Received May 19, 2020, accepted June 10, 2020, date of publication June 22, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004156

Performance Prediction and Interpretation of a Refuse Plastic Fuel Fired Boiler

ADITYA PANKAJ SHAHA[1], MOHAN SAI SINGAMSETTI[2], B. K. TRIPATHY[3], (Senior Member, IEEE), GAUTAM SRIVASTAVA[4,5], (Senior Member, IEEE), MUHAMMAD BILAL[6], (Member, IEEE), AND LEWIS NKENYEREYE[7]

[1]Supply Chain Team, Grofers, Bengaluru 560029, India
[2]Department of Computing Science, University of Alberta, Edmonton, AB T6G 2R3, Canada
[3]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India
[4]Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada
[5]Research Center for Interneural Computing, China Medical University, Taichung 44020, Taiwan
[6]Division of Computer Engineering, Hankuk University of Foreign Studies, Yongin-si 17035, South Korea
[7]Department of Computer and Information Security, Sejong University, Seoul 04900- 05199, South Korea

Corresponding authors: Gautam Srivastava (srivastavag@brandonu.ca) and Lewis Nkenyereye (nkenyele@sejong.ac.kr)

This work was supported by the Sejong University New Faculty Program through the National Research Foundation of Korea NRF.

ABSTRACT In order to cater to the energy requirement in the form of steam at a reasonable cost, the process industries are relying on the waste incineration plants by engaging themselves through industry symbiosis. However, before the establishment of industrial symbiosis, it is very crucial to monitor and predict the operational performance of the boiler used in the waste incineration plants. The existing works focus on using Artificial Neural Networks (ANNs) for prediction of the performance of the boiler in terms of pressure, temperature, and mass flow rate of steam using the input parameters viz. feed water temperature, feed water pressure, incinerator exit temperature and conveyor speed. However, the problem with this approach is that shallow ANNs cannot model the complex mathematical non-linear relationships so precisely. In addition, ANNs are not interpretable which makes stakeholders apprehensive to use these networks in production. In this paper, we address these drawbacks of ANNs by modeling the complex relationship governing the boiler performance by using a set of machine learning and deep learning models. Also, the research paper introduces multiple techniques like feature importance, Partial Dependence Plots(PDP) plots etc. which interpret the reason behind the model's output to make it more reliable for the stakeholders. It has been empirically shown that the new Machine Learning and Deep Learning models performed better than the ANNs for predicting the boiler performance. The Random Forest model made a Mean Absolute Percentage Error (MAPE) of 1.12 and LSTMs had a MAPE of 1.14 in the prediction of steam temperature C^o which is a significant improvement in comparison to the original ANN model which had a MAPE of 6.93. In the case of the predictions for steam pressure kgf/cm^2 the MAPE for the Random Forest model and LSTM was 5.54 and 4.21 respectively as compared to ANNs MAPE of 1.49. Similarly for steam mass flow rate(t/h), the MAPE was improved to 15.6 and 9.63 by Random Forest Model and LSTM respectively, which was originally 18.77 for ANN based model. These results clearly show that LSTM based models outperformed ANNs and Random Forests in terms of prediction accuracy.

INDEX TERMS Long short-term memory, neural networks, random forest, predictors, LIME, refuse plastic fuel, boiler, green energy.

I. INTRODUCTION

The unprecedented increase in the demand for energy has forced mankind to look for alternative sources of energy. This has led to an increase in research in the field of generating renewable energy especially using biomass waste [1]. Industrial symbiosis systems use waste material to generate energy which is then utilized by different processes in an industrial area which has shown to decrease significantly the consumption of new raw material for energy generation and has thus in turn reduced the generation of waste and harmful emissions. The usage of heat energy recovered from waste incinerators in the form of steam is one plausible option. In order to meet the steam requirement for several processes in

The associate editor coordinating the review of this manuscript and approving it for publication was Min Xia.

processing industries, boilers are extensively used. Thus it is crucial to monitor and predict the efficiency of the boiler which is determined by steam production rates, measuring the gas consumption rates, and performing combustion analysis. However, the prediction of boiler performance is a complex function to model because of the constantly changing characteristics of boilers over prolonged periods of its operation [2], [3]. In recent years, several researchers have worked on applying Machine Learning (ML) algorithms successfully in several applications such as weather prediction, industrial automation, vehicular area networks, Internet of Things (IoT), healthcare, and many others [4], [5].

There has been some success in modeling prediction functions of fresh steam properties from a newly installed boiler using Artificial Neural Networks (ANN) [6]. In [6], for prediction of pressure, mass flow rate of steam of a refuse plastic fuel (RPF)-Boiler, and temperature, Behera *et al.* used a feed-forward neural network. The neural network used the feed water pressure, feed water temperature, conveyor speed and incinerator exit temperature as input parameters to evaluate and predict the operational performance of a boiler. There have been many papers successfully modeling the prediction of fresh steam properties from the boilers. [7], [8]. ANN is used to model complex non-linear relationships when trained with a large number of operational data points and have high accuracy on predictions [9]. However, as these neural networks are shallow they do a poor job in extracting better features from the data than their deep counterparts [10].

Recurrent Neural Networks (RNN) are a category of ANN where connections between the network nodes form a directed graph that captures the dynamics of temporally related sequences [11].In [11], Lipton *et al.* provided a comprehensive overview of RNNs for sequence learning primarily focusing on modern applications. In [12], More *et al.* used a feed-forward as well as a recurrent neural network to forecast the wind speed at the coastal locations. The model was trained in an auto-regressive manner and used backpropagation and cascade correlation algorithms. In [13] Yona *et al.* used Feed Forward Neural Networks (FFNN), Radial Basis Function Neural Network (RBFNN), and recurrent neural network (RNN) for power output forecasting of photovoltaic systems based on insolation prediction. Their work showed that RNN performed better for time-series data forecasting than FFNN. In [14] Sainlez *et al.* used a dynamic Elman's recurrent neural network and a static multilayer perceptron to model the high pressure (HP) steam flow rate from a Kraft recovery boiler. The model took raw data for one day and predicted the next 12-hours of HP steam flow production from the boiler to the steam turbine. In [15] Hochreiter showed that RNN faced the problem of vanishing and exploding gradients based on the temporal evolution of the backpropagated error exponentially which depended on the size of the weights. In [16] Hochreiter *et al.* proposed an LSTM which solved the problem of vanishing gradients thus enabling networks to solve complex, artificial long time lag tasks. In [17] Gers F. et. al. identified that LSTMs faced a problem in cases where the input streams were not apriori segmented into subsequences with explicitly marked end at which the network can reset its internal state. They solved this problem by using an adaptive ''forget gate'' using which the LSTM cell can learn to reset itself without explicit supervision and can free its internal resources. In [18] Cho *et al.* used a Gated hidden unit as activation function of an RNN which was an alternative to the conventional simple unit like element-wise tanh. This hidden gated unit was used to mitigate the RNN problem of vanishing gradients. In [19] Zhang, T. et. al. studied LSTM, GRUs and Bidirection-RNN to predict the gas concentration in coal mines. Their study showed that LSTMs outperformed the performance of BidirectionRNN and GRUs. In [20] Abdel-Nasser et. al. used a long-short term memory recurrent neural network (LSTM-RNN) to accurately forecast the output power of PV systems on datasets that were captured hourly from different sites for a year. The LSTM Networks successfully modeled the temporal changes in PV(Photovoltaic) output power which the authors attributed to recurrent architecture and memory units of LSTMs. In [21] Pathak, N et. al. used the LSTMs and Generalized Additive Models (GAM) to model the highly irregular, non-stationary, and volatile nature of gas consumption by big buildings. They empirically showed that LSTMs outperformed (GAM) but failed in case of interpretability.

Ensemble Learning in ML uses multiple learning algorithms instead of a single learning algorithm to obtain a better predictive performance on the underlying problem. Random Forests is an ensemble learning technique that uses the mode (in case of classification) and mean (in case of regression) of multiple trees trained on the same training data. Random Forest model has a few appealing characteristics which make it useful in modeling complex underlying distribution that generates the data. These properties include

1) ability to learn simple as well as complex functions
2) ability to incorporate interactions between predictors
3) ability to perform exceptionally well over default parameters and no requirement of fine tuning a lot of parameters unlike ANN
4) comparatively easy methods for defining feature selections [22]

In [22] Ahmad M. et. al. studied the artificial neural networks (ANN) with random forest (RF), an ensemble-based method that has been successfully applied in forecasting problems, for predicting the hourly HVAC(Heating Ventilation and Air Conditioning) energy consumption. The paper showed that ANNs performed marginally better than RFs. However, the random forests were shown to be more interpretable than ANNs which helped energy managers take more informed decisions based on the model predictions. In [23] Lahouar A. *et al.* used random forests method to build an hour-ahead wind power predictor based on the input parameters which were selected based on correlation and importance measures. It was empirically shown that

Random Forests performed better than neural networks. In [24] Lahouar A. *et al.* modeled the load prediction problem using random forests. The model used the season, temperature, type of the day and hourly load as input parameters to predict the power load for one hour.

Most ML models, even though accurate during the testing phase, are not used in most of the critical real-world applications. This is because a real-world task cannot be judged using a single metric [25]. To facilitate learning and make these models more trustworthy, it is necessary for machines to accompany their predictions with an explanation. ML and DL models are intrinsically black box models i.e. they are not easily interpretable. However, recently there has been ample research in this domain to make ML models interpretable. Models like Random Forests can be interpreted using permutation feature importance and tree interpreter algorithms which are specific to these models. However, there are model agnostic interpretable models like Partial Dependence Plots (PDP) [26] and Local Interpretable Model-agnostic Explanations (LIME) [27] which can be used to interpret the learning and the decision basis of these models. In [27] Ribeiro *et al.* introduced a local interpretable modeling technique Local Interpretable Model-Agnostic Explanations (LIME) which can be used to interpret the decisions made by black-box models for a particular instance. The interpretability can be used by data scientists to understand any unknown biases learned by the model or by domain experts to understand the parameters that are leading to aberrations in output.

