A Real-Time Estimation Framework of Carbon Emissions in Steel Plants Based on Load Identification

Guolong Liu
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
E-mail: guolongliu@link.cuhk.edu.cn

Jinjie Liu
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
E-mail: jinjieliu@link.cuhk.edu.cn

Junhua Zhao*
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
E-mail: zhaojunhua@cuhk.edu.cn

Fushuan Wen
School of Electrical Engineering
Zhejiang University
Hangzhou, China
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
E-mail: fushuan.wen@gmail.com

Yusheng Xue
NARI Group Corporation (State Grid Electric Power Research Institute)
Nanjing, China
E-mail: xueysh@cae.cn

***Abstract*—In recent years, extreme weather disasters have occurred frequently, so reducing greenhouse gas emissions is an urgent task. How to accurately estimate corporate carbon emissions in real-time directly affects this task. Therefore, a real-time estimation framework of corporate carbon emissions based on load identification is proposed in this paper. In the proposed framework, the total carbon emission consists of two parts: direct carbon emission and indirect carbon emission. First, a Convolutional Neural Network - Bidirectional Long Short-Term Memory (CNN-BLSTM) model is presented and employed to monitor the states of devices in a factory in real-time. Then, the direct carbon emission is estimated according to the state and related carbon emission intensity. Meanwhile, the indirect carbon emission can be obtained through multiplying the marginal carbon emission factor by the electricity consumption in the factory. The proposed framework was used in a case study to estimate the carbon emission of a steel plant in real-time, which proved its effectiveness and accuracy.**

Keywords*—*Corporate carbon emission estimation, load identification, deep learning, carbon footprint.

I. Introduction

Climate change and carbon dioxide (CO_2) based greenhouse gas (GHG) emissions have become a major challenge for the world. To address the challenge, many GHG emission standards have been published, and scientific measurement of GHG is the basis for the effective implementation of these standards. As an important component of GHG, carbon emission has become the key monitoring object of various organizations. Meanwhile, corporates are increasingly concerned about their sustainable development. They hope to quantify their contributions to global climate change and take measures to control their carbon emissions through some carbon footprint projects [1].

Against this background, the concept of corporate carbon footprint (CCF) is proposed to measure corporate carbon emissions. Different from measuring carbon emissions by 'land area' (m^2, km^2, etc.), CCF refers to the total amount of carbon emissions caused by activities of a corporate, derived from the ecological footprint [2]. It does not require various assumptions, thus avoiding the increase in uncertainty. Based on CCF estimation, carbon auditing and carbon accounting can be conducted to measure the environmental performance of a corporate [2].

A lot of protocols and standards have been published to provide guidances on how to calculate carbon emissions, such as ISO 14067 [3], PAS 2050 [4], WBCSD/WRI 2004 [5]. Among them, WBCSD/WRI is the most widely used standard, which divides the carbon emission into three categories: direct emission, indirect emission caused by electricity, heat or cooling used, and indirect emission in upstream and downstream of the value chain. Currently, there is no a generally accepted method for carbon emission estimation [6]. The following three methods are mainly used to measure CCF in life-long cycle analysis (LCA) in existing studies: the bottom-up method based on process analysis (PA), top-down method based on environmental input-output (EIO) analysis, and a hybrid EIO-LCA method to combine the advantages of the aforementioned two methods [7].

The existing standards and methods have many limitations and shortcomings. Firstly, most of the current corporate carbon emission estimation methods are specially designed for a company or a type of companies based on LCA. They are not universal and cannot be directly applied to another different corporate or industry. For example, the CCF calculation is proposed for a wine production company in [8], for the cement industry in [9], for the manufacturing industry in [10], respectively. Secondly, there are few studies on device-level carbon emissions, and most of them are mainly for household devices rather than industrial devices [8]–[10]. Thirdly, the existing methods are for pre-evaluation or post-evaluation, not for real-time. Real-time carbon monitoring can help cope with the changing environment by providing information and give some suggestions to control the emission.

Aimed at addressing these deficiencies, a universal real-time corporate carbon footprint measurement framework is proposed in this paper. This work focuses on the mandatory

*Corresponding author
This work is supported by NARI Technology Co. Ltd (Technology Project: Study on Key Techniques of Integrated Energy System Simulation and Evaluation).

