Journal Pre-proof

Development of a predictive emissions model using a gradient boosting machine learning method

Minxing Si, Ke Du

PII: S2352-1864(20)31328-6
DOI: https://doi.org/10.1016/j.eti.2020.101028
Reference: ETI 101028

To appear in: *Environmental Technology & Innovation*

Received date : 31 March 2020
Revised date : 30 June 2020
Accepted date : 2 July 2020

Please cite this article as: M. Si and K. Du, Development of a predictive emissions model using a gradient boosting machine learning method. *Environmental Technology & Innovation* (2020), doi: https://doi.org/10.1016/j.eti.2020.101028.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Elsevier B.V. All rights reserved.

Manuscript File

Journal Pre-proof

Development of a Predictive Emissions Model using a Gradient Boosting Machine Learning Method

Minxing Si[a,b] and Ke Du[a*]

[a] Department of Mechanical and Manufacturing Engineering, University of Calgary, 2500 University Drive NW, Calgary AB T2N 1N4 Canada

[b] Tetra Tech Canada Inc., 140 Quarry Park Blvd Suite 110, Calgary, AB T2C 3G3 Canada

*Corresponding author: Ke Du (email: kddu@ucalgary.ca)

Journal Pre-proof

Journal Pre-proof

ABSTRACT: Predictive emissions monitoring systems (PEMSs) are alternatives to continuous emissions monitoring systems (CEMSs) for monitoring air pollutants, such as NO_x. Existing PEMSs and related research have focused on applying artificial neural network (ANN) algorithms. However, ANN-based models are treated as "black boxes". Regulators and decision makers without a statistical background often have difficulty understanding these models, which poses a significant challenge for a broader application of PEMSs. In this study, we proposed a tree-based ensemble method with gradient boosting techniques for PEMS development. Compared to ANNs, tree-based methods are easier to understand and require less effort to preprocess data, fewer hyperparameters for model tuning, and less time for model training. We developed a predictive model using a gradient boosting machine learning library called XGBoost to monitor NO_x emissions from a boiler located in Alberta, Canada. The model uses five process parameters as inputs and the predicted NO_x emissions as output. We trained the model with 202,047 samples using random search methods to determine the best model and tested the model with 50,512 samples. We evaluated the test results against US EPA PEMS standards. The model passed all the statistical tests for precision outlined by US EPA Performance Specification 16. The Pearson correlation r value was 0.98 between the XGBoost-predicted NO_x values and the CEMS-measured NO_x values. The RMSE was 0.14, and the MAE was 0.09. We conclude that XGBoost is a good option for developing PEMSs. Facility operators can use the method provided in this study to develop PEMSs by themselves using the open-source library XGBoost at no cost.

KEYWORDS: Predictive Emissions Monitoring System, PEMS, XGBoost, Gradient Boosting, NO_x Monitoring

Journal Pre-proof

1.0 Introduction

The nitrogen oxides (NO_x) family comprises seven compounds, including nitric oxide (NO), nitrogen dioxide (NO_2), and other oxides of nitrogen derivatives (US EPA, 1999). NO_x, or oxides of nitrogen, commonly refers to NO and NO_2 together (Environment and Climate Change Canada, 2019; US EPA, 2019). NO_x has significant adverse environmental and health effects, contributing to acid rain and smog and forming fine particles (PM) and ozone in the ambient air (tropospheric ozone or ground-level ozone) (Boningari and Smirniotis, 2016; Mauzerall et al., 2005). Industrial facilities with large stationary combustion sources are typically required to be equipped with one or more continuous emissions monitoring systems (CEMSs) to monitor NO_x emissions for compliance with regulatory emission limits (Cozza and Faulkner, 1993; US EPA, 1994; White, 1993). However, CEMSs require high initial capital and operating costs, as well as frequent maintenance and operator training (Roth and Lawrence, 2010).

To address these issues with CEMSs and offer an alternative for monitoring NO_x, a predictive emissions monitoring system (PEMS) was first developed in the 1970s (Hung, 1975). Since 1990, PEMS applications have been widely reported for a range of emission sources in various industries (Chien et al., 2010, 2005; Cooper and Andreasson, 1999; Hung and Langenbacher, 1995; Kamas and Keeler, 1995). The first generation of PEMSs was developed using first principles methods based on energy and mass balances (Chien et al., 2005; Faravelli et al., 2000; Harnevie et al., 1996). The second generation of PEMSs explored the statistical relationship between operating parameters and air emissions (Chien et al., 2010; Lee et al., 2005; Saiepour et al., 2006). When the operating parameters used to build statistical models were selected, thermodynamics was often considered. The second generation was referred to as statistical methods or statistical hybrid methods. The most recent PEMSs, or third generation, were developed using machine learning algorithms called artificial neural networks (ANNs) (Cuccu et al., 2017; Vanderhaegen et al., 2010). An ANN-

based PEMS is more flexible than a PEMS of the first or second generations because it can use operating parameters beyond those related to NO_x formation. Botros *et al.* (2011) developed ANN-based predictive models for NO_x monitoring in gas turbines in natural gas compressor stations. The models used four process parameters and had one hidden layer with two units. The study found that the uncertainty of NO_x prediction was ±2.5% to ±6%. Zain and Kien Kek Chua (2011) developed ANN-based models for CO, O_2, NO_x, and CO_2 monitoring in an incinerator. The ANN used three process parameters as the input layer and had two hidden layers. Each hidden layer had 5 units. The study concluded that the accuracy of the models was 98%. Tan *et al.* (2016) developed NO_x predictive models using a single-hidden-layer feedforward ANN for process optimization and emission reductions in a 700 MW coal-fired boiler. The mean square error (MSE) was 62.1, and the correlation coefficient was 0.98. Si *et al.* (2019) developed ANN-based models for NO_x monitoring in a cogeneration unit. The best model had a mean absolute error (MAE) of 0.60 and Pearson r value of 0.95.

The primary regulatory frameworks for using PEMSs as compliance monitoring and reporting tools were developed by the United States Environmental Protection Agency (US EPA) under Title 40 of the Code of Federal Regulations (CFR), specifically Performance Specification 16 (PS16) and Part 75 (US EPA, 2009). The European Committee for Standardization (CEN) published a technical specification (TS) for PEMS applicability, execution, and quality assurance (CEN/TS 17198:2018) in August 2018 (European Committee for Standardization, 2018). Many countries outside of the US and Europe, such as Gulf countries, use these guidelines for PEMS certification, especially the US EPA standards (ABB S.p.A., 2014). In the US, the first regulatory approval of a PEMS for emissions monitoring and reporting occurred in June 1993 (Kamas and Keeler, 1995). To date, a few hundred PEMSs have been installed worldwide and are used for compliance

Journal Pre-proof

reporting, offline what-if analysis, and analyzer availability enhancement (Eisenmann et al., 2014; Swanson, 2018).

Although PEMSs were developed 50 years ago and have evolved from first principles-based models to machine learning-based models to become more accurate, they have not been widely installed and used because facility operators did not realize significant cost savings when using commercial PEMSs, especially for the initial capital cost. For example, Arkansas Electric Cooperative Corporation installed seven PEMSs for emissions monitoring and regulatory reporting. They estimated that the initial capital costs of installing commercial PEMSs were similar to the costs of installing CEMSs (Bivens, 2019, 2017) because of the software licensing fee and consulting cost for model training and setup.

Si *et al.* were the first to introduce open-source machine learning libraries for PEMS development. Their study built feedforward ANN-based models for NO_x monitoring using Google's TensorFlow and Keras libraries and aimed to provide facility operators with methods to develop predictive emissions models by themselves (Si *et al.*, 2019). However, the models developed in the study inherited some common drawbacks of ANN algorithms. For example, ANN models are treated as "black boxes" because strong statistical knowledge is needed to understand ANN algorithms. Decision makers and regulators are often skeptical about the outputs produced from these black boxes due to the lack of statistical background (Guelman, 2012).

In this study, we developed a predictive model for NO_x emission monitoring and regulatory reporting using a tree-based ensemble machine learning algorithm called XGBoost. XGBoost was implemented as an open-source machine learning library and developed based on gradient boosting techniques introduced by Friedman (2001). Gradient boosting has been successfully applied to tree models (Guelman, 2012; Krauss et al., 2017; Semanjski and Gautama, 2015; Zhang and Haghani, 2015). Gradient tree boosting adds multiple weak trees together and forms a strong learner to optimize predictive performance.

Journal Pre-proof

This method is also known as a gradient boosted regression tree (GBRT) and a gradient boosting machine (GBM). XGBoost was first released in 2014 (Labram, 2019) and has since become popular among machine learning practitioners. In 2015, 17 out of 29 winners in the Kaggle machine learning competitions used the XGBoost algorithm (Nielsen, 2016). XGBoost is easier to interpret than ANN-based algorithms, so decision makers and regulators can better understand how the predictive models work and how outcomes are produced. In addition, XGBoost requires less effort for data preprocessing and has fewer hyperparameters for tuning.

Existing PEMSs and related research have focused on the use of ANN algorithms. To the best of our knowledge, the gradient boosting approach has not been applied to predictive models for emissions monitoring and reporting, even though PEMSs have been researched, developed, and installed for almost five decades. In addition, according to US EPA and EU PEMS standards, PEMSs need to be able to generate data for substitution when one or more process parameters (sensors) fail, such as missing data. Previous studies removed missing data for model development. In this study, we test a method for data substitution to bridge this gap. Using the methods presented in this study, facility operators will have more complete knowledge to develop their emission models by themselves and obtain regulatory approval.