In recent years there has been an increase in the research to predict the energy generation from boilers, furnaces, turbines etc [28]–[30]. However, most of this research has not been focused on RPF-Boilers. The objective of this study was to test the new deep learning and machine learning models to assess and optimize the performance of the RPF-Boiler as compared to the existing work done by [6]. The previous work done in this domain did not focus on making the predictive model interpretable. Our work focuses on making the intrinsically blackbox models interpretable so that their predictions can be made more reliable.

Rest of the paper is organized as follows: Section 2 discusses about the methods and methodology followed in this work, section 3 presents the results and discussion, and the paper is concluded in section 4.

II. METHODOLOGY

In this section, the methods used in the proposed work along with the methodology followed are presented.

A. DATA ACQUISITION AND DATA FILTERING

We had to deal with the processed data used in [6] and the acquisition process has been explained in that paper. This data was collected at every one minute during the period of February 2010 - June 2010. The data contained transient-data as well as steady-data and captured the following parameters: feed water temperature, steam pressure, feed water pressure, steam mass flow rate, steam temperature, incinerator exit temperature, and conveyor speed. The transient-state data was removed from the available plant and data was filtered during initial data screening. Plant operators were consulted and data points showing huge aberrations from reality were omitted because of their unreliability and non-representativeness. After the screening stage, the data was averaged for every 12 hours resulting in 215 data points with all the parameters used in the study.

B. SELECTION OF TRAINING AND TESTING DATA

In this study, 215 data points (non-randomized) were divided into training(N_{Tr}) and testing(N_{Te})sets. 70% (N_{Tr}: 150) of the initial data points for each month were used for training the network whereas the remaining 30% (N_{Te}: 65) were used for testing the developed models. The testing data set of the boiler was kept aside during the training process and was used only to check the final accuracy of the models.The (70-30) split was chosen so as to make sure that the performance on the models in this study and the base research paper are evaluated under the same conditions [6].

C. MACHINE LEARNING METHODOLOGY

1) RANDOM FOREST MODELS

Decision Trees are simple machine learning models that are highly interpretable. However, decision trees have some shortcomings like suboptimal solutions and lack of robustness. There have been different studies in the field of machine learning conducted to overcome these limitations. One of the popular and empirically successful techniques is an ensemble of trees followed by the vote of the most popular class, labeled forests. Random Forests are a class of ensemble learning techniques in which the performance of a lot of weak learners is boosted via a voting scheme [31]. In [32] Jiang *et al.* suggested that major distinctive features of Random Forests are (1) bootstrap resampling (2) Randomized Feature selection techniques (3) Out-Of-Bag (OOB) error estimations and the ability to grow full length decision trees. A random forest is an ensemble of N trees (also called the estimators) $T_1(X)$, $T_2(X)$, $T_3(X) \ldots T_N(X)$, where $X = (x_1, x_1, x_1 \ldots x_p)$ is a p-dimensional vector of inputs. Each of these trees produces an output resulting in N outputs $\hat{Y_1} = T_1(X), \hat{Y_2} = T_2(X) \ldots, \hat{Y_N} = T_N(X)$, where $\hat{Y_N}$ is the output produced by $T_N(X)$. These N outputs are then aggregated using averaging, voting etc. to output the final prediction of the random forest $\hat{Y}$. In the case of each tree in the random forest, the input space is randomly sampled with replacement from the training set to create a training set for the estimator. Some of the data points which are not sampled i.e. data points in the training data but not in the training data for the estimator from the Out- Of-Bag (OOB) samples. The newly sampled training set is used to grow an estimator using the CART(Classification And Regression Tree) algorithm. Gini index is an information theoretic measure used by the CART algorithm as a criterion for splitting the data into different nodes. The Gini index can be calculated as given by Equation 1

$$Gini = 1 - \sum_{i=1}^{C}(p_i)^2 \tag{1}$$

For every split of an estimator q random features are selected to split from a total of m features called random feature selection. This process is then repeated for all the N estimators in the random forest.

Random Forest Training procedure can be summarised as

1) Bootstrapping: Randomly sampling the training set with replacement
2) Build the estimator using the bootstrapped sample wherein at each split, of the m attributes q attributes are randomly selected and the best split among these attributes is selected using information gain. The estimator is built until there are no further splits possible.
3) Repeating steps 1 and 2 for N estimators.

2) PERMUTATION FEATURE IMPORTANCE METHOD

Random Forests are widely used in modeling many real world processes because of their ability to calculate feature importance. The importance of a variable is calculated by observing how much the prediction error changes when the value of the variable of concern in the OOB sample is permuted while keeping the value of other features unchanged. [31], [33]. There is also a problem of multicollinearity(independent variables in a regression model are correlated) while analyzing feature importance in which highly correlated features not only make interpreting the variable importance difficult but also reduce the model performance. [34], [35]. Although Lasso regression is used to do feature selection, it assigns zero weight to the highly correlated features. But, it has been shown that sometimes these features also can improve the model performance [34], [36]. Variable sampling in RF algorithm, along with bagging and bootstrapping, reduces the selection possibility of the highly correlated features [31], [37]. RF does not solve the multicollinearity problem completely. It retains the contributions of the highly correlated features without affecting much the rank of top influential features [33].

3) PARTIAL DEPENDENCE PLOTS

The Partial Dependence Plots(PDPs) are used to visualize the marginal effects of features on the dependent variable in a machine learning model [26]. A PDP can be used to draw insights about the relationship between the dependent and independent variables. Equation 2 defines the partial dependence function for regression

$$\hat{f}_{x_S}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C)d\mathbb{P}(x_C) \tag{2}$$

The set x_s defines the independent variables of concern for which the PDP is to be plotted and x_c is the remaining independent variables used in the machine learning model $\hat{f}$. Because of the limitations on the number of features that can be visualized, x_s usually contains one or two features. The dependent variable is marginalized over the variables inx_c thus only showing the relationship between the dependent variable and the variables in x_s.

Equation 3 is the partial function estimated by Monte Carlo Method i.e. calculating averages of N instances in the training data

$$\hat{f}_{x_S}(x_S) = \frac{1}{N}\sum_{i=1}^{N}\hat{f}(x_S, x_C^{(i)}) \tag{3}$$

The partial function outputs the average marginal effect on prediction for the given value(s) of features in X_s. PDP assumes that variables in X_c are not correlated with the variables in X_s [38].

4) TREE INTERPRETER

Since prediction can be defined as a sum of features contributions added with the bias which is the mean of the entire training set, tree interpreter defines the predictions as the additions or subtractions the feature used for splitting makes to the value at the parent node.

Equation 4 defines the feature contribution for a decision tree.

$$\hat{Y} = Y^{mean} + \sum_{f} FC^{f} \tag{4}$$

In Equation 4, $\hat{Y}$ is the prediction and Y^{mean} is the mean of the training sample of the tree,f is the number of features and FC is the Feature Contribution.

This equation for feature contribution looks similar to the Linear Regression equation. However, in the case of linear regression, coefficients of the variables are independent but in the case of decision trees, the contribution by each feature depends on the rest of the feature vectors which are responsible for defining the decision path. This in-turn defines the contributions passed by each feature along the way. An RF is a collection of estimators. Their prediction is the average of the predictions by all the estimators. So, in the case of a forest, the contribution of each feature is the average of bias of all the estimators added to the contribution of each feature.

Equation 5 defines the feature contribution for random forest.

$$\hat{Y} = \frac{1}{T}\sum_{t=1}^{t=T} Y_t^{mean} + \sum_{f=1}^{f=F}(\frac{1}{T}\sum_{t=1}^{t=T} FC_t^{f}) \tag{5}$$

In Equation 5, $\hat{Y}$ is the prediction and Y_t^{mean} is the mean of the training sample of the tree t, f is the number of features and FC_t^f is the feature contribution of feature f in tree t and T is the total number of estimators in the random forest.

D. DEEP LEARNING METHODOLOGY

1) DEEP LEARNING MODEL AND PARAMETER OPTIMIZATION ALGORITHM

The traditional feed forward neural network model has connections between the input layers, hidden layer and output layer. These connections do not form a cycle. Every node in a layer is connected to every other node in the next layer while there exiss no connections within layers. This architecture has some limitations in modeling data wherein there exists some relation within the data points like the time series data wherein the previous data point can be highly useful in predicting the output of the current datapoint. This shortcoming of feed forward neural networks is solved by using recurrent neural networks in which the connections between nodes form a directed graph along the direction in which the time series progresses. Thus RNNs are able to use the internal state which is the context from the previous set of inputs to predict the output of the current state. In Equation 6 X indicates the value of the input layer responsible for ingesting and processing the input data, S represents the value of the hidden layer, U is the weight matrix of the input layer to the hidden layer, O represents the value of the output layer, V is the weight matrix of the hidden layer to the output layer, and W is the last value of the hidden layer as the input of this time. The subscript t states the time-step in the sequence.

$$\begin{cases} O = g(VS_t) \\ S_t = f(UX_t + WS_{t-1}) \end{cases} \tag{6}$$

The architecture of RNNs enables them to model nonlinear time series data but they face the limitation of not being able to model data with excessive delays because of the gradient vanishing and gradient explosion problems. LSTM Networks solved the shortcomings faced by RNNs using the input gate, a forget gate, and an output gate. The LSTM architecture is shown in Figure 1 where f_t is the forget gate, i_t is the input gate, C_t is the output gate and O_t are the timings of the output gates. The forget gate is responsible for deciding the amount of the unit state C_{t-1} at the previous moment which can be retained in the current state C_t. The input gate is responsible to determine the amount input at time t X_t of the network is saved to the unit state C_t at the current time step and the amount the output gate control unit state C_t is output to the LSTM. The value of the current output h_t is calculated by the Equations 9. The Equations (7,8,9) are of the forget gate, input gate and output gate respectively.

$$f_t = \delta(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

$$\begin{cases} i_t = \delta(w_i \cdot [h_{t-1}, x_t] + b_i) \\ C' = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t \end{cases} \tag{8}$$

$$\begin{cases} O_t = \delta(w_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = O_t \cdot tanh(C_t) \end{cases} \tag{9}$$

A variation of LSTM is a Gated Recurrent Unit which has an update gate and a reset gate. The update gate is used to update the contextual information and reset gate determines the output information.