978-1-7281-8550-7/20/$31.00 ©2020 IEEE
DOI 10.1109/SGES51519.2020.00180

requirements set by the WBCSD/WRI protocol: the direct emission (scope 1) and indirect emission caused by the use of electricity (scope 2). For scope 1, a CNN-BLSTM based load identification method is proposed to monitor the states of devices in real-time, and then calculate the direct emission in combination with the device carbon emission intensities; for scope 2, a more accurate estimation method of carbon emissions in a given period based on the marginal carbon emission factor is used. The proposed framework can provide more general and accurate CCF measurements, which can not only measure the overall emissions of the entire corporate but also provide more refined device-level carbon emissions.

II. CNN-BLSTM Based Corporate Carbon Emission Estimation Framework

A. Marginal Carbon Emission Factor

For scope 2, some agencies and organizations have published guidelines for electricity emission factors in the form of spreadsheets or databases [11]–[14], based on the weighted average values of the electricity mix in different countries or regions [15]. The indirect emission caused by electricity used, that is, the carbon footprint of electricity is calculated as the product of average electricity emission factor and electricity consumption [16]. Evidently, there remain some errors due to the rough emission factors.

A more accurate factor named the marginal carbon emission factor based on the locational marginal price (LMP) is applied in a power system with market-based dispatch [17]. The marginal carbon emission factor of a node equals the carbon intensity of the marginal plant caused by unit load added on this node. By calculating the marginal carbon emission factor of the grid node where the corporate is located at, the indirect carbon emission from electricity consumption can be accurately measured in real-time [18].

B. Deep Neural Network

In recent years, deep neural networks have achieved remarkable actual applications in many aspects including visual object recognition, object detection, speech recognition, natural language processing, drug discovery and genomics [19]. Among them, the deep recurrent neural network (RNN) is very successful on the analysis of sequential data, such as text and time series, due to its structural advantages [20]. Since the RNN performs the same operations for each of its hidden states and each operation depends on the previous calculation result. It can be thought that the RNN remembers the information that has been calculated so far.

Unfortunately, during the process of training a deep RNN by using stochastic gradient descent, the calculated gradient will exponentially decay or amplify as it travels forward, which causes the memory of the RNN to be short-lived. In order to solve the long-term dependency problem of the RNN and avoid vanishing gradient, several methods have been proposed. The well-established long short-term memory network (LSTM) is designed to memorize long-term information [21]. In [22], the gradient is preserved by using second-order optimization methods to estimate its curvature. Informed random initialization is used for preserving the gradient in [23]. On the basis of LSTM, the Bidirectional Long Short-Term Memory (BLSTM) is proposed to elegantly make full use of context information in both forward and backward directions [24][25].

In previous work, BLSTM is superior to LSTM and RNN in many fields including sequence modeling [26], speech recognition [27][28], sentiment classification [29], and many others. BLSTM has the capability of capturing the timing relationships between data, and can attain good results in industrial load identification based on time series of system states such as voltage, current and power. Different from BLSTM's time series modeling superiority, the convolutional neural network (CNN) can effectively extract salient features from the original signals. Therefore, in order to simultaneously utilize the advantages of both, a joint CNN-BLSTM model is used for load identification.

C. One-dimensional Convolutional Neural Network (1D CNN)

Since CNN has the characteristics of parameter sharing and sparse connection, it can effectively learn key features from the original data. 1D CNN is usually used to process sequence data, and its key structure is the convolution layer. The convolution layer includes filters that can perform convolution operations on input feature and generate corresponding feature maps. The convolution operation is expressed as

$$m = \varphi(e \otimes k + b) \tag{1}$$

where m represents the generated feature map, e represents the input feature of the convolution layer, k represents the kernel of the convolution layer, b represents the bias, $\otimes$ represents the convolution operator, and φ represents the activation function that is usually the rectified linear unit (ReLU).

D. Bidirectional Long Short-Term Memory

From time step $t = 1$ to T, given the input sequence $x = (x_1, x_2, \dots, x_T)$, the hidden vector $h = (h_1, h_2, \dots, h_T)$ of a standard RNN can be calculated according to the following equation:

$$h_t = \sigma(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h) \tag{2}$$

where σ represents the activation function, W represents the weight matrix and b represents the bias vector. The output $y = (y_1, y_2, \dots, y_T)$ can be calculated through equation (3):

$$y_t = W_{hy} \cdot h_t + b_y \tag{3}$$

where W_{hy} represents the hidden-output weight matrix and b_y represents the output bias vector. As mentioned earlier, the RNN cannot remember long-term information. To solve this problem, the LSTM that contains memory blocks is proposed. A LSTM network is recursively composed of multiple identical memory units. For each memory block in the LSTM network, there are three gates, i.e., the input gate, forget gate and output gate. The input gate shown in equation (4) controls which information is entered into the memory block. The forget gate shown in equation (5) controls which information will be recorded. The output gate shown in equation (7) controls which information will be outputted to next cell state. Here i, f, c, o represents input gate, forget gate, cell state and output gate respectively.

