it might seem from visual inspection of the signal. The asymmetric nature of the outliers would elevate the original signal if approximated directly, since there are no corresponding negative outliers. In other words, the noise and outliers do not sum to zero with respect to the *true* signal.

Besides, there are short burning periods within feeding stages, due to possible pauses in feeding (see Figure 2 right), which depend on human factor. These interruption regimes can vary from 5 to 20 seconds are difficult to discriminate.

In addition, we need to take into account that the mass flow signal may have a nonzero second derivative, i.e. the speed of the mass change depends on the amount of fuel in the container - the more fuel is in the container, the higher is the acceleration, thus the more fuel gets into the screw. The weight of the fuel at higher levels of the tank compresses the fuel in the lower levels and in the screw, and the fuel density is increased. Besides, compression and thus the burning speed depends on the type and quality of the fuel.

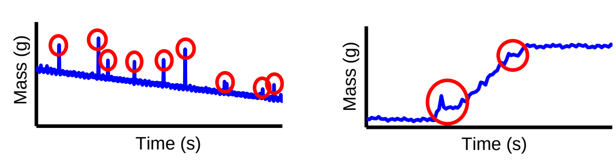

Figure 2: Peculiarities in the data: upward outliers (left) and short burning periods within the feeding stage (left).

3. ONLINE MASS FLOW PREDICTION

In this section we present our solution to online mass flow prediction. We start with setting up a general framework, followed by depicting the base model, change and outlier detection mechanisms.

3.1 General Framework

Let us define the original signal as $\mathbf{x} = (x_1, x_2, \ldots, x_t, \ldots, x_n)$. Having $\mathbf{x}$ as input we want to obtain the actual mass flow signal $\mathbf{y}$ that can be achieved by learning a functional mapping, so that $\mathbf{y} = \mathcal{F}(\mathbf{x})$.

This problem has a connection to the problem of *concept drift* [7] that refers to unforeseen changes over time in the phenomenon of interest. Our phenomenon of interest here is the true signal, or actually the concept we aim to learn is the functional mapping $\mathcal{F}$ of noisy sensor measurements to the true actual signal.

Once a change in the system stage happens (reasons are described in Section 2) the functional mapping $\mathcal{F}$ might become outdated. The learners capable of handling concept drift can be classier into proactive (explicitly detecting the change and dropping out the old training sample) or reactive (using forgetting heuristics at each time step to have the best adapted learner) [8]. The boiler data exhibits abrupt changes, thus we employ a proactive approach.

The intuition behind the model is the following: at each point in time t we fit a model $\mathcal{F}(x)$, using all or a subset of the historical data $\mathbf{x}$. If a change is detected, the old portion of the historical data is dropped out. A simplified estimation procedure is presented in Figure 3, the steps are explained in more detail in the following subsections. We start with selecting the functional mapping $\mathcal{F}$ corresponding to the step 3 and then consider methods for detecting sudden changes corresponding to the step 4.

Online mass flow prediction for a time point $t+1$

input: historical signal $\mathbf{x} = \{x_1, \ldots, x_t\}$.

1. Replace outliers with an average of neighboring points.
2. Find the last change point c.
3. Learn the model $\mathcal{F}(x)$ from $\{x_c, \ldots, x_t\}$.
4. Cast the prediction $\hat{y}_{t+1} = \mathcal{F}(x_{t+1})$.

output: $\hat{y}_{t+1}$.

Figure 3: Online Mass Prediction.

3.2 The Predictor

We assume that the mass flow signal has a nonzero second derivative. The nature of the measured phenomena *in a single stage* can be modeled using the following equation:

$$y_t = \frac{a \cdot t^2}{2} + v_0 \cdot t + m_0 + A \cdot \sin(\omega_{feed} \cdot t + \alpha_{feed}) + + B \cdot \sin(\omega_{mix} \cdot t + \alpha_{mix}) + e(t), \quad (1)$$

where y_t denotes the output of the scales at time t, a is acceleration of the mass change, v_0 stands for the speed of the mass change at time t_0, m_0 is the initial mass at time t_0; A and B, ω_{feed} and ω_{mix}, α_{feed} and α_{mix} are amplitude, frequency and phase of the fluctuations caused by the feeding and mixing screws, respectively; $e(t)$ denotes the random peaked high amplitude noise caused by the jamming of the fuel particle at time t. Note that here we assume t_0 was the time of switch from the feeding stage to burning or other way around.