Adaptive Moment estimation (Adam) is a simple and memory efficient optimization algorithm based on gradients which is suitable for nonstationary objective functions. It has a benefit of intuitive hyperparameter interpretations and complex tuning procedures are not necessary. Its update mechanism can be explained by the following equations where Equations 10,11 handle deviation correction and Equation 12 defines parameter changes

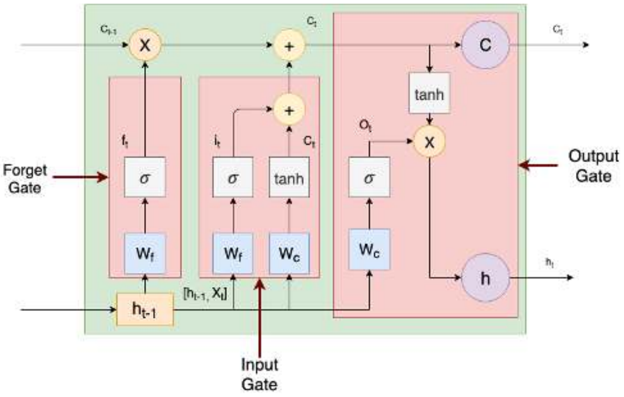


FIGURE 1. LSTM Cell.

$$\begin{cases} m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \\ v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \end{cases} \tag{10}$$

$$\begin{cases} m'_t = \dfrac{m_t}{(1 - \beta_1^t)} \\ v'_t = \dfrac{v_t}{(1 - \beta_2^t)} \end{cases} \tag{11}$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{m'_t}{\sqrt{v'_t} + \epsilon} \tag{12}$$

E. INTERPRETABILITY USING LIME

Local surrogate models interpret the machine learning model's predictions based on a small neighbourhood of the data point. This data point is the one for which the decision basis of the machine learning model needs to be inferred. In [27] the authors introduced LIME, an implementation of the local surrogate models which focus on training local surrogate models which are used to explain individual predictions from the underlying black-box model.

LIME perturbs the dataset and gets predictions from the black-box models for the new training dataset. The new-training dataset is then used to train an interpretable model for which the instances in the vicinity of point of interest are given more weightage than the instances that are far away. The newly learned model is a good local approximation but is not necessarily a good global approximation.

$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_{(x)}) + \Omega(g) \tag{13}$$

In Equation 13, g is the interpretable model which explains the decision by the black box model for instance x. g minimizes loss L which is a metric to measure how close the original model(f)'s prediction is while the complexity of the model $\Omega(g)$ is kept low. G is a set of all possible interpretable models. $\pi_(x)$ is the proximity measure that defines

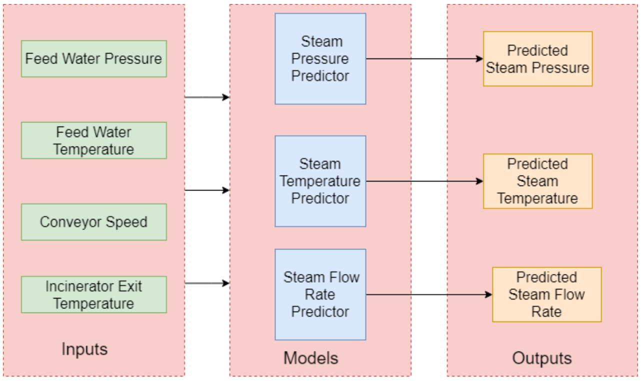

Feed Water Pressure
Feed Water Temperature
Conveyor Speed
Incinerator Exit Temperature
Inputs
Steam Pressure Predictor
Steam Temperature Predictor
Steam Flow Rate Predictor
Models
Predicted Steam Pressure
Predicted Steam Temperature
Predicted Steam Flow Rate
Outputs

FIGURE 2. Random Forest Model Framework.

the permissible neighborhood around instance x. In the implementation, LIME only optimizes the loss and the complexity is determined by the user by selecting the maximum number of features.

F. CONSTRUCTION OF PREDICTION MODEL

1) RANDOM FOREST BASED PREDICTION MODEL

Three different random forest models were built to predict the mass flow rate of steam, pressure, and temperature respectively. These models used feed water pressure, feed water temperature, conveyor speed, and incinerator exit temperature as the input parameters. Mean Squared Error (MSE) criterion used to measure the quality of the split.

During the prediction time, the readings of feed water pressure, feed water temperature, conveyor speed, and incinerator exit temperature from the boiler are fed to the three models simultaneously and the output is predicted for each input data point.

2) DEEP LEARNING BASED PREDICTION MODEL

The architecture of a basic LSTM model can be divided into 3 major parts viz. the input layer, the hidden layer, and the output layer. The input layer has the dimension of the number of features of the input and is responsible for processing the input. The weights to the hidden layer are learned from the training data using the learning algorithm like Back Propagation Through Time (BPTT). Learning algorithms can use different optimizers like Adam, Adagrad etc. which is used for model parameter optimization to achieve minimum loss value for the objective function like Mean Squared Error (MSE) or Mean Absolute Percentage Error (MAPE). The output layer models the values received from the hidden layer to the output dimension and rescale the data. Figure 3 shows the architecture of the LSTM model framework.

In the forecast model the hidden layer is the LSTM cell. The model is empirically shown to perform the best when a lookback of one data point is available as an input to predict for the next time step. The MinMaxScaler normalizer is used to scale the training and testing data into 0 and 1. Equation 14 represents the formula for MinMaxScalar

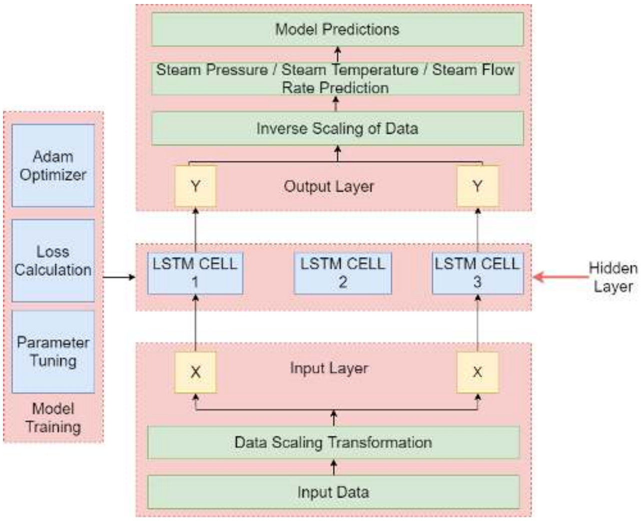


FIGURE 3. LSTM Model Framework.

$$\begin{cases} x_{std} = \dfrac{x - min(X)}{max(X) - min(X)} \\ x_{new} = x_{std} \times (max - min) + min \end{cases} \tag{14}$$

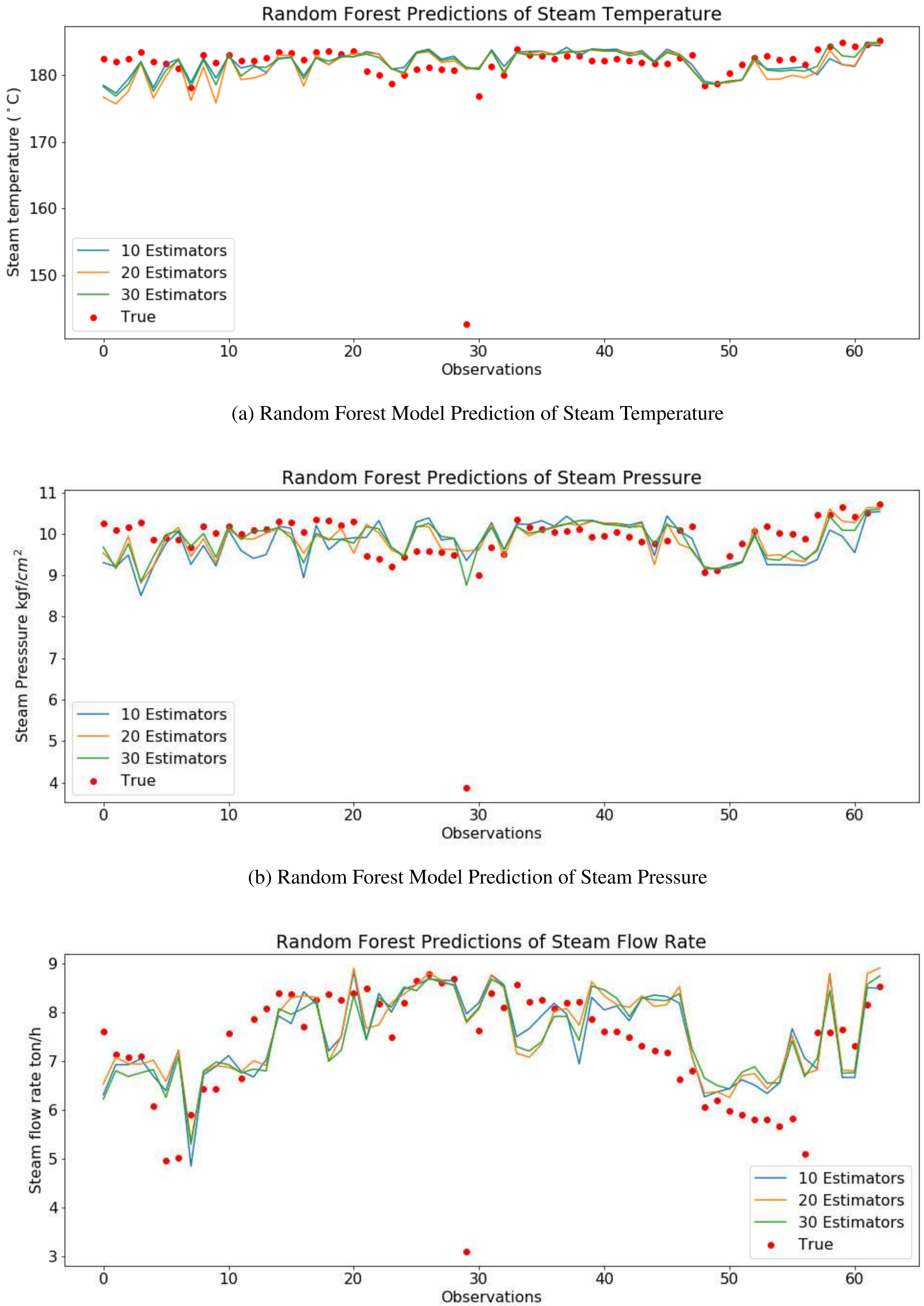
In Equation 14, $min(X)$ represents the minimum value of the feature to which x belongs and $max(X)$ represents the maximum value of the feature. min and max are the range of the training dataset.