2.0 Material and Method

2.1 Tree-based Methods and Tree-based Ensemble Learning

Tree-based methods build predictive models by repeatedly splitting the predictor space into rectangles by some criteria and then fitting a model for each rectangle. Figure 1 shows that the space for two predictors, fuel gas flow and exhaust gas temperature, is divided into three rectangles, as summarized in Figure 2. R represents the divided areas in Figure 1 called the terminal regions or leaf nodes in the tree model shown in Figure 2. j represents

Journal Pre-proof

the number of leaf nodes. The output w of each leaf node is calculated by averaging the observed values in the training dataset.

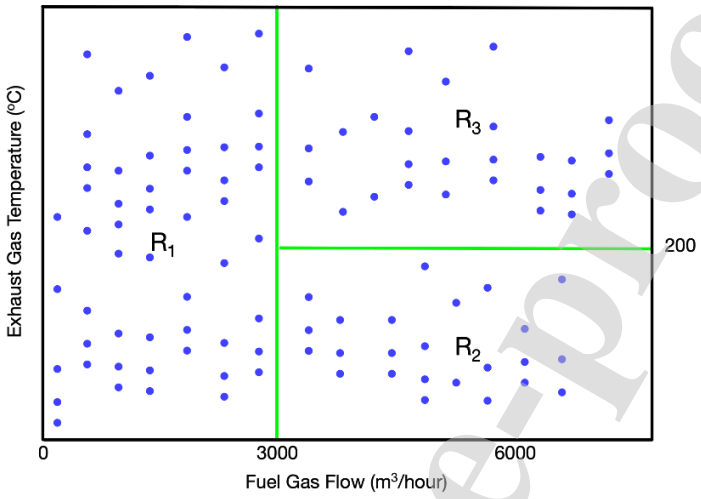


Figure 1: Example of Feature Splitting

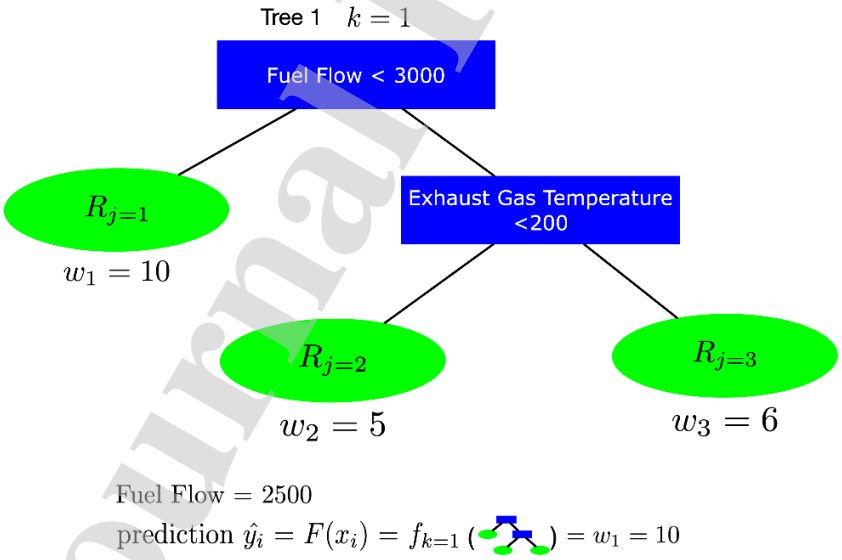


Figure 2: Example of a Tree-Based Predictive Model. R denotes the terminal regions or leaf nodes in the tree model. j is the number of leaf nodes. w is the output of each leaf node. k is the number of trees.

Journal Pre-proof

Ensemble learning builds strong predictive models by combining the strengths of a set of weak models. Tree-based ensemble learning can be written as Equation 1:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F} \qquad \text{Equation 1}$$

where $\mathcal{F}$ is the space of all regression trees and K is the number of trees.

A prediction method using tree-based ensemble learning is presented in Figure 3.

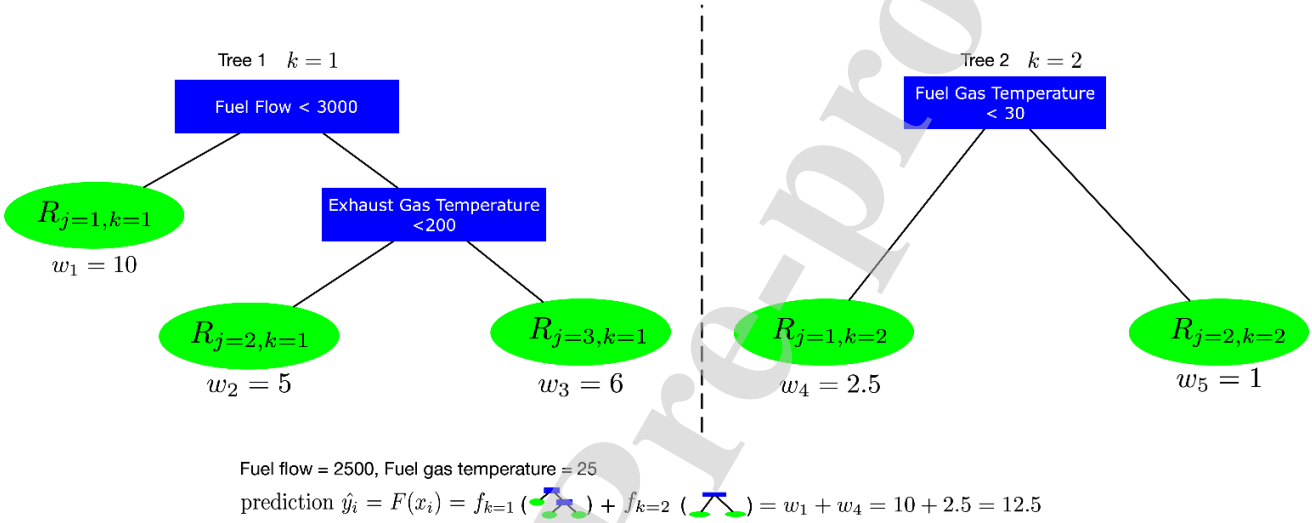


Figure 3: Example of Tree-based Ensemble Learning

2.2 Gradient Boosting and XGBoost

Boosting is an ensemble method that combines multiple weak learners to produce a power learner. A boosting method adds new learners f_x (tree or estimator) sequentially. At iteration k, the new estimator $f_k(x_i)$ tries to correct the previous prediction $\hat{y}_i^{(k-1)}$ or $F^{(k-1)}(x)$ and generates a new prediction $\hat{y}_i^{(k)}$ or $F^{(k)}(x)$, and this process can be written as Equation 2. $F^{(k-1)}(x)$ and $F^{(k)}(x)$ represent the functions that produce the predicted values at iterations $k-1$ and k, respectively. Iteration k also indicates that k trees are ensembled in the model.

$$\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(x_i) \qquad (2)$$

Algorithm 1: Gradient Tree Boosting Algorithm	**Note**
Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$	x_i represents the values of the input

A loss function $L(\boldsymbol{y}, F(x))$

The number of iterations K

The learning rate η

parameters for the i[th] sample.

$\boldsymbol{y}_i$ represents the observed value for the i[th] sample.

n represents the number of samples in the training dataset

$$L = \frac{(Observed\ -\ Predicted)^2}{2}$$

$F(x)$ is the function that produces the predicted values

1 Initiate a model with a constant value

$$F^0(x) = arg\ min_\rho \sum_{i=1}^{n} L(y_i, \rho)$$

ρ is an initial predicted value. ρ could be the mean of the observed values. In XGBoost, the default value is 0.5.

2 For $k\ =\ 1$ to K

k is the k[th] tree

i is the i[th] sample in the training data

a) For $i\ = 1\ to\ n$, compute "pseudo-residuals"

$$r_{ik} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F^{(k-1)}(x)}$$

b) Fit a regression tree to predict the residuals r_{ik}; create a terminal region (leaf) $R_{jk}, j\ = 1, 2, \ldots, J_k$

c) For $j\ = 1, 2, \ldots, J_k$, compute

$$\rho_{jk} = arg\ min_\rho \sum_{x_i \in R_{ij}} L\left(y_i, F^{(k-1)}(x_i)\ +\ \rho\right)$$

j is the j[th] terminal region (leaf)

ρ_k is the output of the k[th] estimator (tree) $f_k(x)$

Journal Pre-proof

	to determine the output values of each leaf d) Update $$F^{(k)}(x) = F^{(k-1)}(x) + \eta \times \sum_{j=1}^{J_k} \rho_{jk} \mathrm{I}(x \in R_{jk})$$	$F^{(k-1)}(x)$ is the previous prediction ρ_{jk} is the output from the tree made in Step c). $\sum_{j=1}^{J_k}$ summation is used when a single sample ends up in multiple leaf nodes
3	Output $F^{(K)}(\mathrm{x})$ $$\hat{y}_i = F^0(x) + \sum_{k=1}^{K} f_k(x_i)$$	

The learning steps by gradient boosting are illustrated in Figure 4.