Since we are not interested in estimating the signal generated by the oscillations of the screw and the noise signal, we make a simplifying assumption that these parts can be treated as noise. Thus we choose the following model:

$$\hat{y}_t = \frac{a \cdot t^2}{2} + v_0 \cdot t + m_0 + E(t), \quad (2)$$

where $E(t)$ is the aggregated noise component.

In our estimator we use a linear regression approach with respect to the second order polynomial given by (2). The model is inspired by the domain knowledge of the underlying process in the boiler, therefore seem more reasonable choice than alternative autoregressive models.

3.3 Learning the Predictor

To learn a regressor, the Vandermonde matrix [5] $\mathbf{V}$, which elements $v_{i,j}$ are the powers of independent variable x, can be used. In our case the independent variable is time $x_i = t_{i-1} - t_0$, $i = 1, \ldots, T$, where T denotes the number of the time samples. If the linear regression is done for a polynomial of order n ($p^n(x) = p_n x^n + p_{n-1} x^{n-1} + \ldots + p_1 x + p_0$), $\mathbf{V}$ is computed from the observed time series of the independent variable as follows:

$$v_{i,j} = x_i^{n-j+1}, \; i = 1, \ldots, T, \; j = 1, \ldots, n+1, \quad (3)$$

where i and j run over all time samples and powers, respectively. Provided with $\mathbf{V}$ the problem of polynomial interpolation is solved by solving the system of linear equations $\mathbf{V}\mathbf{p} \cong \mathbf{y}$ with respect to $\mathbf{p}$ in the least square sense:

$$\widehat{\mathbf{p}} = \text{argmin}_{\mathbf{p}} \sum_{i=1}^{T} \left(\sum_{j=1}^{n+1} V_{i,j} p_{n-j+1} - y_i \right)^2 \quad (4)$$

Here, $\mathbf{p} = [p_n \; p_{n-1} \; \ldots \; p_1 \; p_0]^{\mathrm{T}}$ denotes the vector of the coefficients of the polynomial, and $\mathbf{y} = [y(x_1) \; y(x_2) \; \ldots y(x_T)]^{\mathrm{T}} = [y_1 \; y_2 \; \ldots y_T]^{\mathrm{T}}$ is the time series of the dependent variable that is indication of the scales. Provided that the $n+1$ columns of the matrix $\mathbf{V}$ are linearly independent, this minimization problem has a unique solution given by solving the normal equation [11]:

$$(\mathbf{V}^{\mathrm{T}}\mathbf{V})\widehat{\mathbf{p}} = \mathbf{V}^{\mathrm{T}}\mathbf{y}. \quad (5)$$

This procedure is used to estimate the mass flow signal between change points. If the process switches from fuel feeding to fuel burning or the other way around, a new model is learnt on the new data.

3.4 Detection of Sudden Concept Drift

We consider four different approaches to detect change points and thus facilitate modeling the transitions from one state of the system to another. The first two approaches are based on statistical control of predictor performance. One approach is nonparametric and is based on the Mann-Whitney U test [12] and another is based on a parametric test on the performance of the local models. The other two approaches use the raw data for change detection. The first approach is ADWIN [2] which checks whether there are statistically significant differences between the means of each possible split of the sequence. The second approach is based on a heuristic tailored to the peculiarities of our problem.

3.4.1 Performance monitoring-based detection

Let's assume that when the underlying process in the boiler changes, the current model for the prediction of the data will perform worse. The idea behind (non)parametric change

test is to compare the prediction performances obtained on different subsets of the data.
When a new point arrives, it is tested for being an outlier (peak due to a fuel particle jamming) with respect to the preceding points. If it is classified as outlier, it is replaced by an extrapolation from these points. If outliers follow one another in a row, the state change is alarmed (pointing to the first outlier in the row) and the detected 'false outliers' that actually belong to the after-change period are restored from the backup buffer. The process continues then from the beginning of the newly detected phase.