Once the inputs are scaled by the input layer the LSTM cells in the hidden layer operate on them as represented in Equations 7 -9. The processed information is then passed to the output layer where it is re-scaled to the original magnitude using inverse scalar.

III. EXPERIMENTS AND RESULTS

The experiments carried out verifies the performance of Random Forest Model, LSTM, and GRU Networks in comparison to MLP (Multi-Layer Perceptron) Model used in [6] to model the boiler performance. The experiment built 3 models that used 4 input variables each viz. conveyor speed, feed water temperature, feed water pressure, and incinerator exit temperature. These three models were used to predict the steam pressure, steam temperature, and the steam flow rate respectively.

For experimentation, the following computer configurations are used: Windows 10 Operating System with 64-bit, Intel Core i5-4210U CPU processor with 1.7GHz, memory of 8 GB and the development language used is Python 3.6. The LSTM and GRU models used in the experiment were built using Keras 2.2.4 package with tensorflow backend version 1.9.0. The Random Forest model was built using scikit-learn version 0.19.1 library. The plots and graph used to visualize the output of the network used matplotlib 2.2.3 library and numpy 1.14.3 was used for



(a) Random Forest Model Prediction of Steam Temperature

(b) Random Forest Model Prediction of Steam Pressure

(c) Random Forest Model Prediction of Steam Flow Rate

FIGURE 4. Random Forest Model Predictions.

numerical computation on high dimensional tensors. The interpretability of networks was enabled by lime version 0.1.1.34 library and tree interpreter was used to interpret tree output.

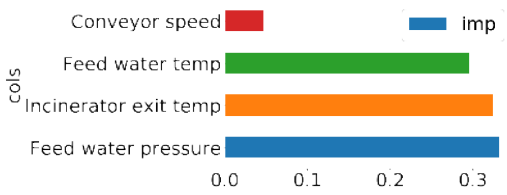


FIGURE 5. Feature Importance for predicting Steam Temperature.

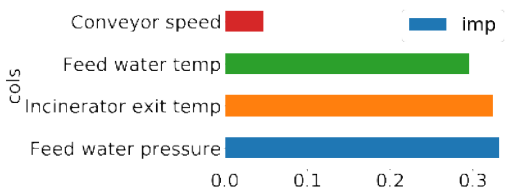


FIGURE 6. Feature Importance for predicting Steam Pressure.

A. RESULTS OF RANDOM FOREST MODEL

Table 1 shows the performance of the Random Forest Model for different hyper parameters. The table compares the performance of these models in terms of different statistical metrics viz. Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) on the testing data of Steam Temperature, Steam Pressure, and Steam Flow Rate. Random Forest model is able to capture the sudden minor aberrations in the steam flow rate.

The observations show that the random forest model with 10 estimators performs best for predicting the steam temperature, steam pressure and steam flow rate. Figure 4 shows the prediction of steam temperature, steam pressure and steam flow rate of the random forest model with different numbers of estimators. The random forest model showed comparatively better results to model for sudden aberrations in the output values within a range. For example, during testing, when the steam flow rate fluctuated from 7.09 to 6.08 the random forest model made an MAE of 0.9. Similarly in the next instance, the value changed from 6.08 to 4.95 the MAE was 1.44.

B. INTERPRETABILITY OF RANDOM FOREST MODEL

The experimental results are augmented with the interpretability of the model which lets the stakeholders and domain experts get insights on the basis on which a model makes it’s predictions. This is especially useful when a model forecasts anomalous outputs and the domain experts want to know what is the cause for this output or the data scientist wants to get insights into what caused the model to make this error so that it can be fixed for future forecasts. The interpretability of the model used in this experiment was validated by domain experts.

1) PEARSON CORRELATION

A Pearson correlation Matrix indicates the extent to which variables are linearly related. For any two variables in the matrix, the value lies between -1 to 1 which shows the extent to which these variables are positively or negatively correlated.

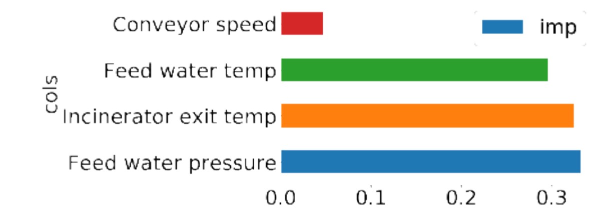


FIGURE 7. Feature Importance for predicting Steam Flow Rate.

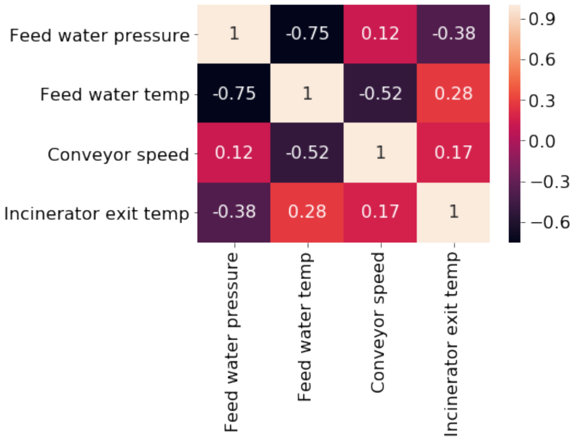


FIGURE 8. Pearson Correlation Matrix.

Figure 8 shows that Incinerator exit temperature is negatively correlated with feed water pressure and positively correlated to feed water temperature. Conveyor speed is highly negatively correlated to feed water temperature. Feed water temperature is highly negatively correlated feed water pressure.

2) FEATURE IMPORTANCE OF RANDOM FOREST MODEL

Figure 5 shows that the model trained to predict the Steam Temperature gave highest importance to the feed water temperature following incineration exit temperature, Conveyor speed and feed water pressure in that order. Figure 6 exhibits that feed water temperature and feed water pressure were equally important in determining Steam Pressure followed by the incineration exit temperature, Conveyor speed. Figure 7 shows that feed water temperature was given the highest importance for predicting Steam Flow rate followed by feed water temperature following Conveyor speed, feed water pressure and incineration exit temperature in that order.

3) PARTIAL DEPENDENCE PLOTS

Figures [17]–[25] shows the Partial Dependence Plots(PDP) for the Random Forest model. Figure 17 shows that to achieve a steam pressure of more 10.0 kgf/cm^2 the feed water temperature has to be between 45 and 60 $°C$ and

TABLE 1. **Comparison of RF Models to predict Steam Temperature, Steam Flow Rate and Steam Pressure on Different Models with different number of estimators.**

Model (Estimators)	Steam Temperature(a)				Steam Flow Rate(b)				Steam Pressure(c)			
	Time(ms)	MAPE	RMSE	MAE	Time(ms)	MAPE	RMSE	MAE	Time(ms)	MAPE	RMSE	MAE
Model 1 (10 Estimators)	110	1.21	5.16	2.04	112	10.64	0.91	0.64	111	5.84	0.82	0.44
Model 2 (20 Estimators)	113	1.25	5.16	2.12	113	11.42	1.02	0.69	117	6.08	0.83	0.47
Model 3 (30 Estimators)	119	1.26	5.20	2.17	118	12.62	1.06	0.77	120	6.11	0.83	0.47

the Conveyor speed has to be between 12 to 19 *rpm*. As the feed water temperature decreases from 45 and increases from 60, the steam pressure decreases monotonically. Figure 18 shows that to achieve a steam pressure of more 10.0 kgf/cm^2 the feed water pressure has to be between 14 and 16 kgf/cm^2 and feed water temperature the has to be between 40 to 60 $°C$. As the feed water pressure decreases from 14, the steam pressure decreases monotonically. It can be interpreted from Figure 19 that to achieve a steam pressure of more than 9.75 kgf/cm^2, the conveyor speed should be more than 17 *rpm* when the incineration temperature is more than 900 $°C$. Figure 20 shows the interaction between conveyor speed and feed water temperature and how it affects the steam temperature collectively. It can be seen that to achieve a steam temperature between 182 - 183 $°C$, the feed water temperature should be more than 20 $°C$ when the conveyor speed is between 12 and 25 *rpm*. Steam temperature decreases monotonically with feed water temperature when it is less than 20 $°C$. Figure 21 shows that to achieve a steam temperature between 182 - 184 $°C$ the feed water pressure should be between 14 and 17 kgf/cm^2 and the feed water temperature should be between 40 and 65 $°C$ or between 20 and 30 $°C$. The steam temperature increases monotonically with the feed water temperature when it is between 0 - 40 $°C$. Figure 22 shows that to achieve a steam temperature between 182 - 183 $°C$ the incineration temperature should be more than 900 $°C$ when the conveyor speed is less than 32 *rpm*. Figure 23 shows the interaction between conveyor speed and feed water temperature and how it affects the steam flow rate collectively. To achieve the steam flow rate between 7.2 - 8 t/h, the feed water temperature should be more than 25 $°C$ and the conveyor speed should be less than 25 *rpm*. Figure 24 shows that to achieve a steam flow rate between 8 - 9 t/h the feed water pressure should be between 16 - 18 kgf/cm^2 and feed water temperature should be more than 25 $°C$. Figure 25 shows that achieve a steam flow rate between 8 - 8.8 t/h the incineration temperature should be more than 1000 $°C$ and the conveyor speed should be between 15 and 22.5 *rpm*.