$\hat{y}_i^0 = F^0(x)$

$\hat{y}_i^1 = F^1(x) = F^0 + f_1($  $)$

$\hat{y}_i^2 = F^2(x) = F^0 + f_1($  $) + f_2($  $)$

•
•
•

$\hat{y}_i^k = F^k(x) = F^0 + f_1($ $) + f_2($ $) + \bullet\bullet\bullet + f_k($ $)$

Figure 4: Illustration of Gradient Boosting

Unlike traditional boosting, XGBoost adds a regularization function to prevent overfitting and optimizes the loss function by a Taylor expansion (Chen and Guestrin, 2016). The objective function J in XGBoost describes the model's performance and can be written as Equation 3:

$$\mathrm{J} = \sum_{i=0}^{n} L(y_i, \hat{y}_i) + \sum_{k=0}^{K} \Omega(f_k) \quad (3)$$

where n is the number of training samples and $\Omega(f_k)$ is a regularization function. $\Omega(f_k)$ is written as Equation 4:

$$\Omega(f_k) = \gamma \mathrm{T} + \frac{1}{2}\lambda \sum_{j=0}^{T} \mathrm{w}_j^2 \quad (4)$$

where T is the number of leaf nodes and γ and λ are hyperparameters in XGBoost that users can define. w is the leaf weight (i.e., a predictive value in a terminal node). The objective function for the m[th] iteration can be written as Equation 5, where m represents elements of K iterations ($m \in \{k_1, k_2, \ldots . K\}$).

$$\mathrm{J}^m = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(m)}\right) + \sum_{k=1}^{m} \Omega\left(f_m(x_i)\right) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{m-1} + f_m(x_i)\right) + \sum_{k=1}^{m} \Omega\left(f_m(x_i)\right) \quad (5)$$

To optimize the objective function, the XGBoost algorithm takes the following steps:

1) Use a second-order Taylor expansion to approximate the derivatives, and write J^m as Equation 6:

$$\mathrm{J}^m \simeq \sum_{i=1}^{n} [L\left(y_i, \hat{y}_i^{(m-1)}\right) + g_m(x_i) f_m(x_i) + \frac{1}{2} h_m(x_i) f_m(x_i)^2] + \sum_{k=1}^{m} \Omega\left(f_m(x_i)\right) \quad (6)$$

where $L\left(y_i, \hat{y}_i^{(m-1)}\right)$ is the loss function for the previous prediction and $g_m(x_i)$ represents the first derivative of the loss function $L\left(y_i, \hat{y}_i^{(m-1)}\right)$, called the gradient. $h_m(x_i)$ is the second derivative of the loss function $L\left(y_i, \hat{y}_i^{(m-1)}\right)$, called the Hessian. $g_m(x_i)$ and $h_m(x_i)$ are written as follows:

$$g_m(x_i) = \frac{dL\left(y_i, \hat{y}_i^{(m-1)}\right)}{d\hat{y}_i^{(m-1)}} \; and \; h_m(x_i) = \frac{d^2 L\left(y_i, \hat{y}_i^{(m-1)}\right)}{d(\hat{y}_i^{(m-1)})^2}$$

2) Remove the constant term $\sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(m-1)}\right)$ because it is not related to the output values $f_m(x_i)$ and has no effect on optimizing the objective function. Equation 6 is then written as Equation 7:

$$\mathrm{J}^m = \sum_{i=1}^{n} [\, g_m(x_i) f_m(x_i) + \frac{1}{2} h_m(x_i) f_m(x_i)^2] + \sum_{k=1}^{m} \Omega\left(f_m(x_i)\right) \quad (7)$$

3) Replace $f_m(x_i)$ with the sum of the tree leaves written as Equation 8. The sum of the tree leaves is the sum of the outputs of the trees:

Journal Pre-proof

$$f_m(x_i) = \sum_{j=1}^{T} w_{jm} \mathrm{I}(x \in R_{jm}) \quad (8)$$

Replace Equation 7 with Equation 8, and J^m is written as Equation 9:

$$\mathrm{J}^m = \sum_{i=1}^{n} [\, g_m(x_i) \sum_{j=1}^{T} w_{jm} + \frac{1}{2} h_m(x_i) \sum_{j=1}^{T} w_{jm}^2] + \gamma \mathrm{T} + \frac{1}{2} \lambda \sum_{j=1}^{T} w_{jm}^2 \quad (9)$$

The sums of $g_m(x_i)$ and $h_m(x_i)$ can be simplified as in Equation 10:

$$G_{jm} = \sum\nolimits_{i \in I_{jm}} g_m(x_i) \,, H_{jm} = \sum\nolimits_{i \in I_{jm}} h_m(x_i) \quad (10)$$

where I_{jm} denotes the set of training samples (instances, x_i) in the region (leaf) R_{jm}.

4) Replace $g_m(x_i)$ and $h_m(x_i)$ in Equation 9 with Equation 10. J^m is written as Equation 11.

$$\mathrm{J}^m = \sum_{j=1}^{T} [\, G_{jm} w_{jm} + \frac{1}{2} (H_{jm} + \lambda) w_{jm}^2 \,] + \gamma \mathrm{T} \quad (11)$$

5) Minimize the objective J^m in Equation 11 by the weight w_{jm} for each leaf using the derivative of J^m with respect to w_{jm}, as written in Equation 12.

$$\frac{\partial J^m}{\partial w_{jm}} = G_{jm} + (H_{jm} + \lambda) w_{jm} = 0 \quad (12)$$

6) Calculate the best weight w_{jm} using Equation 13.

$$w_{jm} = -\frac{G_{jm}}{H_{jm} + \lambda} \quad (13)$$

For regression with MSE used for the loss function, the gradient G_{jm} is the sum of negative residuals, or $-\sum_{i=1}^{n}(y_i - \hat{y}_i)$, and the Hessian H_{jm} is the number of residuals n.

Therefore, Equation 13 is used to calculate the output value for a leaf.

7) Replace w_{jm} in Equation 11 with Equation 13; then, the objective J^m in Equation 11 for

the best tree structure can be written as Equation 14.

Journal Pre-proof

Journal Pre-proof

$$J^m = -\frac{1}{2}\sum_{j=1}^{T}\left[\frac{G_{jm}^2}{H_{jm}+\lambda}\right] + \gamma T \quad (14)$$

2.3 Data

The predictive model was developed to monitor NO_x emissions from a boiler. The NO_x emissions were continuously monitored by a CEMS unit. The CEMS unit was an in situ monitoring system with path in situ analyzers manufactured by SICK. The NO_x concentrations were measured by a differential optical absorption spectroscopy method. The NO_x concentration range of the analyzer in the CEMS unit was 0–100 ppm by volume on a wet basis with 0-80 ppm for NO and 0-20 ppm for NO_2. The CEMS unit measured NO_x in ppm, exhaust temperature, and exhaust flow and then converted NO_x ppm into kg/h. The NO_x mass flow, exhaust temperature, and exhaust flow data were reported to Alberta's provincial regulatory agency under the facility's operational permit. The facility CEMS report included two parts: station status (SS), which records the boiler downtime, and record details (RD), which records the detailed NO_x emissions data, including substitute data when the CEMS unit was offline or in maintenance.

We selected five process parameters as inputs for machine learning (Table 1). NO_x in kg/h was the output of the model. The input parameters are also called PEMS sensors. We retrieved process data and NO_x emissions data at one-minute intervals from 00:00 on January 1, 2019, to 23:59 on June 30, 2019, from a data storage system called the process historian database (PHD). The data of the five process parameters and NO_x emissions formed a structured dataset with 260,641 rows and 6 columns. Each row is called one sample or example. The five sensors were integrated into the facility's distributed control system and had routine preventative maintenance schedules for calibration and inspection to ensure that the instrument worked properly. The metered data obtained by the five sensors were automatically transferred to the PHD.

Table 1: Model Inputs

Sensor ID	Input (process) parameters	Unit
S0	fuel gas temperature	°C
S1	boiler feed water temperature	°C
S2	fuel gas flow	m^3/h
S3	combustion air flow	10^3 m^3/day
S4	Exhaust gas temperature	°C

The following steps were taken to process the raw data:

- Step 1: Remove samples when the boiler was offline based on the SS records because the facility does not report NO_x emissions in these periods (7,326 samples removed).
- Step 2: Remove samples that have substitute NO_x emissions data (240 samples removed).
- Step 3: Remove samples for which the fuel gas flow was 0 (515 samples removed). There was a delay between the time that the fuel gas supply was stopped and the time that the exhaust exited the stack. The CEMS recorded small NO_x readings for approximately 10 minutes after the fuel gas readings were 0. The NO_x values ranged from 0 to 5.3 kg/h and contributed to 0.007% of NO_x emissions.
- Step 4: Shuffle the remaining dataset (252,559 samples), and randomly split 80% of the samples into the training dataset and 20% into the test dataset. The training dataset had 202,047 samples and was used for model building and training. The test dataset had 50,512 samples and was used to test the model predictive power for an unbiased evaluation.

2.4 Machine Learning and Model Evaluation

The paired process data and CEMS-measured NO_x emissions data were used to train the PEMS model using the XGBoost library (version 0.90) in a Python (version 3.7.4) environment. The CEMS was the reference method in this study. Ten-fold cross-validation

was used to select the best model during training. The following hyperparameters were tuned by a random search technique:

- Number of trees to fit (n_estimator)
- Maximum depth of a tree (max_depth)
- Step-size shrinkage used in update (learning_rate, η)
- Subsample ratio of columns when constructing each tree (colsample_bytree)
- Minimum loss reduction required to make a further partition on a leaf node of the tree (gamma, γ)
- L2 regularization on weights (reg_lambda, λ)
- Minimum sum of instance weight needed in a child (min_child_weight)

A detailed explanation of each hyperparameter is provided in the XGBoost documentation (XGBoost developers, 2019). We used a random search method for model tuning using the scikit-learn (0.21.3) library. "Random search" means that combinations of the parameters are randomly selected to find the best model structure.

We compared the NO_x outputs from the model using the test dataset with the corresponding CEMS-measured NO_x data. The results were evaluated against the criteria outlined in the US EPA PS16 standard for PEMS precision. The criteria are presented in Table 2.