Parametric approach. Deterioration in the performance of the current model can be detected by keeping track of two statistical properties of the performance [4]. The first is the error rate that signifies the probability of miss classifying the actual value y_t of the signal. The second is the standard deviation of the error rate. In [4] it is assumed that the classification task can be modeled by a binomial distribution. Since the mass flow is continuous, we have to assume that given a large enough sample or window, the binomial distribution is close to a normal distribution with the same mean and standard deviation. This is used in an online test to see whether the signal has changed.
Since this task is in continuous space, we use the Mean-Squared-Error (MSE) as metric for change detection instead of the error rate. For each time step the window is moved and for point x_t all local reference MSEs are calculated with a leave-one-out cross validation (LOOCV).
We assume that the model prediction performance is stable if the local found reference MSE (E_t) satisfies

$$E_t + S_t < E_{min} + \alpha S_{min}, \tag{6}$$

where E_{min} is the minimal found reference MSE, S_{min} the minimal found reference standard deviation, and α a parameter that reflects the level of confidence. If there is a large enough deviation in the signal the algorithm will report a warning. This level is determined by the condition:

$$E_{min} + \alpha S_{min} < E_t + S_t < E_{min} + \beta S_{min}, \tag{7}$$

where β is the upper confidence bound signifying a change. Since it might occur that there is a local change in the signal (an outlier) it is not possible to go the change state immediately. If $E_t + S_t > E_{min} + \beta S_{min}$ the algorithm reports a change and it will switch to the initial state of the other process. When this happens E_{min} and S_{min} are reset to the minimal found values in the new regime.
This procedure puts a strict lower bound on the window size. Since we have to assume that the normal distribution is representative for the distribution of reference MSEs, the window size should be at least 30 consecutive points. In principle, a small window size is preferable when trying to detect rapid changes. But a smaller window size will result in a higher variation in the local models. In our experiments we used $\alpha = 2$, $\beta = 3$ (lower and upper bounds for the confidence interval), and the window size of 30 points.
For each 30 accumulated points, the LOOCV is used. For each $N-1$ local model the MSE and it's standard deviation are calculated. The transition conditions are checked and when an outlier is detected it is ignored by the global fit. If, after this, the algorithm detects a change, the boundaries for the transition criteria are reset and the global fit is relearnt for the future points.

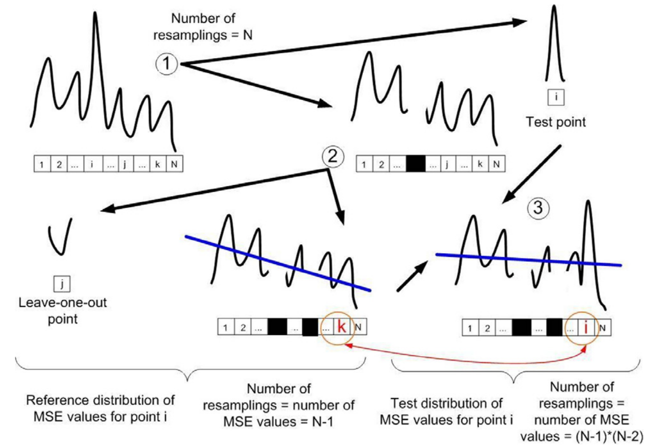

Figure 4: The resampling procedure for constructing two distributions to be compared by Mann-Whitney U test.

Nonparametric approach. The Mann-Whitney U test [12] applied to the reference and test distributions for a given confidence level α shows whether MSE of the fitted points including test point is systematically different from the case when the test point is excluded. If such tendency is detected by the test, the test point is classified as outlier.
The resampling procedure for constructing two distributions are demonstrated in Figure 4. We start with an outer LOOCV cycle, in which each point in the initial window of size N is made a test point, and the rest form reference set. Then, an inner LOOCV is done on a reference set of $N-1$ points. A line is fitted to each of the possible subsets of size $N-2$, and MSE of the fit is computed for fitted $N-2$ points only. Thus, at this stage we obtain a sample of $N-1$ MSE values that characterizes the set of reference points. We call the distribution of these MSE values a reference distribution for the tested point. Now, the test distribution composed of $(N-1)(N-2)$ MSE values is computed by replacing each point in each subset of size $N-2$ from previous stage for a test point, computing a linear fit and MSE value for the $N-2$ points used to compute the fit. Therefore, the test distribution contains MSE values for a linear fit, when a test point is used. (This way we ensure that the fits and MSE computations are always done using $N-2$ points, but the sizes of reference and test samples are different.)
Note that in Figure 4 a less general case is shown compared to a discussion. The illustration considers a set of neighboring points, although in general the points in the set can be drawn from distanced time locations depending on the pursued goal. By changing the size and spread of the reference set, one controls how the decision made using the Mann-Whitney U statistics is tailored/diversified over time. In other words, varying the window size adjusts the context for decision making to be more local, global, or balanced.