4) TREE INTERPRETER OUTPUT

Tree Interpreter was used to interpret the output for the instance in Table 2 in the test dataset. The instance was randomly chosen from the testing data to analyze which attributes were given higher weightages by the model during the prediction phase.

Table 3 shows the weightages assigned by model to the independent variables for interpreting the Steam Temperature, Steam Pressure, and Steam Flow Rate. It can be inferred from the data that the highest weightage was given to Conveyor Speed to predict the Steam Temperature and Steam Pressure and Feed water pressure to predict Steam Flow Rate

C. RESULTS OF DEEP LEARNING MODELS

The number of cascaded LSTM cells not only has an impact on the model's learning ability, training, and testing time but also adds up to the complexity of the model. Although empirically and theoretically deeper models have more learning ability but it also increases the training and the testing time of the model. In the paper, it was empirically shown that networks with 2 and 3 LSTM Cells performed better in terms of accuracy but poor when it came to the execution time.

The screened data refined the unscreened data and compressed it to a few observations representative of a sequence of data points in the unscreened data. So empirically, batch size of 1 performed better with respect to training speed and accuracy.LSTM and GRU based models did not perform well with minor aberrations in comparison to the Random Forest Model. For example, during testing when the steam flow rate fluctuated from 7.09 to 6.08 the LSTM model made a MAPE of 1.5 and the GRU model had a MAPE of 1.4 Similarly in the next instance the value changed from 6.08 to 4.95 the LSTM MAPE was 2.78 whereas the GRU model had a MAPE of 2.8. So, from the observations that we made in section 3.A, it can be inferred that in cases where the inputs to the boiler are going to fluctuate frequently, a random forest model would work comparatively better.

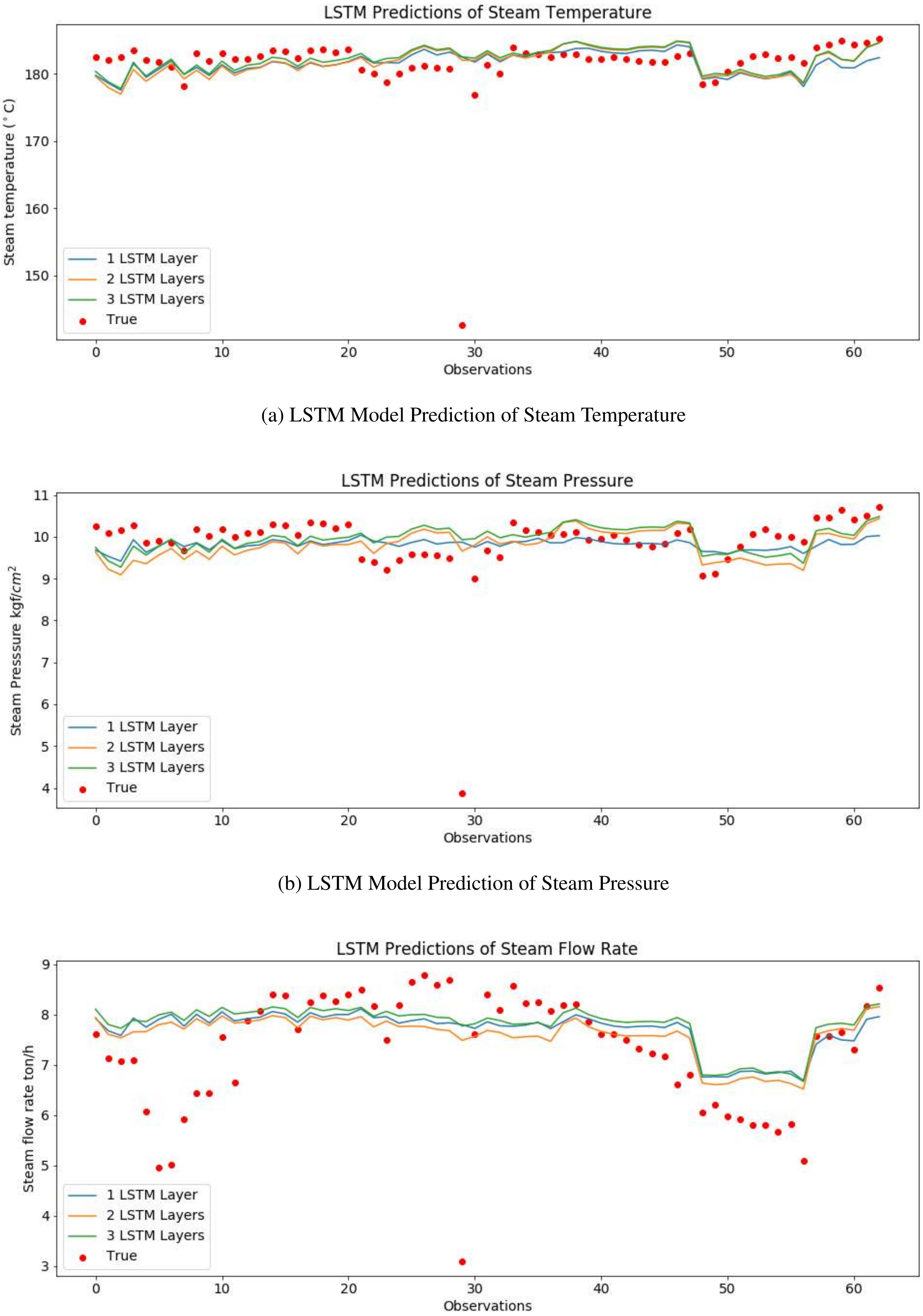
Figure 9 shows the predictions made by LSTM models with 1,2 and 3 number of LSTM layers. It can be inferred from Table 4 and these figures that the model with 2 and 3 LSTM Cells performed better in terms of accuracy except for Steam Temperature where the model with 1 LSTM cell outperformed the others.

Similar experiments were performed on tuning the number of layers for GRU Networks and even in the case of GRU, it was experimentally observed that GRUs with 2 GRU Layers performed empirically better with a batch size of 1.

Comparative analysis in Table 4 shows that LSTMs perform marginally better than GRUs and both the deep learning models perform better than Random Forest Models. Similarly, Table 4 also shows that LSTMs are empirically better performers in the case of 3 LSTM Cells as compared to 3 GRU Cells.

D. INTERPRETABILITY OF MODELS USING LIME

LIME was used on LSTM and GRU models to interpret the local decision making basis of these black box models.



(a) LSTM Model Prediction of Steam Temperature

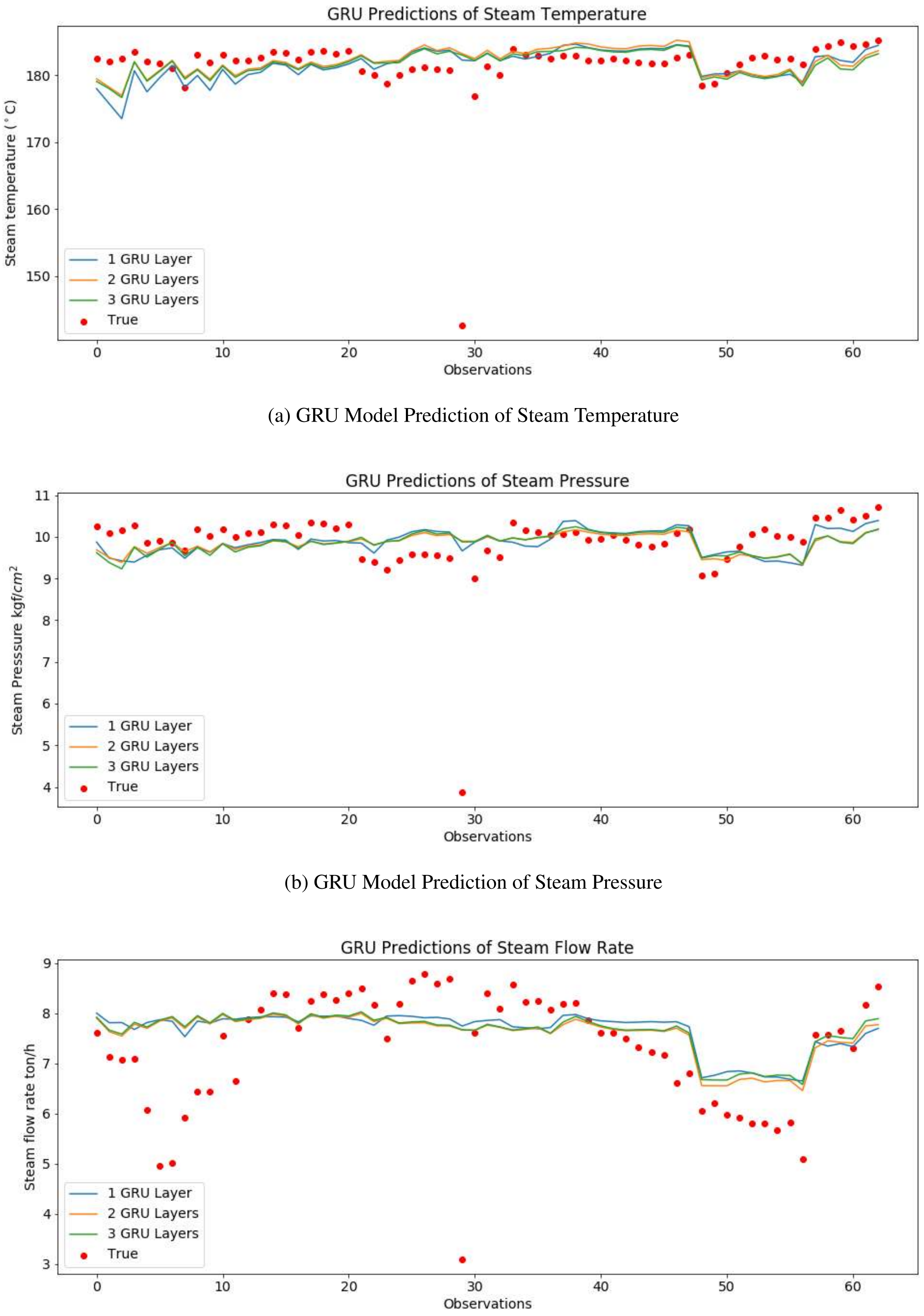
(b) LSTM Model Prediction of Steam Pressure

(c) LSTM Model Prediction of Steam Flow Rate

FIGURE 9. LSTM Model Predictions.