Table 2: Precision Requirements for Predictive Models

Statistical test required by PS16	**Criteria**
Bias test	$d_{avg} \leq \lvert cc \rvert$ The mean difference d_{avg} between Reference Method (RM) values and predicted values is less than or equal to the absolute value of the confidence coefficient (CC) at a 97.5% one-sided confidence

Journal Pre-proof

	interval. Otherwise, a bias factor needs to be applied to the predicted values.
Pearson correlation coefficient	$r \geq 0.8$ The correlation between CEMS values and predicted values must be 0.8 or greater.
F-test	$F_{value} \leq F_{critical}$ The variance ratio of predicted values and CEMS measured values must be less than or equal to the $F_{critical}$ value.

The model was also assessed by root mean square error (RMSE), MAE and MSE using the test dataset.

2.5 Model Sensitivity Test

A model sensitivity test was used to evaluate changes in model performance using substitute data when one or more sensors failed. As required by the EU PEMS TS and the US EPA's PEMS standards, a PEMS needs to have a sensor validation system to identify sensor failure hourly. The sensor validation system is responsible for generating substitute data, informing operators when sensors (process instruments, such as a fuel gas flow meter) need repair and indicating that the PEMS is out of control (US EPA, 2006).

We followed the procedures for model sensitivity testing provided by the EU PEMS TS and US EPA CFR 40 Part 75 Subpart E. The procedures are summarized as follows:

I. Select a set of reference sensor values and NO_x emissions values.

II. Artificially fail one sensor, and then run the predictive model using substitute data. We assessed the effect on the model's accuracy by calculating the hourly percentage difference between the reference NO_x values in Step I and the predicted values using

Journal Pre-proof

substitute data in Step II. Repeat this procedure for all the sensors used by the model individually.

III. Flag the outputs as invalid if the hourly percentage difference is greater than 10%.

IV. Perform a two-sensor failure test by repeating Step I, and then identify invalid data by conducting Step III. We only performed the test for up to two sensor failures because the US EPA requires a predictive model to have at least three input sensors, and this study used five inputs for machine learning.

We used directly measured sensor data and NO_x values from 00:00 on March 30, 2019, to 23:59 on April 5, 2019, for the set of reference sensor values and NO_x emission values as required in Step I. The boiler was in a normal operational condition in this period. The data substitution method used for the sensitivity test is illustrated in Table 3. Data were substituted by interpolating the time distance. We tested sensor failure for one hour (1H), one day (1D), two days (2D), three days (3D), four days (4D), and five days (5D). According to Alberta's CEMS code, a CEMS can only allow a sensor to be out of control or offline for up to five days.

Table 3: Example of Data Substitution by Interpolating the Time Distance

Timestamp	Fuel Gas Flow (10^3 m^3/h)	Substitute data
March 31, 2019 00:00	5	
March 31, 2019 00:01	Invalid data (sensor failure)	6
March 31, 2019 00:02	Invalid data (sensor failure)	7
March 31, 2019 00:03	Invalid data (sensor failure)	8
March 31, 2019 00:04	9	

3.0 Results and Discussion

3.1 Hyperparameter Tuning

The results of the random search for the hyperparameters are presented in Table 4. The gradient boosting model comprised 38 trees $f_{K=38}(x)$, and the learning rate η was 0.405. The

Journal Pre-proof

two user-defined values γ and λ in the regularization function were 0.12 and 69.05, respectively.

Table 4: Random Search Results for Hyperparameters

XGBoost Hyperparameters	Values
Number of trees to fit (n_estimator)	38
Maximum depth of a tree (max_depth)	32
Step-size shrinkage used in update (learning_rate, η)	0.41
Subsample ratio of columns when constructing each tree (colsample_bytree)	0.96
Minimum loss reduction required to make a further partition on a leaf node of the tree (gamma, γ)	0.12
L2 regularization on weights (reg_lambda, λ)	69.05
Minimum sum of instance weight needed in a child (min_child_weight)	61.63

3.2 Evaluation

The results of the statistical tests are presented in Table 5. The model passed all three EPA requirements for precision. The RMSE was 0.14, and the MAE was 0.09.

Table 5: Model Evaluation Results

Statistical tests	Results
Bias test	$d_{avg} = -0.0011$, $\lvert cc \rvert = 0.0012$ $d_{avg} < \lvert cc \rvert$; the model passed the EPA requirement. Bias adjustment is not needed.
Pearson r value	$r = 0.98$ $r > 0.8$; the model passed the EPA requirement.
F-test	$F_{value} = 0.96, F_{critical} = 1.01$ $F_{value} < F_{critical}$; the model passed the EPA requirement.
RMSE	0.14

Journal Pre-proof

MAE	0.09
MSE	0.02

Note: The statistical results were based on 50,512 test samples.

Figure 5 shows a comparison between the values measured by the reference method (CEMS) and the values predicted by the XGBoost model. Most of the data fall along the 1:1 ratio line, which is the yellow dashed line.

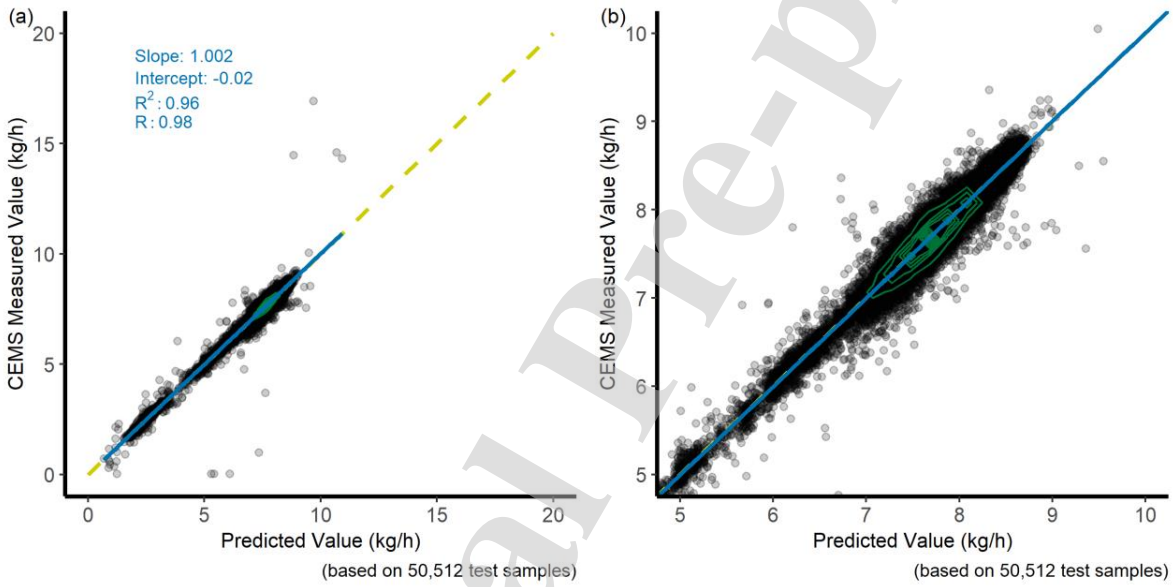


Figure 5: CEMS Values vs. XGBoost Predicted Values. The yellow dashed line is a 1:1 line. The solid blue line is a regression line. (a) Plot is in full scale, (b) plot is a zoom-in plot of plot (a) for the range of 5 kg/h to 10 kg/h. The green circles represent data density.

Figure 6 shows the residuals of the XGBoost model. A total of 50,481 out of 50,512 (99.94%) samples had residuals from -1 kg/h to 1 kg/h. Nine out of 50,512 samples had residuals over ±3 kg/h and are not shown in Figure 6. The largest absolute residual was 7.26, and the smallest absolute residual was 2.7×10^{-6}.

Journal Pre-proof

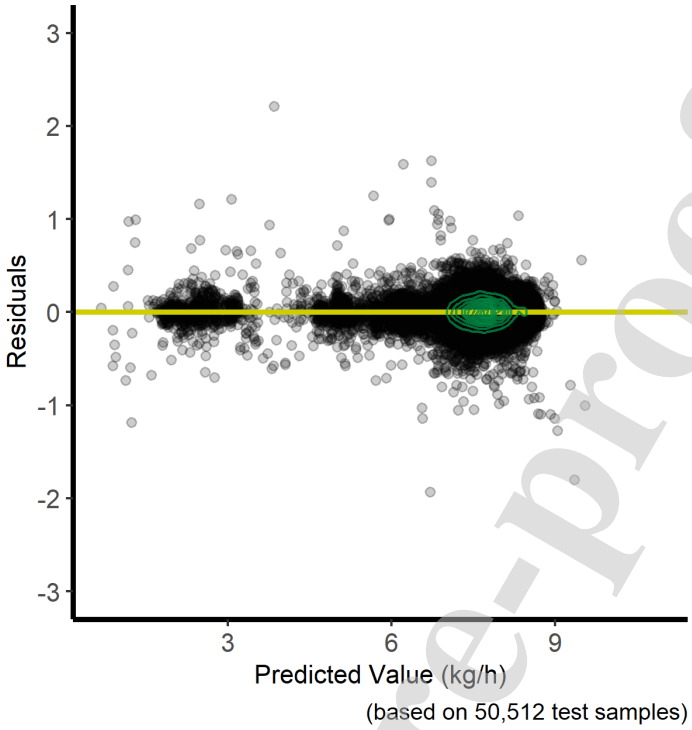


Figure 6: Residuals vs. XGBoost Predicted Values. The green circles represent data density. Nine samples are out of the residual range [-3,3] and are not shown in the plot.

XGBoost also ranks the importance of input parameters using the F score. The F score is calculated based on the frequency of one input parameter used in splitting the data across all the trees. In this model, fuel temperature and fuel flow were the most important inputs for the predictive model (Fig. 7).