3.4.2 Detection using raw data

To reduce the degrees of freedom introduced by MSE based change detectors, we employ change detection methods based on raw data. We consider ADWIN [2] and a heuristic approach tailored to our problem [15].

ADWIN method was originally designed for univariate sequential data. The method works as follows: given a sequence of signals it checks whether there are statistically significant differences between the means of each possible split of the sequence. If a statistically significant difference is found, the oldest portion of the data backwards from the detected point is dropped and the splitting procedure is repeated until there are no significant differences in any possible split of the sequence. More formally, suppose m_1 and m_2 are the means of the two subsequences as a result of a split. Then the criterion for a change detection is $|m_1 - m_2| > \epsilon_{cut}$, where

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \log \frac{4n}{\delta}}, \quad (8)$$

$$m = \frac{1}{\frac{1}{n_1} + \frac{1}{n_2}}, \quad (9)$$

here n is total size of the sequence, while n_1 and n_2 are sizes of the subsequences respectively. Note that $n = n_1 + n_2$. $\delta \in (0, 1)$ is a hyperparameter of the model. In our experiments we used $\delta = 0.3$, $n = 200$.

Heuristic approach. We design an online signal prediction approach, which takes into account the properties of mass flow signal (noise, trends, specific outliers, switch between operational stages).

An intuitive solution for detecting the feeding stages, which are characterized by a steep increase in the signal value, would be to take the first order differences of the signal $d_t^{(1)} = x_t - x_{t-1}$ and threshold these values. If $d_t^{(1)} > 0$ the system is in feeding stage, if $d_t^{(1)} < 0$ the system is in burning stage.

Unfortunately, due to noise, the stages are undistinguishable directly (see Figure 5a). We can try replacing the original signal with the moving average, before taking the first order differences, this already gives apparent feed regions, but that still is noisy (see Figure 5b).

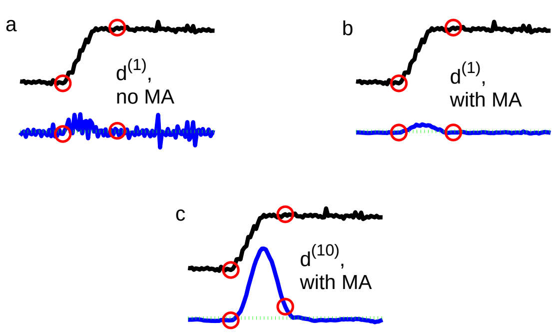

Figure 5: Change detection using L^{th} order signal differences $d^{(L)}$ and moving averages (MA). The upper (black) line represents the original signal and the lower (blue) is the differentiated signal. Dashed line (green) is the threshold for a change. Circles indicate *the ground truth* change.

We propose using L^{th} order differences $d_t^{(L)} = x_t - x_{t-L}$, applied to a moving averaged signal for the detection of stage changes. The more noisy the signal is, the larger lag is needed. In this case study we use $L = 10$ (see Figure 5c) and threshold feeding stage at $Th_{ch} = 100$ to avoid false positives.

3.5 Validation

For evaluation of the performance of signal estimators labeled data is needed. There is no hard evaluation method for the actual amount of fuel present. It could be generated by the domain experts. It is difficult to extract the actual signal, since the data includes the effects of external influencers. In our work we suggest using an offline best fit method as internal validation for the estimators.

4. EXPERIMENTAL STUDY

In this section we first illustrate the data available for out experiments, discussion methodological issues related to evaluation and proceed with the presentation of the main results of this study.

4.1 Datasets

We use three sensor signal datasets obtained from experimental CFB boiler. The datasets A,B and C are plotted in Figure 6. They represent different types of fuel. The summary of the data properties is provided in Table 1.