Figure 11 - Figure 16 shows the relative importance given by the models to predict the Steam Flow Rate, Steam Temperature and, Steam Pressure for the test instance in Table 5.

It can be inferred from the Figures 11 14, that both LSTM and GRU models give highest weightage to Feed Water Temperature followed by Feed water pressure. The



(a) GRU Model Prediction of Steam Temperature

(b) GRU Model Prediction of Steam Pressure

(c) GRU Model Prediction of Steam Flow Rate

FIGURE 10. **GRU Model Predictions.**

decision boundary is scaled value because the model is fed the scaled input data. Similarly Figures 16, 13 show that LSTM and GRU models give the highest weightage to feed water pressure and negative weightage to feed water temperature above a scaled threshold. Figure 15 and Figure 12 both show that the GRU model and LSTM model

TABLE 2. Sample from test the data.

Feed water pressure	Feed water temperature	Conveyor speed	Incinerator exit temp	Steam Pressure	Steam Temperature	Steam Flow Rate
16.5	18.0	23.11	847.0	10.24	182.5	7.61

TABLE 3. Tree Interpreter Outputs for sample instance.

Models	Feed water pressure	Feed water temperature	Conveyor speed	Incinerator exit temp	Predictions
Steam Temperature Regressor	-0.97	0.87	1.80	-1.45	181.979
Steam Pressure Regressor	-0.34	0.17	0.30	-0.14	9.825
Steam Flow Rate Regressor	0.68	-1.47	0.42	-0.07	6.615

TABLE 4. Performance on Steam Pressure, Steam Temperature and Steam Flow Rate for different number of LSTM and GRU Cells.

Models	Steam Temperature(a)				Steam Flow Rate(b)				Steam Pressure(c)			
	Epochs	MAPE	RMSE	MAE	Epochs	MAPE	RMSE	MAE	Epochs	MAPE	RMSE	MAE
LSTM Model 1 (1 cell)	31	1.18	5.19	2.14	16	11.39	1.19	0.86	25	6.33	0.99	0.60
LSTM Model 2 (2 cells)	16	1.20	5.24	2.19	11	9.78	1.00	0.72	15	5.97	0.92	0.57
LSTM Model 3 (3 cells)	11	1.51	5.54	2.74	9	8.94	0.97	0.67	19	4.28	0.84	0.42
GRU Model 1 (1 cell)	20	1.48	5.53	2.68	20	9.31	1.03	0.70	29	5.04	0.86	0.48
GRU Model 2 (2 cells)	25	1.21	5.25	2.20	17	9.36	1.02	0.71	15	4.78	0.85	0.47
GRU Model 3 (3 cells)	27	1.52	5.53	2.76	19	9.39	1.02	0.71	14	5.29	0.90	0.52

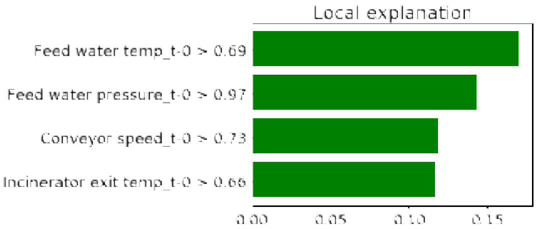


FIGURE 11. LSTM Model LIME Feature Importance for predicting Steam Pressure.

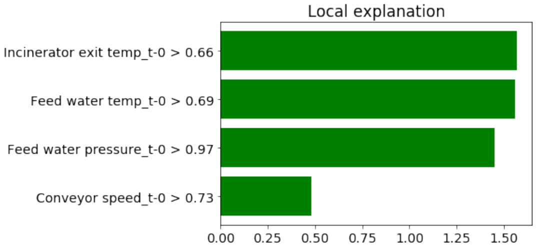


FIGURE 12. LSTM Model LIME Feature Importance for predicting Steam Temperature.

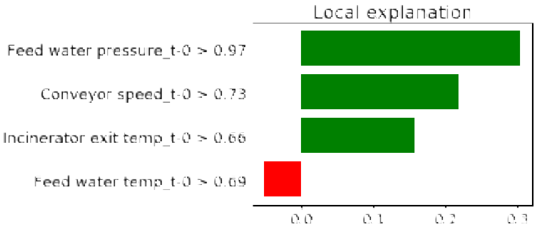


FIGURE 13. LSTM's Model LIME Feature Importance for predicting Steam Flow Rate.

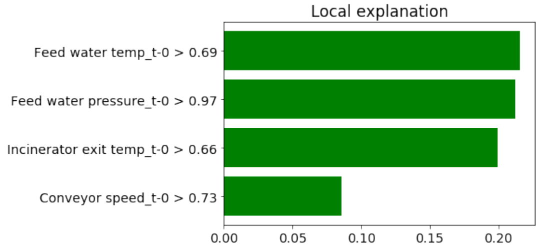


FIGURE 14. GRU Model LIME Feature Importance for predicting Steam Pressure.

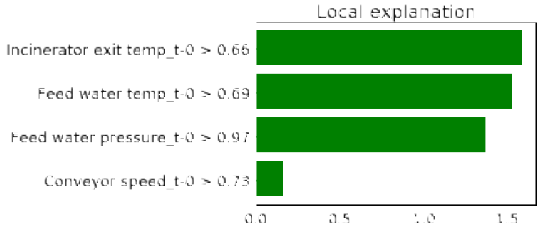


FIGURE 15. GRU Model LIME Feature Importance for predicting Steam Temperature.

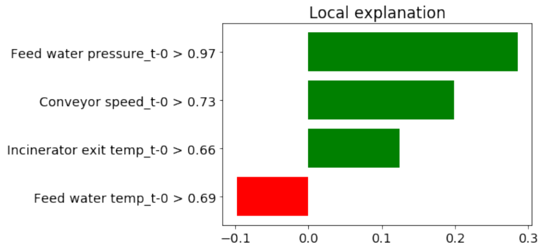


FIGURE 16. GRU Model LIME Feature Importance for predicting Steam Flow Rate.

give the highest weightage to incinerator exit temperature to predict the Steam Pressure for the test instance. These values are different from actual feature importance values because these values are used by these models to make decisions for a particular instance and not for the entire dataset.

TABLE 5. Instance of Test Data for LIME Interpretation.

Feed water pressure	Feed water temperature	Conveyor speed	Incinerator exit temp
16.67	15.5	17.55	781.5

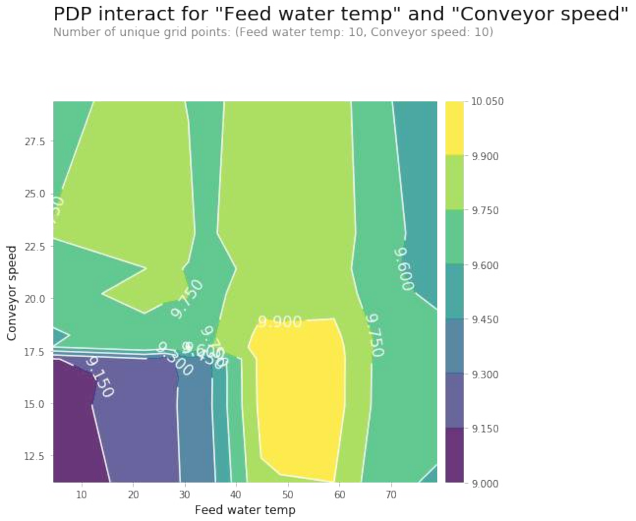


FIGURE 17. Interaction plot between Feed Water Temperature and Conveyor Speed showing effects on Steam Pressure.

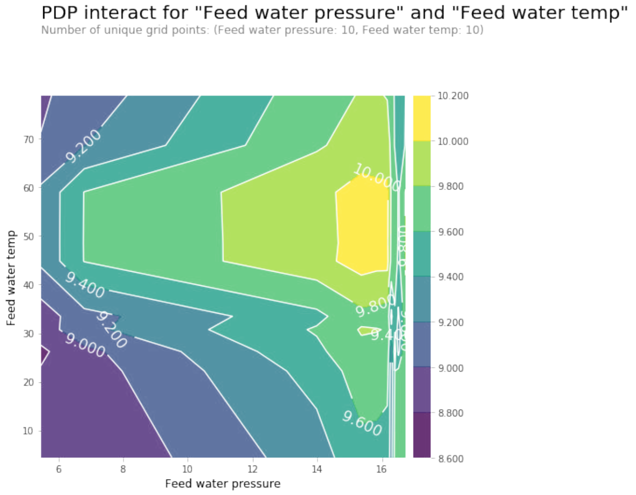


FIGURE 18. Interaction plot between Feed Water Temperature and Feed Water Pressure showing effects on Steam Pressure.