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + W_{ci} \cdot c_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma\left(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + W_{cf} \cdot c_{t-1} + b_f\right) \tag{5}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \tag{6}$$

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + W_{co} \cdot c_t + b_o) \tag{7}$$

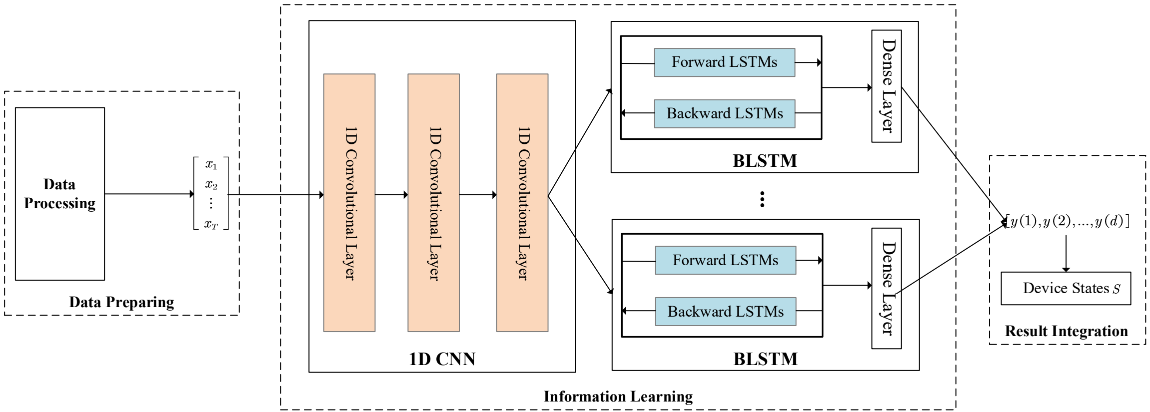

Fig.1. The framework of load identification method

$$h_t = o_t \cdot tanh(c_t) \tag{8}$$

It is worth noting that LSTM utilizes context information only in the forward direction and not in the backward direction, while BLSTM makes use of context information in both directions through recording forward information and backward information. Therefore, equation (8) cannot be used directly. Instead, the forward hidden sequence $\vec{h}$ and the backward hidden sequence $\overleftarrow{h}$ are defined for recording forward and backward information, the definitions of which are given by equation (9) and equation (10), respectively.

$$\vec{h} = \mathcal{H}\left(W_{x\vec{h}} \cdot x_t + W_{\vec{h}\vec{h}} \cdot \vec{h}_{t-1} + b_{\vec{h}}\right) \tag{9}$$

$$\overleftarrow{h} = \mathcal{H}\left(W_{x\overleftarrow{h}} \cdot x_t + W_{\overleftarrow{h}\overleftarrow{h}} \cdot \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{10}$$

So, the output of BLSTM is

$$y_t = W_{\vec{h}y} \cdot \vec{h} + W_{\overleftarrow{h}y} \cdot \overleftarrow{h} + b_y \tag{11}$$

E. Deep CNN-BLSTM based Load Identification

For load identification, compared with using intrusive load monitoring (ILM) methods, identifying industrial devices by using non-intrusive load monitoring (NILM) methods is the most practical way to realize. NILM methods utilize and analyze the data that collected by the smart meters which installed at the entry point of the factory to identify states of industrial devices. In this paper, a deep CNN-BLSTM based industrial load identification method is proposed. The framework of the proposed method is shown in Fig.1, where d is the number of devices in the factory.

The framework of the proposed industrial load identification method can be divided into three parts: data preparation, info rmation learning and results integration. Firstly, for each class of devices, a unique identification model is designed which takes the relevant load data from time $t = 1$ to T as the input. For information learning, three one-dimensional CNN layers are used to extract salient features from the original data, and the BLSTM-based identification model is used to extract forward and backward information from the feature map output by 1D CNN. Three BLSTM layers are used to learn the deep relationships of industrial load data. Lastly, each identification model will output the corresponding identified results of different devices. The results will be integrated to form a state vector that indicates the state of the device to be identified during the corresponding period x.