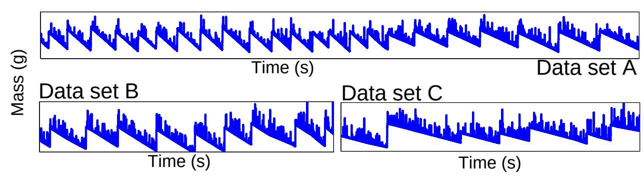

Figure 6: Datasets A, B, and C used in the experiments.

Table 1: Datasets used

Name	Size	Feeding stages	Fuel
A	50 977	24	bio
B	25 197	9	bio
C	25 197	6	coal

4.2 Qualitative Evaluation

We pretested our predictor with alternative change detectors using dataset A. We conducted qualitative evaluation of considered approaches by visual inspection of predicted signal against the raw data.

Change detectors showed varying performance. Nonparametric and tailored heuristic outperformed parametric approach. The ADWIN method was very precise in the detection of change from burning to feeding, but the lag of detection was quite large (60 s). Therefore this method could be used only as a reference for the *the ground truth*, but is not applicable for the online operational settings of the CFB boiler.

Based on visual inspection, our predictor performed reasonably well. A delay of up to 5 seconds allow verification before using the result as an indicator for the CBF control system [6].

4.3 Quantitative Validation

Next we present quantitative validation of the change detection and signal prediction performance. Online mass flow prediction is an unsupervised learning task. The need for prediction arises from the fact that there is no method to measure *the ground truth*. However, to verify the validity of the model we still need a benchmark. To obtain an approximation to *the ground truth* we use the following.

Construction of the ground truth. We know that the outliers are oriented upwards. We identify the outliers by by comparing the difference between a point of the signal and the moving average against a threshold Tr_{out}. Then we take a moving average of the signal with removed upward oriented outliers to obtain an approximation to *the ground truth*, which we associate with y, so predictions can be compared against *the ground truth* by computing mean absolute errors, $MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t|$, where $e_t = y_t - \mathcal{F}(x_t)$.
Next we identify the change points from burning to feeding stage and vice versa (C_{feed} and C_{burn}). We employ ADWIN, which showed to be robust to false positives in semi-online settings. ADWIN identifies C_{feed} approximately. To get the exact change points we search for a maximum and minimum of the moving average in the neighborhood of the points identified by ADWIN. We validate *the estimated ground truth* by visual inspection of a domain expert and introduce changes where necessary.

Experimental setup. We use dataset A for training and parametrization of the model. Datasets B and C are used as test sets, applying the model trained on A with identical parameterization. Note that the level of noise and outliers in the datasets are different. B and C represent two fuel tanks, operating in parallel, therefore there are nearly twice as much noise sources as in A.
We conduct a set of experiments allowing a delay D in predictions. E.g. having $D = 10$ (maximum possible delay suggested by the domain experts) we would predict (filter) the signal x_t, but will have the historical data available up to time x_{t+9} inclusive. This gives smoother moving average (nearest neighbors from both sides are available) and it also allows verification of outlier and change detection.
We do the following verification: the stage (feeding or burning) is defined to be consistent if it lasts for not less than D time steps. Say at time t the system is at burning stage and at time $t+1$ we detect a feeding stage. Having a delay $D = 10$ we are able to see the next four examples before casting the signal prediction for time $t+1$. Thus we check if the feeding stage sustains at time $t+2, \ldots, t+10$. If positive, we fix the change point, if negative, we cancel the detected change and treat this as an outlier.
Once a change is detected, old portion of the data is dropped out of the training sample. We do not use the 2^{nd} order polynomial model until we pass 10 samples after the change. For the first 2 samples we use simple moving average rule: $x_{t+1} = x_t + s$, where s is a linear intercept term obtained using an average feeding stage pattern of the training dataset A. For burning stage $s_c = -2$ is used, for feeding stage $s_f = 81$. If more than 2 but less than 11 historical data points are available after the change, we fit the 1^{st} order polynomial model.
We test for prediction accuracy and for change detection accuracy.

4.3.1 Change detection accuracies

We report the performance of the change detection in online settings for the nonparametric method, which has shown the most robust behavior, in Table 2. We present confusion matrixes of detecting sudden changes to feeding (φ) and burning (β) stages and detecting of outliers (o). For φ and β we allow ± 10 sec deviation, i.e. if a change is detected within the allowed region it is considered as identified correctly. For outliers we require the identification to be precise.