Journal Pre-proof

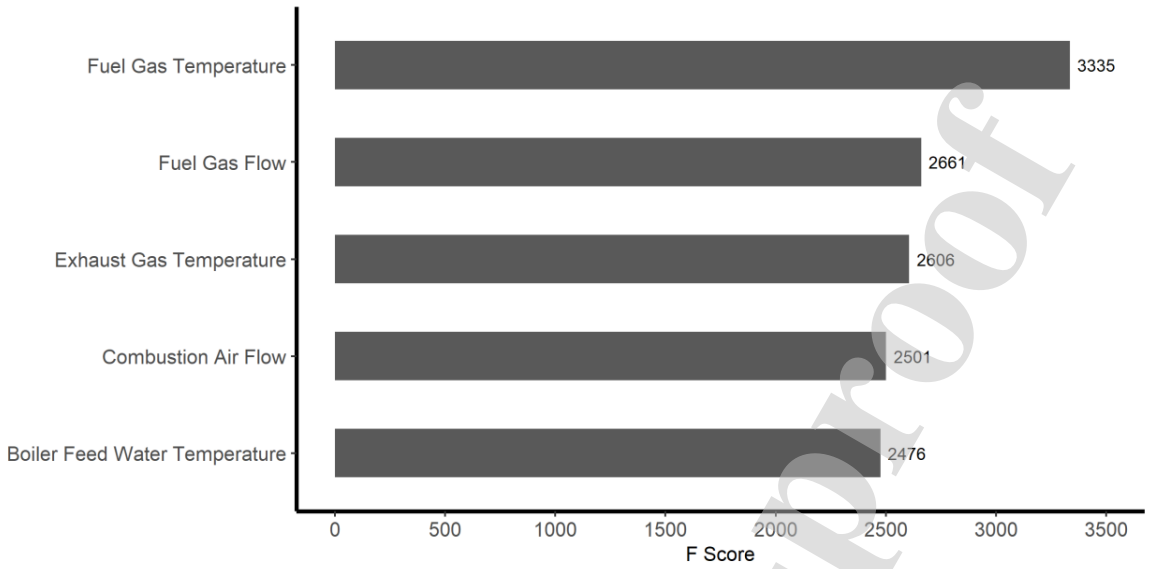


Figure 7: Importance of the Input Parameters

3.3 Model Sensitivity

Figure 8 shows the absolute differences between the CEMS measured values and the predicted values with substitute data when one sensor failed. The largest differences were found when the boiler feed water temperature sensor failed for five days (3.28%) and when the exhaust gas temperature sensor failed for three days (3.01%). The differences were within the maximum allowable difference of 10%. The impact on the model performance of using substitute data for fuel gas temperature, fuel gas flow, and combustion air flow was less than 1% when one of the sensors failed for up to five days.

The absolute differences between the CEMS measured values and the predicted values with substitute data for two-sensor failure are presented in Figure 9. The largest difference was found when fuel gas temperature and boiler feed water temperature failed together for five days (5.14%), but the difference was still within the 10% regulatory limit.

Journal Pre-proof

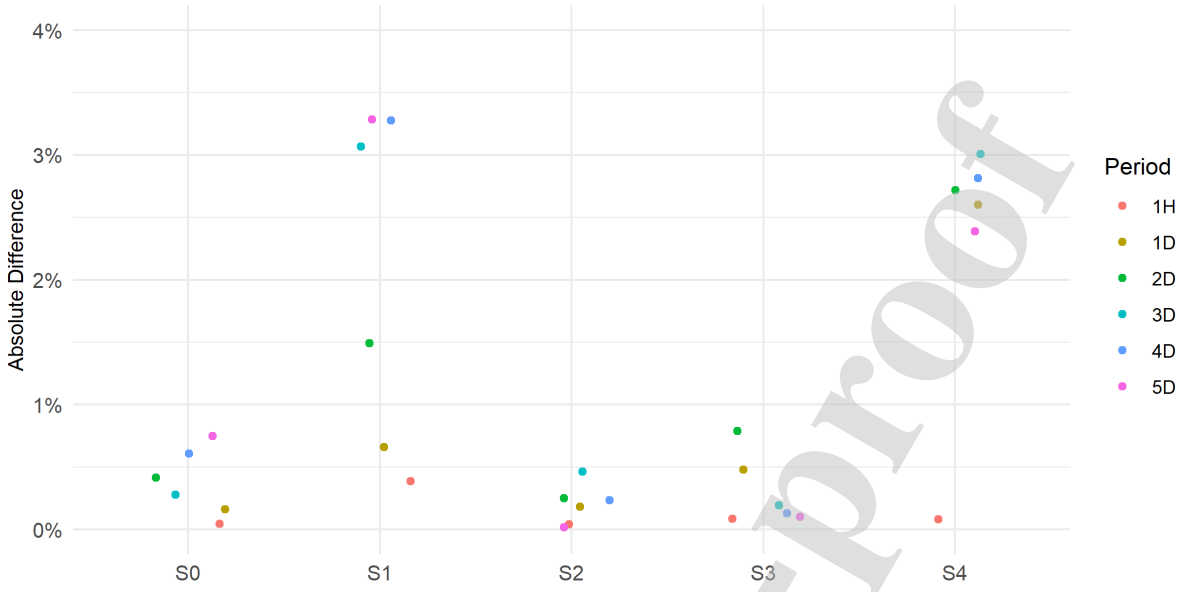


Figure 8: Sensitivity Test for One-Sensor Failure. The X-axis is the sensor ID. See Table 1 for details.

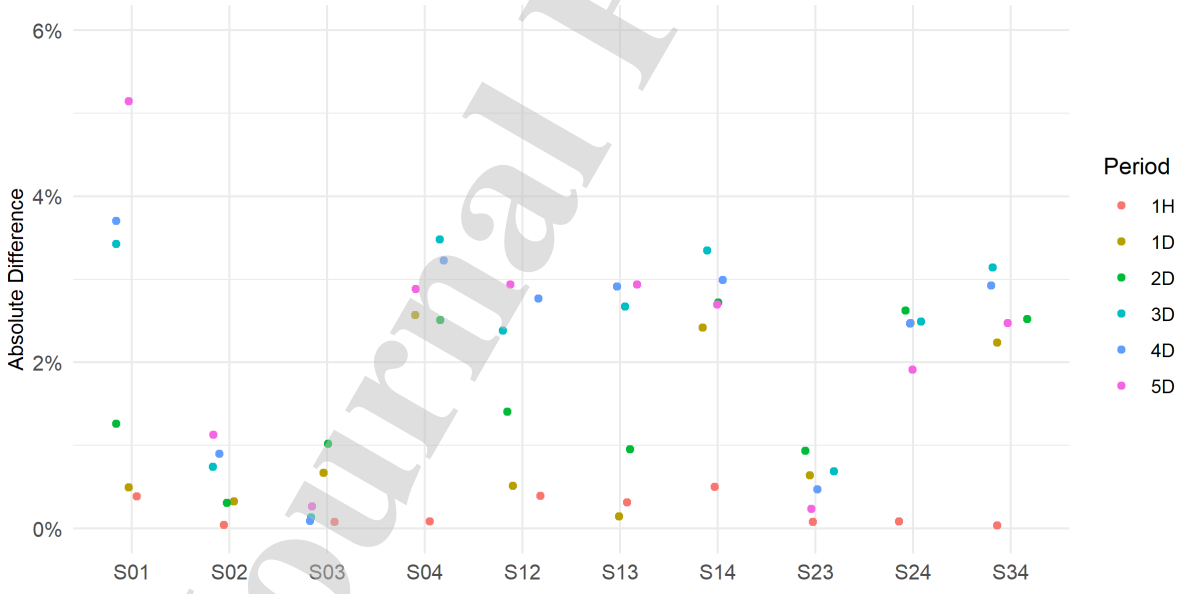


Figure 9: Sensitivity Test for Two-Sensor Failure. The x-axis is a combination of sensor IDs. For example, S01 represents the failures of sensor 0, fuel gas temperature, and sensor 1, boiler feed water temperature.

Journal Pre-proof

3.4 Other Regulatory Requirements for PEMSs

In addition to the precision requirements that are outlined in Table 2, the US EPA (2018) also has the following requirements to validate PEMSs for emissions monitoring (Table 6) in CFR 40 PS16 and Part 75.

Table 6: Other Regulatory Requirements for PEMSs

Requirement	Criteria
Reliability	PEMS availability should be greater than or equal to 95%.
Quality Assurance	Input sensors must operate within the permitted ranges, such as model training ranges and manufacturing ranges.
	Daily check to ensure model is not modified
	Input sensors must be maintained in accordance with the manufacturer's recommendations.
	A PEMS should be equipped with an alarm system. The alarm system will inform facility operators when the PEMS is out of control, such as sensors out of permitted ranges and sensor failures.
	Routine relative accuracy test audits must be conducted to ensure a constant performance of a PEMS after initial certification.

3.5 Comparison with an Artificial Neural Network

We also developed NO_x predictive models using a feedforward neural network algorithm for comparison with the XGBoost model. The ANN model was trained using random search methods for the best model structure. The ANN model was trained and tested with the same

Journal Pre-proof

training and test datasets as the XGBoost model. The best ANN model had the following hyperparameters:

- Number of hidden layers: 3
- Units in each hidden layer: 128
- Learning rate: 0.0005
- L2 regularization: 0.01
- Optimization: Nadam

The ANN model inputs and output (input and output layers) were the same as those in XGBoost.

The Pearson r value for the XGBoost model was 0.98, compared to 0.92 for the ANN model. The RMSE for the XGBoost model was 56% less than that for the ANN model. The MAE for the XGBoost model was 61% less than that for the ANN model (Table 7). The higher r value and lower RMSE indicated that the XGBoost model was a better model than the ANN model.