IV. CONCLUSION

A machine learning model viz. Random Forest and two deep learning models LSTM network and GRU network were used to model the steam temperature, steam pressure, and steam flow rate of the rpf-boiler. Random forest model comparatively performed better in capturing the sudden variations in data but performed poorly in comparison to the cyclic neural network variants viz. LSTM networks and GRU networks. The hypothesis that having knowledge of the previous state of the boiler would be useful in predicting the next state was empirically proven to be valid. The drawbacks of the black box models like neural networks and random forests

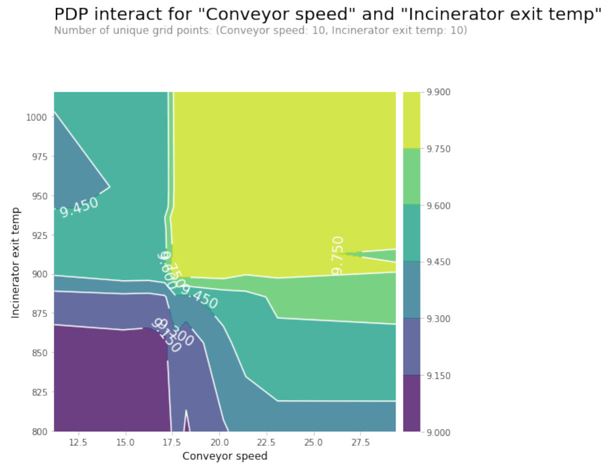


FIGURE 19. Interaction plot between Conveyor Speed and Incinerator exit temperature showing effects on Steam Pressure.

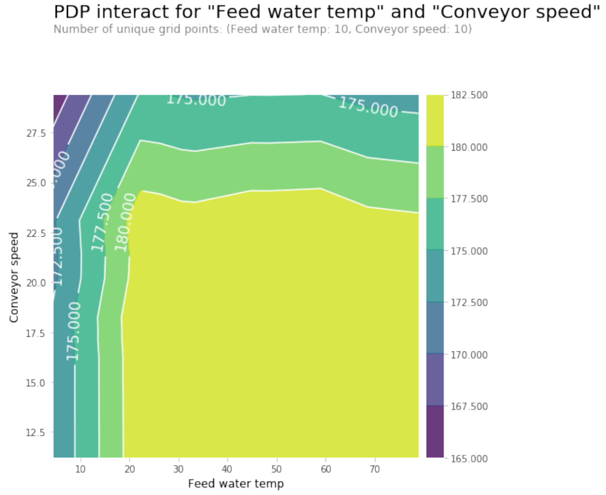


FIGURE 20. Interaction plot between Feed Water Temperature and Conveyor Speed showing effects on Steam Temperature.

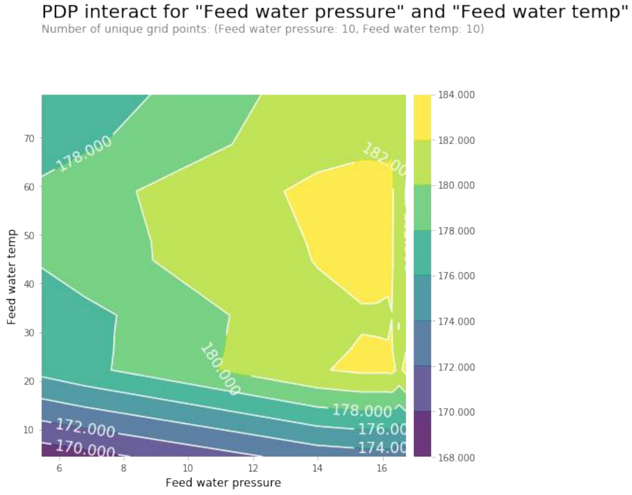


FIGURE 21. Interaction plot between Feed Water Temperature and Feed Water Pressure showing effects on Steam Temperature.

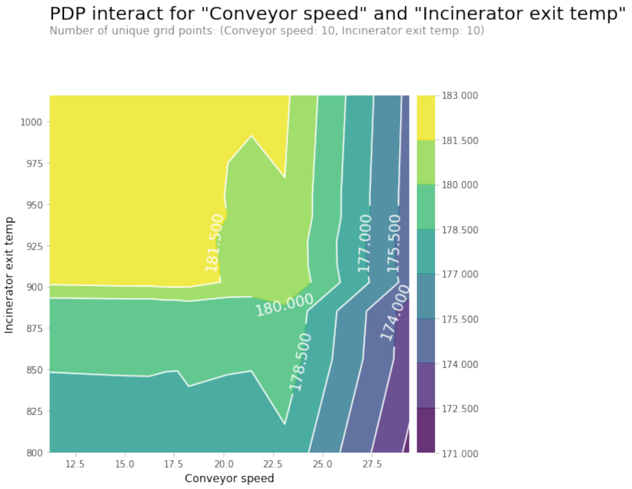


FIGURE 22. Interaction plot between Conveyor Speed and Incinerator exit temperature showing effects on Steam Temperature.

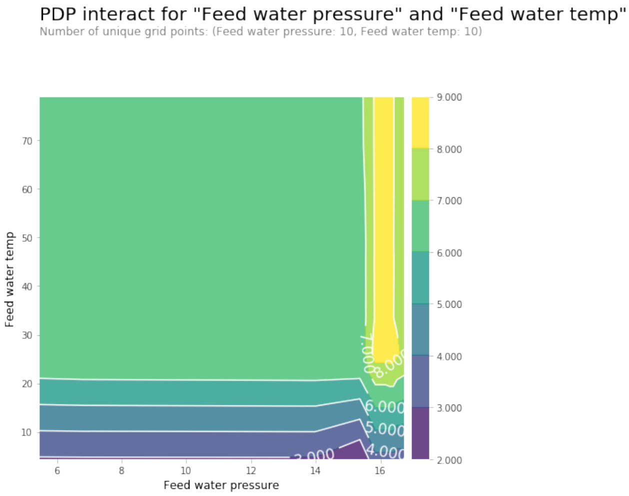


FIGURE 24. Interaction plot between Feed Water Temperature and Feed Water Pressure showing effects on Steam Flow Rate.

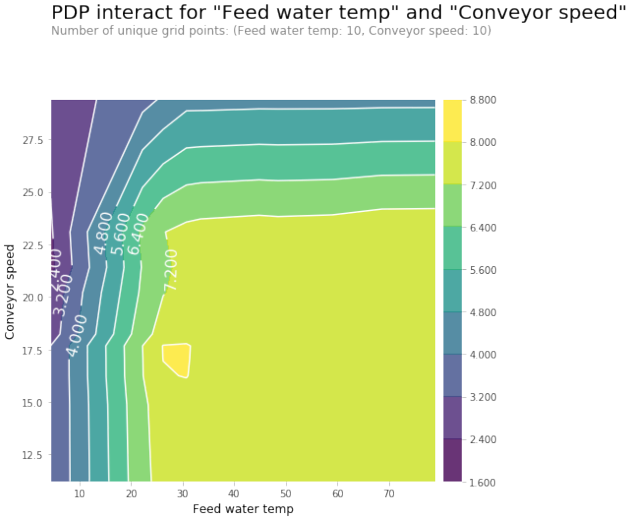


FIGURE 23. Interaction plot between Feed Water Temperature and Conveyor Speed showing effects on Steam Flow Rate.

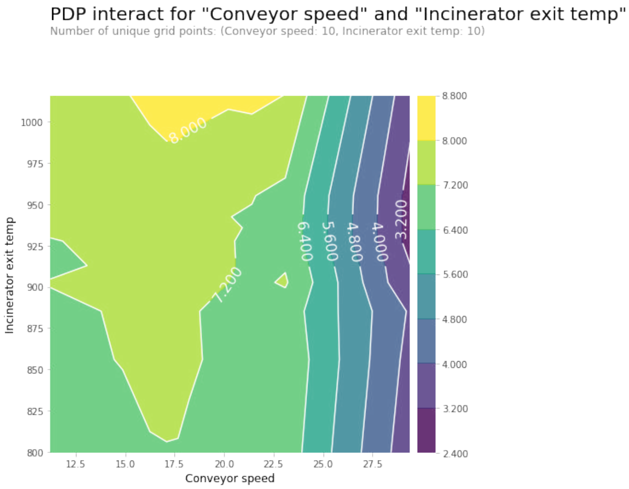


FIGURE 25. Interaction plot between Conveyor Speed and Incinerator exit temperature showing effects on Steam Flow Rate.

are their lack of interpretability which makes the stakeholders without domain knowledge apprehensive to use these models in production systems. This drawback was tackled by using multiple interpreting tools like permutation feature importance, Tree Interpreter, and Partial Dependence Plots in random forests and LIME for Deep learning models. The interpretability of a model was used to understand the decision making basis of the model and get visibility into the learnings of the model. For example, if the predictive model suggested that the steam pressure is showing a sudden increase by two times, the operator can infer from the PDP plots that this increase is because of increase in feed water temperature from original $30°C$ to $40°C$. This can help the operator manage the steam pressure outputs by moderating the feed water temperature. Not only was this able to increase the confidence of the operator on the model predictions, but also it helped researchers understand that the model predictions are not based on some spurious correlations. In future, we plan on building a real time control system which can use these forecasts to manage the boiler to achieve the planned energy output.

APPENDIX

PARTIAL DEPENDENCE PLOTS

See Fig. 17–25.

REFERENCES

[1] K. W. Chew, S. R. Chia, H.-W. Yen, S. Nomanbhay, Y.-C. Ho, and P. L. Show, "Transformation of biomass waste into sustainable organic fertilizers," *Sustainability*, vol. 11, no. 8, p. 2266, Apr. 2019.

[2] M. Si, T. J. Tarnoczi, B. M. Wiens, and K. Du, "Development of predictive emissions monitoring system using open source machine learning library—Keras: A case study on a cogeneration unit," *IEEE Access*, vol. 7, pp. 113463–113475, 2019.

[3] M. Moleda, A. Momot, and D. Mrozek, "Predictive maintenance of boiler feed water pumps using SCADA data," *Sensors*, vol. 20, no. 2, p. 571, Jan. 2020.