Table 2: Accuracy of detecting changes to feeding (φ) and burning (β) stages and outlier detection (o).

Cumalative over test datasets B and C									
	φ	P	N	β	P	N	o	P	N
t-2	T	12	50376	T	11	50375	T	816	48532
	F	3	3	F	4	4	F	994	52
t-4	T	12	50376	T	11	50375	T	819	48525
	F	3	3	F	4	4	F	1001	49
t-9	T	12	50376	T	11	50375	T	828	48499
	F	3	3	F	4	4	F	1027	40

The number of FPs is decreasing along with the increase in allowed prediction delay. A delay allows to inspect the following signal values after the detected change and, if necessary, cancel the alarm within the delay period.
We visualize change and outlier detection accuracies for three different approaches across datasets B and C in Figure 7 showing the tradeoffs between TP and FP (top plot), and TN and FN (bottom plot) rates.

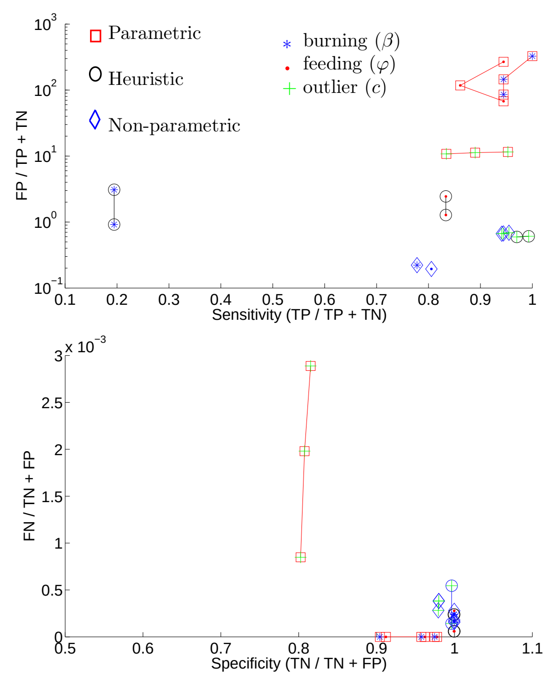

Figure 7: TP vs. FP (top) and TN vs. FN (bottom) tradeoffs with respect to two groups of changes points and outliers for heuristic, parametric and nonparametric approaches over the test datasets B and C (in total).

Each point has an inner shape denoting type of point to be detected (change in a feeding stage, change in a burn-

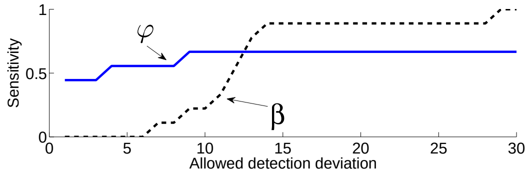

Figure 8: Sensitivity of the heuristic method as a function of allowed change detection deviation for two different stages.

ing stage or an upwards oriented outlier) and outer shape denoting the corresponding detection method (parametric, nonparametric or heuristic). Points having the same inner and outer shape are connected with lines such that they form a 'trace' of points corresponding to different lags ($t+1$, t, $t-1$, etc.).

Both the heuristic and the nonparametric methods are accurate in identifying outliers (sensitivity over 90%) and change points in burning stages (about 80%), heuristic approach seeming to be slightly ahead. However, the heuristic method has rather low TP rate with respect to change detection in feeding stages[2], while the sensitivity of the nonparametric method here is more than 80% and the number of FPs is very small. The specificity of both methods is close to 100%.

The parametric method is worse in detecting the changes generating the highest amount of FPs (e.g. roughly 300 FPs to 6 or 9 detected changes). The method is too sensitive and reports a change point roughly every 10 seconds. This does result in good accuracies for the prediction (since it is closely following the original signal), despite having a lot of local variation and relatively low TN rate (see Figure 7).

4.3.2 *Prediction accuracies*

We present MAE's for the whole datasets and for feeding and burning stages separately in Table 3.