F. A Real-Time Estimation Framework for Corporate Carbon Emission

For device i, after identifying its state S_i, its direct carbon emission DE_i ($kgCO_2$) can be estimated through the carbon emission intensity DEI_{S_i} ($kgCO_2/kWh$) of device state S_i. Therefore, the total direct emission of the factory (scope 1) during the period x (hour) can be estimated by summing the direct emissions of all d devices, as shown in equation (12). α_x is the total direct carbon emission of the factory and d is the number of devices directly emitting carbon dioxide. DEI_{S_i} is the carbon emission intensity of device i at state S_i, which represents how many kilograms of carbon dioxide will be emitted by the device in this state for every kilowatt-hour of electricity consumed. It can be obtained through self-measurement or provided by the device manufacturer.

$$\alpha_x = \sum_{i=1}^{d} DEI_i * x \tag{12}$$

For the indirect emission of the factory caused by the use of electricity (scope 2) during the period x, it can be estimated using the marginal carbon emission factor of the node connected to the grid. The marginal carbon emission factor of the factory MEF_x ($kgCO_2/kWh$) can be calculated by solving the optimal power flow (OPF). Therefore, the indirect emission β_x can be obtained through multiplying the marginal carbon intensity MEF_x by the electricity consumption E_x of the factory during this period, as shown in equation (13). The total emission $Total_x$ of the factory in the period x can be obtained by summing the indirect emission and direct emission as shown in equation (14). The framework of corporate carbon estimation is shown in Fig.2.

$$\beta_x = MEF_x * E_x \tag{13}$$

$$Total_x = \alpha_x + \beta_x = \sum_{i=1}^{d} DEI_i * x + MEF_x * E_x \tag{14}$$

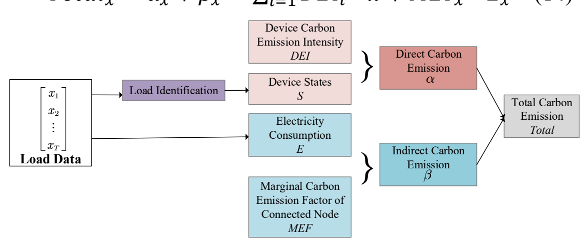

Fig.2. The framework of corporate carbon estimation

III. Case Study and Discussions

A. Data Description

In the experiment, load data with a sampling frequency of 1 Hz and a duration of 30 days are recorded from a steel plant. Four devices need to be identified, i.e. the Grinder, Electric Stove I, Electric Stove II and Filter. Since the working modes of different types of devices are very different, the working time and cycle are also different. Therefore, separate learning inferences are required for each type of devices to provide more accurate identification results. In this paper, individual identification models are provided for four devices that together form the information learning part of the entire identification framework.

B. Data Preparation for Load Identification

In the original data, there are several problems such as missing data, duplicate sampling values and excessive large numerical range. The data cannot be directly used and needs to be preprocessed. First, for missing data, the Mean Completer algorithm is used to fill the corresponding missing values. Then, duplicate sample values are detected and removed based on timestamps. Further, since it is difficult for deep neural networks to learn from data with an excessively large numerical range, the data in the dataset are scaled to between 0 and 1 by applying the min-max normalization [30]. The new values x^{new} of the attribute x can be derived through equation (15):

$$x^{new} = \frac{x - min(x)}{max(x) - min(x)} \tag{15}$$

where $min(x)$ and $max(x)$ are the minimum and maximum values of x, respectively.

In this paper, the work of load identification is composed of multiple multiple-classification models. As an important feature of supervised learning, the classification models require the given input $X = \{x_1, x_2, \dots, x_T\}$ and the corresponding label $Y = \{y_1, y_2, \dots, y_T\}$. Therefore, the load data as a time series is divided into the sub-sequence set with a length of five minutes as input X. The working state of devices is divided into state 1, state 2, …, state n, and is encoded with 0 ,1…, n, respectively. The working state of d devices constitutes the state vector $y_t = \{y_t(1), y_t(2), \dots, y_t(d)\}$ at the corresponding time t. After the above processing, 75% of the data is used for model training, 5% is used for model validation, and the remaining 20% is used for model testing.

C. Metrics and Benchmarks for Load Identification

Since this task is a classification problem, the accuracy for evaluating classification performance is mainly considered. For device i, the accuracy $Accuracy_i$ is defined in equation (16) where CI_i is the number of states that are correctly identified and TS_i is the number of states. For all devices in the factory, the accuracy $Accuracy_{total}$ is defined in equation (17).

$$Accuracy_i = \frac{CI_i}{TS_i} \tag{16}$$

$$Accuracy_{total} = \frac{1}{n}\sum_{i=1}^{d} Accuracy_i \tag{17}$$

For NILM, there are currently some existing methods, so they are chosen as benchmarks for comparison, including the k-Nearest Neighbor (kNN), Support Vector Machine (SVM), Decision Tree, Multiple-Layer Feed-Forward Artificial Neural Network with the Backpropagation training algorithm (BP-ANN) and CNN [31].