[4] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[5] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, D. S. Rajput, R. Kaluri, and G. Srivastava, "Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis," *Evol. Intell.*, vol. 13, pp. 185–196, Nov. 2019.

[6] S. K. Behera, E. R. Rene, M. C. Kim, and H.-S. Park, "Performance prediction of a RPF-fired boiler using artificial neural networks," *Int. J. Energy Res.*, vol. 38, no. 8, pp. 995–1007, Jun. 2014.

[7] J. Smrekar, M. Assadi, M. Fast, I. Kuštrin, and S. De, "Development of artificial neural network model for a coal-fired boiler using real plant data," *Energy*, vol. 34, no. 2, pp. 144–152, Feb. 2009.

[8] M. Kljajić, D. Gvozdenac, and S. Vukmirović, "Use of neural networks for modeling and predicting boiler's operating performance," *Energy*, vol. 45, no. 1, pp. 304–311, Sep. 2012.

[9] S. De, M. Kaiadi, M. Fast, and M. Assadi, "Development of an artificial neural network model for the steam process of a coal biomass cofired combined heat and power (CHP) plant in sweden," *Energy*, vol. 32, no. 11, pp. 2099–2109, Nov. 2007.

[10] F. Gökgöz and F. Filiz, "Electricity price forecasting: A comparative analysis with shallow-ann and dnn," *Int. J. Energy Power Eng.*, vol. 12, no. 6, pp. 421–425, 2018.

[11] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*. [Online]. Available: http://arxiv.org/abs/1506.00019

[12] A. More and M. Deo, "Forecasting wind with neural networks," *Mar. Struct.*, vol. 16, no. 1, pp. 35–49, 2003.

[13] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, and C.-H. Kim, "Application of neural network to 24-hour-ahead generating power forecasting for PV system," in *Proc. IEEE Power Energy Soc. Gen. Meeting-Convers. Del. Electr. Energy 21st Century*, Jul. 2008, pp. 1–6.

[14] M. Sainlez and G. Heyen, "Recurrent neural network prediction of steam production in a kraft recovery boiler," in *Computer Aided Chemical Engineering*, vol. 29. Amsterdam, The Netherlands: Elsevier, 2011, pp. 1784–1788.

[15] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[19] T. Zhang, S. Song, S. Li, L. Ma, S. Pan, and L. Han, "Research on gas concentration prediction models based on LSTM multidimensional time series," *Energies*, vol. 12, no. 1, p. 161, Jan. 2019.

[20] M. Abdel-Nasser and K. Mahmoud, "Accurate photovoltaic power forecasting models using deep LSTM-RNN," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2727–2740, Jul. 2019.

[21] N. Pathak, A. Ba, J. Ploennigs, and N. Roy, "Forecasting gas usage for big buildings using generalized additive models and deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 203–210.

[22] M. W. Ahmad, M. Mourshed, and Y. Rezgui, "Trees vs neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption," *Energy Buildings*, vol. 147, pp. 77–89, Jul. 2017.

[23] A. Lahouar and J. B. H. Slama, "Hour-ahead wind power forecast based on random forests," *Renew. Energy*, vol. 109, pp. 529–541, Aug. 2017.

[24] A. Lahouar and J. Ben Hadj Slama, "Day-ahead load forecast using random forest and expert input selection," *Energy Convers. Manage.*, vol. 103, pp. 1040–1051, Oct. 2015.

[25] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*. [Online]. Available: http://arxiv.org/abs/1702.08608

[26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, Oct. 2001.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.

[28] X. Hu, P. Niu, J. Wang, and X. Zhang, "Multi-objective prediction of coal-fired boiler with a deep hybrid neural networks," *Atmos. Pollut. Res.*, vol. 11, no. 7, pp. 1084–1090, Jul. 2020.

[29] W. Sun, Z. Wang, and Q. Wang, "Hybrid event-, mechanism- and data-driven prediction of blast furnace gas generation," *Energy*, vol. 199, May 2020, Art. no. 117497.

[30] G. Yu, J. Sang, and Y. Sun, "Thermal energy diagnosis of boiler plant by computer image processing and neural network technology," *Thermal Sci.*, vol. 24, no. 5, pp. 3367–3374, 2020.

[31] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[32] R. Jiang, W. Tang, X. Wu, and W. Fu, "A random forest approach to the detection of epistatic interactions in case-control studies," *BMC Bioinf.*, vol. 10, no. S1, p. S65, Jan. 2009.

[33] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[34] G. M. Huebner, I. Hamilton, Z. Chalabi, D. Shipworth, and T. Oreszczyn, "Explaining domestic energy consumption—The comparative contribution of building factors, socio-demographics, behaviours and attitudes," *Appl. Energy*, vol. 159, pp. 589–600, Dec. 2015.

[35] J. C. P. Cheng and L. J. Ma, "A data-driven study of important climate factors on the achievement of LEED-EB credits," *Building Environ.*, vol. 90, pp. 232–244, Aug. 2015.

[36] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[37] C. F. Dormann, J. Elith, S. Bacher, C. Buchmann, G. Carl, G. Carré, J. R. G. Marquéz, B. Gruber, B. Lafourcade, P. J. Leitão, T. Münkemüller, C. McClean, P. E. Osborne, B. Reineking, B. Schröder, A. K. Skidmore, D. Zurell, and S. Lautenbach, "Collinearity: A review of methods to deal with it and a simulation study evaluating their performance," *Ecography*, vol. 36, no. 1, pp. 27–46, Jan. 2013.

[38] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

ADITYA PANKAJ SHAHA received the bachelor's degree in computer science and engineering from the Vellore Institute of Technology, Vellore. He is currently a Data Scientist with Grofers, an Indian groceries e-commerce company. He also works on optimizing the processes in supply-chain using data science. His research interests include interpretable machine learning and reinforcement learning.

MOHAN SAI SINGAMSETTI received the bachelor's degree in computer science and engineering from the Vellore Institute of Technology. He is currently a Visiting Researcher with the University of Alberta, working on combinational creativity approaches and neural architecture search methods. His research interests include machine learning, deep learning, AutoML, and combinational creativity.

B. K. TRIPATHY (Senior Member, IEEE) is currently the Dean of the SITE School, VIT, Vellore, India. He has published more than 550 technical papers in international journals, conference proceedings, and edited research volumes. He has supervised 48 candidates for research degrees, has published two books, six research volumes, monographs, and guest edited some research journals. He is a Senior Member of ACM, IRSS, and CSI. His current research interests include fuzzy sets and systems, rough set theory, data clustering, social network analysis, neighbourhood systems, soft sets, SIOT, big data analytics, multiset theory, decision support systems, DNN, and pattern recognition.

GAUTAM SRIVASTAVA (Senior Member, IEEE) received the B.Sc. degree from Briar Cliff University, USA, in 2004, and the M.Sc. and Ph.D. degrees from the University of Victoria, Victoria, BC, Canada, in 2006 and 2011, respectively. He then taught for three years at the Department of Computer Science, University of Victoria, where he was regarded as one of the top bachelor's professors with the Computer Science Course Instruction. Since 2014, he has been a tenure-track position at Brandon University, Brandon, MA, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank Associate Professor, in January 2018. He, as he is popularly known, is active in research in the field of data mining and big data. In his eight-year academic career, he has published a total of 60 articles in highimpact conferences in many countries and in high-status journals (SCI, SCIE) and has also delivered invited guest lectures on big data, cloud computing, the Internet of Things, and cryptography at many Taiwanese and Czech universities. He is an Editor of several international scientific research journals. He currently has active research projects with other academics in Taiwan, Singapore, Canada, Czech Republic, Poland, and USA. He is also constantly looking for collaboration opportunities with foreign professors and students. He received the Best Oral Presenter Award in FSDM 2017 which was held at the National Dong Hwa University (NDHU), Shoufeng, Hualien, Taiwan, in November 2017.

MUHAMMAD BILAL (Member, IEEE) received the B.Sc. degree in computer systems engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2008, the M.S. degree in computer engineering from Chosun University, Gwangju, South Korea, in 2012, and the Ph.D. degree in information and communication network engineering from the School of Electronics and Telecommunications Research Institute (ETRI), Korea University of Science and Technology, in 2017.

He was a Postdoctoral Research Fellow with the Smart Quantum Communication Center, Korea University, Seoul, South Korea, in 2017. He is currently an Assistant Professor with the Division of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin, South Korea. His research interests include design and analysis of network protocols, network architecture, network security, the IoT, named data networking, blockchain, cryptology, and future Internet.

Dr. Bilal has served as a Reviewer of various international journals, including the *IEEE Communications Magazine*, the IEEE SYSTEMS JOURNAL, IEEE ACCESS, the IEEE COMMUNICATIONS LETTERS, the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the *Journal of Network*, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING AND COMPUTER APPLICATIONS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Personal and Ubiquitous Computing*, and the *International Journal of Communication Systems*. He has also served as a Technical Program Committee Member on many international conferences, including the IEEE VTC, the IEEE ICMC AI-IOT, and the IEEE CCNC. He is an Editor of the IEEE Future Directions Ethics and Policy in Technology Newsletter and the IEEE Internet Policy Newsletter.

LEWIS NKENYEREYE received the Ph.D. degree in information security from Pukyong National University, Busan, South Korea. He was a Research Fellow with the Creative Human Resource Development Program for IT Convergence, Pusan National University. He was a Visiting Scholar with Thompson Rivers University, Kamloops, BC, Canada, and Georgia Southern University, Statesboro, GA, USA. He has served as a member of several technical program committees in various conferences and journals. He is currently an Assistant Professor in the area of computer and information security with the Department of Computer and Information Security, College of Electronics and Information Engineering, Sejong University, Seoul, South Korea. His research spans across the wide range of security and privacy related techniques with a particular interest in the Internet of Things (specifically the Internet of Vehicles). He is also involved in privacy-preserving techniques projects for BlockChain-based applications; Interoperability challenges in the IoT and M2M standards (with a special focus on one M2M).