MA stands for simple prediction by moving averages (over 3, 5 and 10 points for $t-2$, $t-4$ and $t-9$ correspondingly), the number indicates how many instances are averaged. 'win50' uses the 2^{nd} order prediction model presented in Section 3.2, but instead of change detection a simple moving window of the 50 last instances is used for the model training at each time step. 'all' uses the 2^{nd} order prediction model with no change detection at all, it retrains the model at every time step. Finally we include a benchmark of the 2^{nd} order model assuming known change points ('known'). We assume with this method that the change detection is 100% accurate. MAE in 'overall performance' is rather close to MAE in 'burning stages' and very different from 'feeding stages'. This is because of uneven distribution of the stages in the data. 'Burning stages' comprise less than 2% of the data.

Heuristic and nonparametric outperform the competitive methods in terms of overall accuracy, nonparametric being slightly ahead. However, for the feeding stage, simple moving average is the most accurate.[3] The parametric method also performs well with respect to MAE that is due to its hypersensitivity and poor specificity.

Table 3: MAE's for the test dataset (B and C). The best accuracies for each delay are **bold**; the best overall accuracy over a single experiment is underlined.

Delay	t-2	t-4	t-9	t-2	t-4	t-9
Data		B			C	
Overall performance						
MA	46.9	41.7	34.9	35.2	32.5	28.5
Par	18.6	23.1	18.5	10.9	13.4	11.5
Nonpar	**11.3**	**11.1**	**10.9**	**6.6**	**6.4**	**6.0**
Heuristic	16.6	16.3	31.4	10.3	10.1	16.3
win50	32.0	32.0	32.0	15.2	15.2	15.2
all	1308	1306	1301	1019	1019	1016
known	45.1	44.6	65.5	15.7	16.3	22.0
Feeding stages						
MA	**434**	**359**	**171**	**119**	**105**	**61**
Par	361.3	708.2	362.8	127.6	235.2	121.3
Nonpar	485	481.4	470.3	120.7	125.9	89.5
Heuristic	601	682	968	182	180	325
win50	1248	1248	1248	645	645	645
all	3248	3242	3225	2255	2250	2237
known	561	594	752	249	269	296
Burning stages						
MA	46.8	41.5	34.9	34.8	32.1	28.3
Par	17.4	21.3	17.4	9.8	11.5	10.5
Nonpar	**9.9**	**9.8**	**9.7**	**5.8**	**5.4**	**5.2**
Heuristic	16.9	17.2	34.9	10.5	11.3	19.1
win50	32.4	33.3	35.5	12.8	13.8	16.3
all	1315	1314	1311	1013	1013	1013
known	45.7	46.0	69.1	15.5	16.9	25.1

Surprisingly, heuristic method performance gets worse having a large delay in predictions. This is likely due to a fixed number of the nearest neighbors for moving average calculations, as we are using the same parameter settings for all the experiments. Such degradation also suggests, that there might be more accurate cutting points than just the change points themselves. Note that having a delay we allow canceling the detected changes.

Another interesting point is that the global fit in case of the known change points ('known') is worse than local fit with explicit change detection with respect to *our ground truth*. This happens due to the presence of false positives, i.e. more cuts (locally the square term approaches zero and we get a straight line local fit).

4.3.3 *Implications from the results*

Overall, the nonparametric method, despite of requiring more computational power to guarantee in time prediction, can be favored to other methods due to its robustness.

Generally, however, the results suggest that separate handling of prediction in feeding and burning stages is needed. In this case the heuristic method which showed better online change detection performance for burning stages is quite attractive from the domain perspective as it is also much faster

[2]Note that low TP rate in this setting means that change points were not detected *in time* (within 10 sec interval). Figure 8 shows how the sensitivity of the heuristic method improves with increase of allowed detection deviation.

[3]Note that the approximation to *the ground truth* was constructed using moving averages, thus it could be expected that moving average performs well in this test setup.

and more intuitive than the nonparametric method.

There is no trivial correlation between the accuracy of the change detection method and MAE of the eventual prediction. This has to do with the construction of *the ground truth* (e.g. local variance and the accuracy of sensitive methods) and the types of errors that dominate the change detection method. This indicates that having more training window cuts than a 'perfect' number, corresponding to the number of actual change points, might be beneficial. Having a stream of data there is no scarcity of training points and local fit might be accurate enough.

5. CONCLUSION

We developed an effective approach for the online mass flow prediction during the boiler operation. In our approach we try to learn a regressor from the raw sensor measurements and therefore we employed abrupt change detection and noise canceling mechanisms. This appeared to be a challenging task due to the signal properties and peculiarities of the data. We evaluated the performance of our approach and compared the accuracies of three change detection methods of different nature: statistical parametric, nonparametric and heuristic. We used real datasets from experimental CFB boiler, including two distinct fuel types and two distinct operating stages (single vs. multiple fuel).