Table 7: Comparison between XGBoost and the ANN using the Test Dataset

Statistical tests	XGBoost	ANN
Pearson r	0.98	0.92
RMSE	0.14	0.32
MAE	0.09	0.23

Note: The test dataset contains 50,512 samples at 1-minute intervals.

3.6 Model Performance under Non-normal Operating Conditions

Under non-normal operating conditions (N-NOCs), such as equipment startup and shutdown, the RMSE increased from 0.13 to 0.72 for XGBoost (Fig. 10) compared to that of the normal operating condition (NOC). The Pearson r value between the CEMS measured values and the model predicted values decreased from 0.99 under the NOC to 0.91 under N-NOCs for the XGBoost model. The increase in RMSE and the decrease in the r value indicated that the XGBoost model did not perform as well under N-NOCs as under the NOC.

Journal Pre-proof

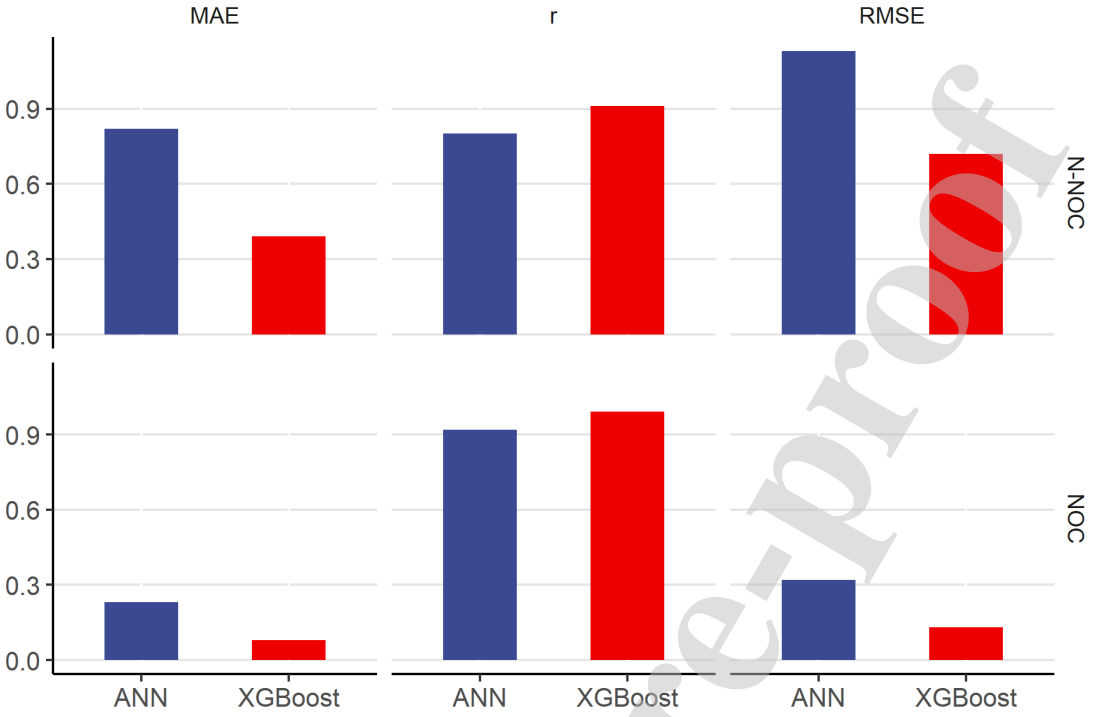


Figure 10: Non-Normal Operating Condition (N-NOC) vs. Normal Operating Condition (NOC). The NOC comprised 252,180 samples at 1-minute intervals. The N-NOC comprised 379 samples at 1-minute intervals.

Using 1-minute high-resolution data, the ANN model generated higher predicted values when the NO_x mass flow rate was less than 2 kg/h and generated lower predicted values when the NO_x mass flow rate was greater than 8 kg/h. In contrast, the XGBoost model performed better for flow rates less than 2 kg/h than the ANN model. However, the XGBoost model still produced values less than 12 kg/h when the CEMS values were greater than 12 kg/h (Fig. 11).

Journal Pre-proof

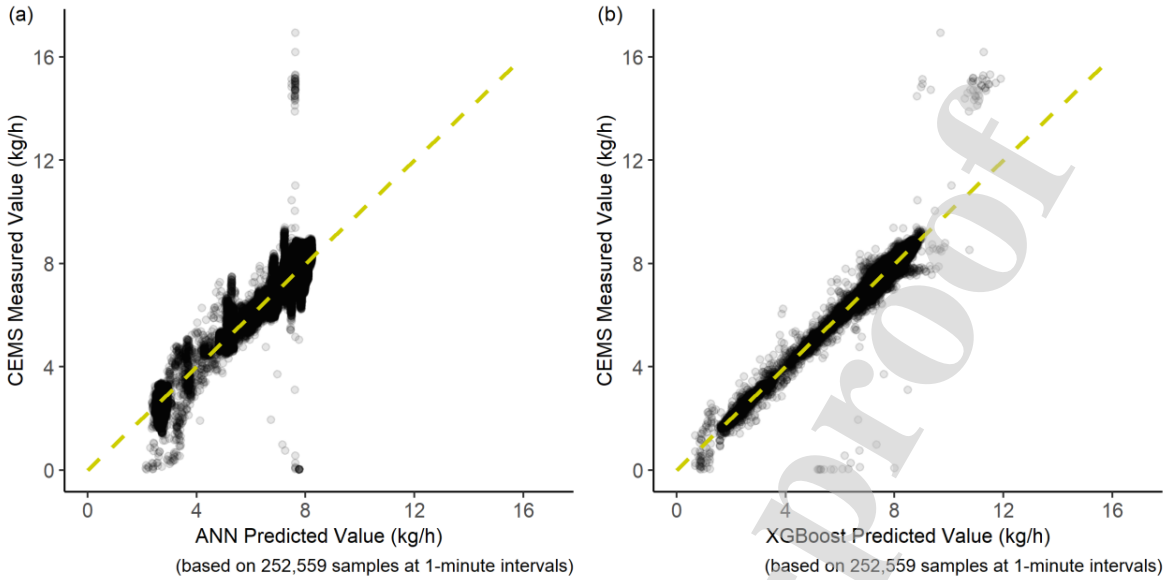


Figure 11: CEMS Values vs. Predicted Values at 1-minute intervals. The yellow dashed line is the 1:1 ratio line.

The facility reports its NOx emissions on an hourly basis, and the emission limit for the facility under its operation permit was set in kg/h. We averaged the 1-minute data to hourly data and compared the model performance. On an hourly basis, no emission rates were greater than 12 kg/h. Using the hourly data, the Pearson r was 0.93 for the ANN and 0.99 for XGBoost. All data provided by XGBoost fall close to the 1:1 ratio line (Fig. 12). The XGBoost model showed a higher accuracy than the ANN model when reporting emissions on an hourly basis.

Journal Pre-proof

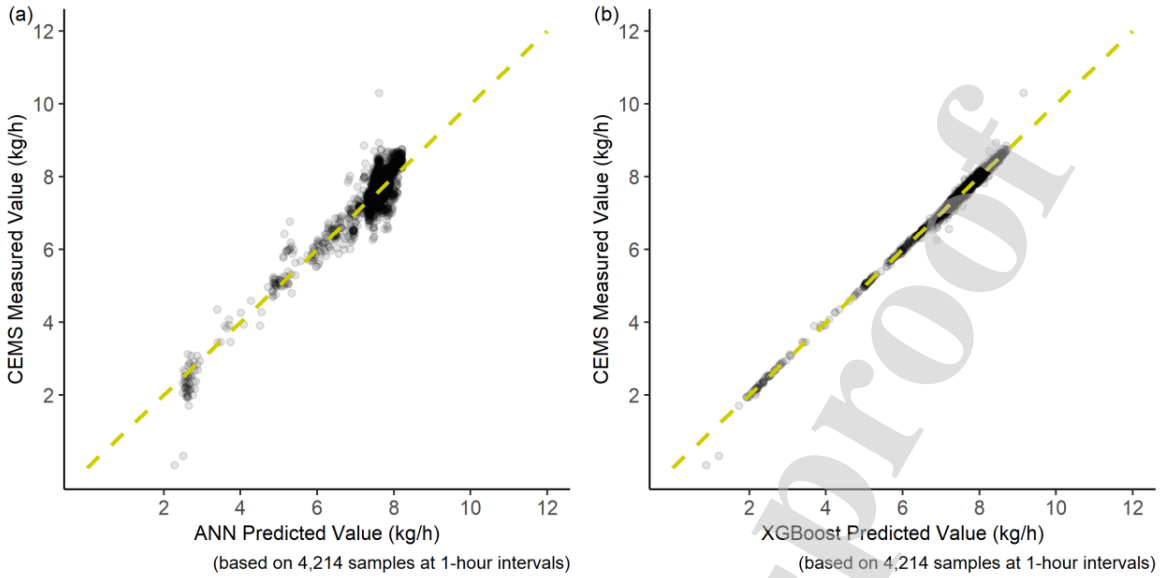


Figure 12: CEMS Values vs. Predicted Values at 1-hour intervals. The yellow dashed line is the 1:1 ratio line.

When upset occurred, the XGBoost model exhibited better performance than the ANN model. However, the XGBoost model did not produce an accurate predictive value when the emission rate reached its peak. For example, at 14:14 on May 1, 2019, a process upset occurred, and the emission rate suddenly dropped from 7.5 kg/h to close to 0 and then increased to a peak of 16.9 kg/h in 42 minutes. After the peak, it took another 18 minutes for the emission rate to return to normal 7.8 kg/h.