One of the methodological challenges in this task is coming up with an approximation for constructing *the ground truth* for the signal, which we handle by a combination of moving average and responding for change and outlier points in the offline settings. We used this approximation to evaluate the performance of the online predictors. Anyhow, our experience shows that both quantitative and qualitative (i.e. visual inspective) evaluation of the performance are important. The process of developing a data mining solution for the problem at consideration appeared to be truly iterative and interactive.

We achieved sufficiently accurate detection of changes in transition from the burning to the feeding stage, where the incline in signal is rather sharp. However, the reverse detection still has room for improvement. It should be noted, that the change point at this stage is hard to distinguish even visually.

Overall, the domain experts found the achieved performance on the provided datasets acceptable and the next step of the research would be to employ the presented approach in operational settings when online mass flow estimation is used in the control system of the boiler.

However, it remains interesting to explore the effects of the feeding screw on the mass signal in the context of gradual drift detection and handling. Furthermore, it would be interesting to come up with different models for different fuel types and different operational contexts.

6. ACKNOWLEDGEMENTS

This research is partly supported by The Netherlands Organization for Scientific Research (NWO) HaCDAIS project, Finnish Funding Agency for Technology and Innovations (TEKES) DYNERGIA project, and LOIS visitor grant. The authors are thankful to the domain experts, Timo Leino and Mikko Jegoroff for their endless support and valuable discussions.

7. REFERENCES

[1] J. Bakker, M. Pechenizkiy, I. Zliobaite, A. Ivannikov, and T. Kärkkäinen. Handling outliers and concept drift in online mass flow prediction in CFB boilers. In *KDD Workshop on Knowledge Discovery from Sensor Data*, pages 13–22. ACM, 2009.

[2] A. Bifet and R. Gavaldà. Learning from time-changing data with adaptive windowing. In *Proc. of SIAM Int. Conf. on Data Mining (SDM 2009)*. SIAM, 2007.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[4] J. Gama and C. G. Learning with local drift detection. In *Proc. Int. Conf. on Advanced Data Mining and Applications (ADMA 2006)*, volume 4093 of *LNCS*, pages 42–55. Springer, 2006.

[5] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1991.

[6] A. Ivannikov, M. Pechenizkiy, J. Bakker, T. Leino, M. Jegoroff, T. Kärkkäinen, and S. Äyrämö. Online mass flow prediction in CFB boilers. In *Proc. of Industrial Conf. on Data Mining (ICDM 2009)*, volume 5633 of *LNCS*, pages 206–219. Springer, 2009.

[7] M. Kubat and G. Widmer. Adapting to drift in continuous domains. In *In Proc. of European Conf. on Machine Learning (ECML 1995)*, pages 307–310. Springer, 1995.

[8] L. I. Kuncheva. Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In *Proc. of ECAI Workshop SUEMA'08*, pages 5–10, 2008.

[9] A. Kusiak and A. Burns. Mining temporal data: A coal-fired boiler case study. In *Proc. of Int. Conf. on Knowledge-Based Engineering Systems (KES 2005)*, volume 3683 of *LNAI*, pages 953–958. Springer, 2005.

[10] B. R. C. Y. L. Huilin, Z. Guangboa and D. Gidaspowb. A coal combustion model for circulating fluidized bed boilers. *Fuel*, 79:165–172, 2000.

[11] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, 1974.

[12] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Math. Statistics*, 18:50–60, 1947.

[13] C. K. Park and P. Bas. A model for prediction of transient response to the change of fuel feed rate to a circulating fluidized bed boiler furnace. *Chemical Engineering Science*, 52:3499–3509, 1997.

[14] J. Saastamoinen. Modelling of dynamics of combustion of biomass in fluidized beds. *Thermal Science*, 8:107 – 126, 2004.

[15] I. Zliobaite, J. Bakker, and M. Pechenizkiy. OMFP: An approach for online mass flow prediction in CFB boilers. In *Proc. of Int. Conf. Discovery Science (DS'09)*, volume 5808 of *LNCS*, pages 272–286. Springer, 2009.