The upset event started at 14:14 on May 1, 2019, and ended at 15:14 on May 1, 2019. When reporting a 1-hour average emission rate, the absolute error of XGBoost for the peak emission rate at 14:00 was 1.14 kg/h (Fig. 14).

Journal Pre-proof

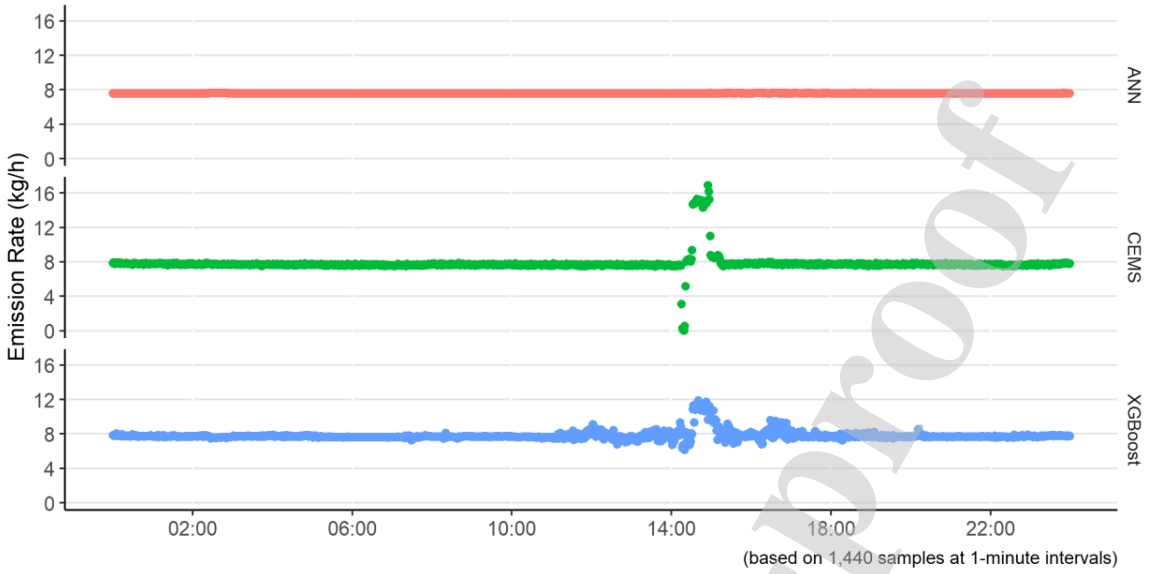
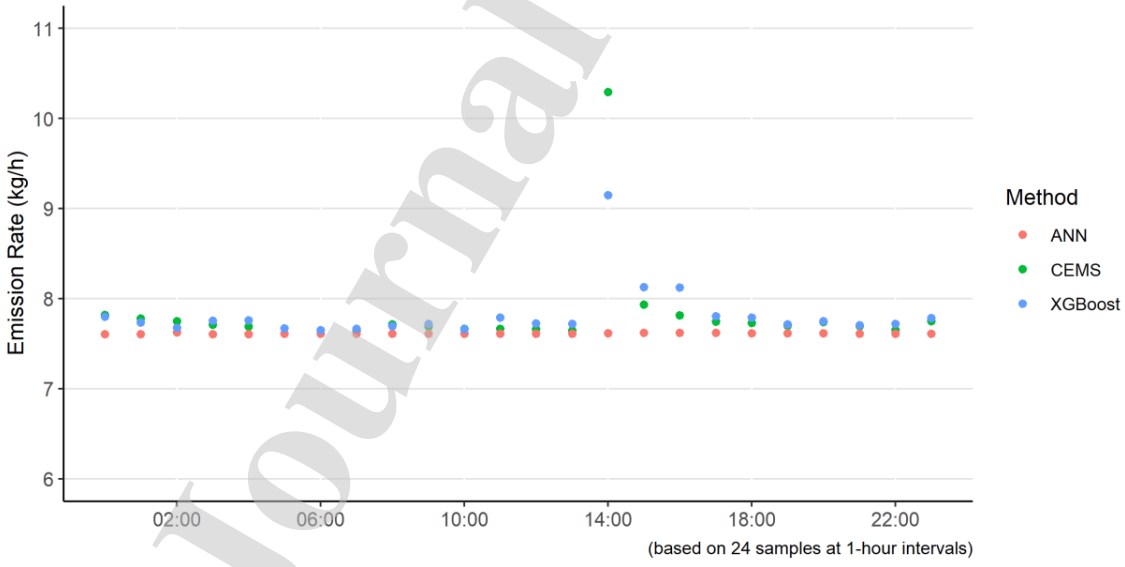


Figure 13: Model evaluation of a process upset event using data at 1-minute intervals. The data period is from 00:00 on May 1, 2019, to 23:59 on May 1, 2019.



Journal Pre-proof

Figure 14: Model evaluation of a process upset event using data at 1-hour intervals. The data period is from 00:00 on May 1, 2019, to 23:00 on May 1, 2019.

3.7 Limitations and Future Study

Although a predictive model is the core component of a PEMS, facility operators need to consider the following when building a PEMS on their own:

1) Deploying predictive models to existing process control networks. Process control devices, such as distributed control systems, on process networks may not have sufficient computational power to run complicated predictive models.
2) Integrating predictive models into existing data acquisition systems and reporting systems.
3) Multiple models may be needed to replace one CEMS unit. For example, facilities in Alberta, Canada, are required to report exhaust flow and temperature, in addition to mass emission rates.

Predictive modeling, including ANN and XGBoost algorithms, has shown great performance for emission monitoring. Computer-based emission monitoring methods can be installed on small stationary combustion sources that do not have regulatory requirements for continuous monitoring. A PEMS provides a more accurate emissions reporting method for small sources than engineering estimation methods with generic emissions factors that are normally used for regulatory reporting. Predictive modeling also offers the potential to construct a high-resolution network for continuously monitoring point source emissions without high capital cost. However, further research is needed because of the following limitations:

- Predictive models tend to be equipment-specific. A model trained for one piece of equipment may not generate the same performance when applied to another piece

Journal Pre-proof

of equipment for air emissions monitoring. Model generalization is needed for wider installation and application.

- The performance of predictive models for low and high emission rates (or low and high emission concentrations) is not as good as that for normal emission rates. Accurately predicting high emission rates is critical because facility operators must demonstrate through monitoring methods that their operations meet the maximum emission limits set by the regulator. In this study, the XGBoost model generated lower predictive values than the CEMS measured values for high emission rates.

3.8 Conclusions

Open-source libraries can play a critical role in the wider installation of PEMSs because commercial PEMSs reportedly have initial capital costs similar to CEMSs. Compared with ANN-based predictive models, tree-based machine learning methods, such as XGBoost, are easier to understand, require less effort for data preprocessing, have fewer hyperparameters for model tuning, and need less time for model training.

In this study, we demonstrated that the XGBoost machine learning algorithm can be used to build predictive models for NO_x emissions monitoring. The NO_x emission values predicted by the best model structure using random search techniques meet all the statistical requirements outlined by US EPA PS16. The Pearson correlation r value between the XGBoost-predicted NO_x values and the CEMS-measured NO_x values was 0.98 at 1-minute intervals. The RMSE and MAE using 1-minute interval data in the test dataset were 0.14 and 0.09, respectively.

In addition, we proposed a data imputation method for sensor failure and tested model sensitivity using this method. Although there are many more complex data imputation methods, such as K-nearest neighbors, K-means, and multiple imputations by chained equations (Schmitt et al., 2015), we demonstrated a simple imputation by interpolating that the time distance is enough to meet regulatory requirements for

Journal Pre-proof

predictive emissions modeling. The model inputs are univariate time series data, and linear interpolation is effective for missing data imputation because the precision of the XGBoost model used in this study changed only up to 5.14% using substitute data when two sensors failed for up to five days.

Further research is still needed to improve model precision for non-normal operating conditions, especially when emission rates (concentrations) are high, because the precision determines if facilities meet the emission limits required by regulations.

Acknowledgments

The authors thank the Environment and Water group and Jessica Coles from Tetra Tech Canada Inc. for editing the manuscript.

This study was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada (fund number CRDPJ535813-18) and by Mitacs through the Mitacs Accelerate program (fund number IT18400).

Author Contributions

MS: Formal analysis; Methodology; Writing - original draft. DK: Project administration; Funding acquisition; Writing - original draft.

Declaration of Competing Interest

The authors declare they have no actual or potential competing interests.

Reference

ABB S.p.A., 2014. Predictive Emission Monitoring Systems The new approach for monitoring emissions from industry [WWW Document]. URL https://library.e.abb.com/public/4128e88396a14d83b8116a2a71b5b88f/PB_PEMS-EN.pdf (accessed 8.22.18).

Bivens, T., 2019. Arkansas Electric Cooperative Corporation. Little Rock, Arkansas. Personal communication

Journal Pre-proof

Journal Pre-proof

Bivens, T., 2017. CEMS Alternatives – A Different Way. https://www.jdsupra.com/legalnews/cems-alternatives-a-different-way-tim-42097/(accessed 8.23.18).

Boningari, T., Smirniotis, P.G., 2016. Impact of nitrogen oxides on the environment and human health: Mn-based materials for the NO x abatement. Curr. Opin. Chem. Eng. 13, 133–141. https://doi.org/10.1016/j.coche.2016.09.004

Botros, K.K., Williams-Gossen, C., Makwana, S., Siarkowski, L., 2011. Predictive Emission Monitoring (PEM) Systems Development and Implementation.

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. Presented at the the 22nd ACM SIGKDD International Conference, ACM Press, San Francisco, California, USA, pp. 785–794. https://doi.org/10.1145/2939672.2939785

Chien, T.W., Chu, H., Hsu, W.C., Tu, Y.Y., Tsai, H.S., Chen, K.Y., 2005. A performance study of PEMS applied to the Hsinta power station of Taipower. Atmos. Environ. 39, 223–230. https://doi.org/10.1016/j.atmosenv.2004.09.062

Chien, T.W., Hsueh, H.T., Chu, H., Hsu, W.-C., Tu, Y.-Y., Tsai, H.-S., Chen, K.-Y., 2010. A Feasibility Study of a Predictive Emissions Monitoring System Applied to Taipower's Nanpu and Hsinta Power Plants. J. Air Waste Manag. Assoc. 60, 907–913. https://doi.org/10.3155/1047-3289.60.8.907

Cooper, D.A., Andreasson, K., 1999. Predictive NOx emission monitoring on board a passenger ferry. Atmos. Environ. 33, 4637–4650. https://doi.org/10.1016/S1352-2310(99)00239-3

Cozza, A., Faulkner, K.F., 1993. Acid rain program offers free-market incentives, portends future regulation. Hazmat World U. S. 6:5.

Cuccu, G., Danafar, S., Cudre-Mauroux, P., Gassner, M., Bernero, S., Kryszczuk, K., 2017. A data-driven approach to predict NOx-emissions of gas turbines, in: 2017 IEEE International Conference on Big Data (Big Data). Presented at the 2017 IEEE International Conference on Big Data (Big Data), IEEE, Boston, MA, pp. 1283–1288. https://doi.org/10.1109/BigData.2017.8258056

Eisenmann, T., Bianchin, D.R., Triebel, D., 2014. Predictive Emission Monitoring (PEM): Suitability and Application in View of U.S. EPA and European Regulatory Frameworks. Presented at the The 11th International Conference & Exhibition on Emissions Monitoring (CEM) 2014, Istanbul,Turkey, p. 15.

Environment and Climate Change Canada, 2019. Canada's Air Pollutant Emissions Inventory Report: annex 1 [WWW Document]. aem. URL https://www.canada.ca/en/environment-climate-change/services/air-pollution/publications/emissions-inventory-report-2019/annex-1.html (accessed 1.4.20).

European Committe for Standardization, 2018. Stationary source emissions - Predictive Emission Monitoring Systems (PEMS) - Applicability, execution and quality assurance.

Faravelli, T., Bua, L., Frassoldati, A., Antifora, A., Tognotti, L., Ranzi, E., 2000. A new procedure for predicting NOx emissions from furnaces. Comput. Aided Chem. Eng., European Symposium on Computer Aided Process Engineering-10 8, 859–864. https://doi.org/10.1016/S1570-7946(00)80145-5

Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. Ann. Stat. 29, 1189–1232.

Guelman, L., 2012. Gradient boosting trees for auto insurance loss cost modeling and prediction. Expert Syst. Appl. 39, 3659–3667. https://doi.org/10.1016/j.eswa.2011.09.058

Journal Pre-proof

Harnevie, H., Sarkoezi, L., Trenkle, S., 1996. Predictive emission monitoring system (PEMS) for emission control in biomass fired plants (No. SVF--575). Stiftelsen foer Vaermeteknisk Forskning.

Hung, W.S.Y., 1975. An Experimentally Verified NOx Emission Model for Gas Turbine Combustors V01BT02A009. https://doi.org/10.1115/75-GT-71

Hung, W.S.Y., Langenbacher, F., 1995. PEMS: Monitoring NOx Emissions From Gas Turbines, in: Volume 5: Manufacturing Materials and Metallurgy; Ceramics; Structures and Dynamics; Controls, Diagnostics and Instrumentation; Education; IGTI Scholar Award. Presented at the ASME 1995 International Gas Turbine and Aeroengine Congress and Exposition, ASME, Houston, Texas, USA, p. V005T15A016. https://doi.org/10.1115/95-GT-415

Kamas, J., Keeler, J., 1995. Predictive emissions monitoring systems: a low-cost alternative for emissions monitoring [in cement industry], in: 1995 IEEE Cement Industry Technical Conference. 37th Conference Record. Presented at the 1995 IEEE Cement Industry Technical Conference. 37th Conference Record, IEEE, San Juan, Puerto Rico, pp. 497–509. https://doi.org/10.1109/CITCON.1995.514350

Krauss, C., Do, X.A., Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. Eur. J. Oper. Res. 259, 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

Labram, A., 2019. Article: Fitting data with XGBoost | Institute and Faculty of Actuaries [WWW Document]. URL https://www.actuaries.org.uk/news-and-insights/news/article-fitting-data-xgboost (accessed 1.23.20).

Lee, Y.-H., Kim, M., Han, C., 2005. Application of Multivariate Statistical Models to Prediction of NOx Emissions from Complex Industrial Heater Systems. J. Environ. Eng. 131, 961–970. https://doi.org/10.1061/(ASCE)0733-9372(2005)131:6(961)

Mandel J, S.P., 2015. A Comparison of Six Methods for Missing Data Imputation. J. Biom. Biostat. 06. https://doi.org/10.4172/2155-6180.1000224

Mauzerall, D., Sultan, B., Kim, N., Bradford, D., 2005. NO emissions from large point sources: variability in ozone production, resulting health damages and economic costs. Atmos. Environ. 39, 2851–2866. https://doi.org/10.1016/j.atmosenv.2004.12.041

Nielsen, D., 2016. Tree Boosting With XGBoost. Norwegian University of Science and Technology, Trondheim, Norway.

Roth, M., Lawrence, P., 2010. A cost-effective alternative to continuous emission monitoring systems. Environ. Sci. Eng. Mag. 56–57.

Saiepour, M., Schofield, N., Leden, B., Niska, J., Link, N., Unamuno, I., Gomes, J., 2006. Development and Assessment of Predictive Emission Monitoring Systems (PEMS) Models in the Steel Industry, in: Iron and Steel Technology Conference Proceedings. Presented at the AISTech 2006, Association for Iron & Steel Technology, Cleveland, Ohio, pp. 1121–1132.

Semanjski, I., Gautama, S., 2015. Smart City Mobility Application—Gradient Boosting Trees for Mobility Prediction and Analysis Based on Crowdsourced Data. Sensors 15, 15974–15987. https://doi.org/10.3390/s150715974

Si, M., Tarnoczi, T.J., Wiens, B.M., Du, K., 2019. Development of Predictive Emissions Monitoring System Using Open Source Machine Learning Library – Keras: A Case Study on a Cogeneration Unit. IEEE Access 7, 113463–113475. https://doi.org/10.1109/ACCESS.2019.2930555

Swanson, B., 2018. CMC Solutions. Pleasant Hill, IA. Personal communication, Nov 2018.

Tan, P., Xia, J., Zhang, C., Fang, Q., Chen, G., 2016. Modeling and reduction of NOX emissions for a 700 MW coal-fired boiler with the advanced machine learning method. Energy 94, 672–679. https://doi.org/10.1016/j.energy.2015.11.020

US EPA, 2019. What Pollutants are Included in "oxides of nitrogen" in MOVES? | MOVES and Other Mobile Source Emissions Models | US EPA [WWW Document]. URL

https://www.epa.gov/moves/what-pollutants-are-included-oxides-nitrogen-moves (accessed 12.31.19).

US EPA, 2018. Performance specification 16—specifications and test procedures for predictive emission monitoring systems in stationary sources.

US EPA, 2009. Performance Specification 16 for Predictive Emissions Monitoring Systems and Amendments to Testing and Monitoring Provisions. Fed. Regist., Rules and Regulations 74, 12575–12591.

US EPA, 2006. Approval of the Predictive Emission Monitoring System Installed on Unit BL2100 at Dearborn Industrial Generation (Facility ID (ORISPL) 55088).

US EPA, 1999. Nitrogen Oxides (NOx), Why and How They are Controlled.

US EPA, 1994. An Operator's Guide To Eliminating Bias in CEM Systems (No. EPA/430/R-94-016).

Vanderhaegen, E., Deneve, M., Laget, H., Faniel, N., Mertens, J., 2010. Predictive Emissions Monitoring Using a Continuously Updating Neural Network. https://doi.org/10.1115/GT2010-22899

White, J.R., 1993. CEMs turn monitoring giant. Pollut. Eng. U. S. 25:13.

XGBoost developers, 2019. XGBoost Parameters — xgboost 1.0.0-SNAPSHOT documentation [WWW Document]. URL https://xgboost.readthedocs.io/en/latest/parameter.html (accessed 1.24.20).

Zain, S.M., Kien Kek Chua, 2011. Development of a neural network Predictive Emission Monitoring System for flue gas measurement, in: 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications. Presented at the 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, pp. 314–317. https://doi.org/10.1109/CSPA.2011.5759894

Zhang, Y., Haghani, A., 2015. A gradient boosting method to improve travel time prediction. Transp. Res. Part C Emerg. Technol. 58, 308–324. https://doi.org/10.1016/j.trc.2015.02.019

Journal Pre-proof

Journal Pre-proof

- Gradient Boosting
- Machine learning on environmental monitoring and modeling
- Predictive emissions monitoring system
- PEMS
- XGBoost

Journal Pre-proof

Journal Pre-proof

Minxing Si: Formal analysis; Methodology; Role/Writing - original draft.

Ke Du: Project administration; Funding acquisition; Role/Writing - review & editing.

Journal Pre-proof

Journal Pre-proof

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Journal Pre-proof