D. Experiments Setup for Load Identification

In the experiments, an individual identification model is used for each device to identify its working states. It is well known that although deep neural networks have powerful extracting and learning capabilities, parameters such as the numbers of layers and hidden units are very difficult to determine. For the task in this work, the training time of the network needs to be as short as possible while ensuring the accuracy. Therefore, there is a trade-off between the training time and accuracy. The validation part of the dataset is used for guiding the optimal parameters. Through experiments, the optimal combination of parameters is determined. For 1D CNN, there are three 1D convolution layers with 32 convolution kernels for extracting salient features from the original data. The kernel size is 16, and the stride length is 1. For BLSTM, there are three BLSTM layers with 256 hidden states used to learn the internal relationships in the data. After BLSTM layers, there is a fully connected layer with the number of nodes equal to the number of device states to output the device state. Overfitting is a particularly prone problem in deep neural networks, so the dropout technique is used in the training of the model [32]. The dropout rate is set to 0.2, which means that 20% connections between hidden layers and within hidden layers will be randomly broken to avoid overfitting. The optimization algorithm is Adam [33] and categorical cross entropy is used as the loss function. The mini-batch size is set at 32. The experiments are implemented in Keras 2.4.3 with Tensorflow 2.3.0 as backend and executed on a GPU cluster with Ubuntu Server 18.04 64-bits, 4 GTX-1070Ti, a 16-core CPU and 32 Gigabytes RAM.

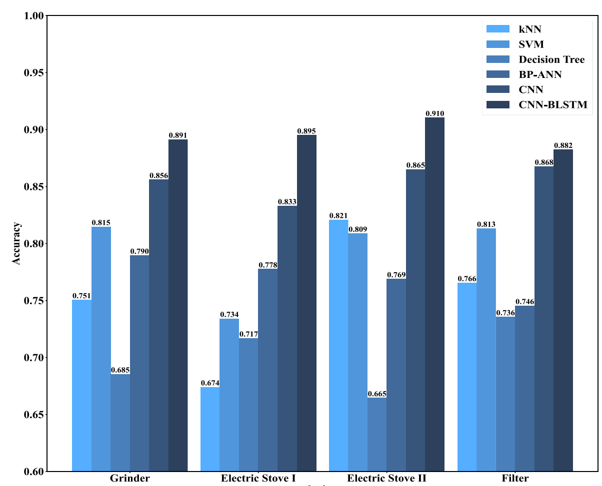

Fig.3. Accuracy of different methods

E. Results and Analysis for Load Identification

The results of the experiments are shown in Fig.3. It can be clearly seen from the figure that the method proposed in this paper outperforms the five existing methods, which can achieve more accurate load state monitoring. The proposed method realizes an overall accuracy close to 90%. Fig.4 shows the identification accuracy of the different methods on four devices. It can be seen that the CNN-BLSTM based identification method has the highest median and maximum and minimum values. The accuracy distribution range is also smaller than the existing methods, which indicates that the proposed method is superior to five benchmarks in accuracy and stability.

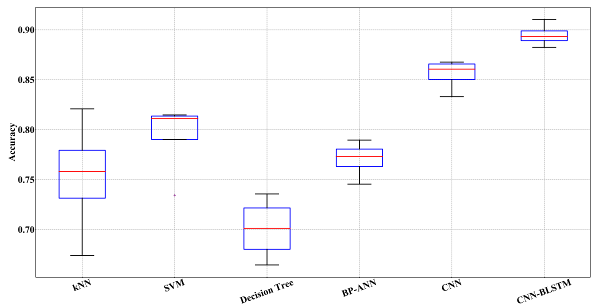

Fig.4. The boxplot of accuracy with different devices

F. Experiments for Carbon Emission Estimation of a Steel Mill

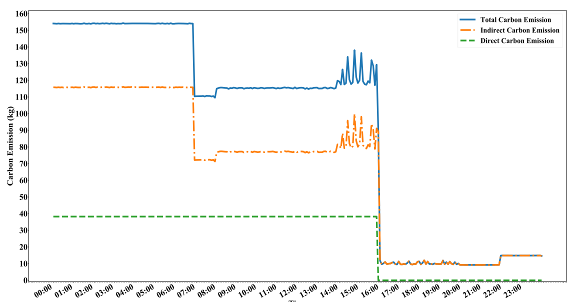

Fig.5. The carbon emission curve of a steel mill in one day

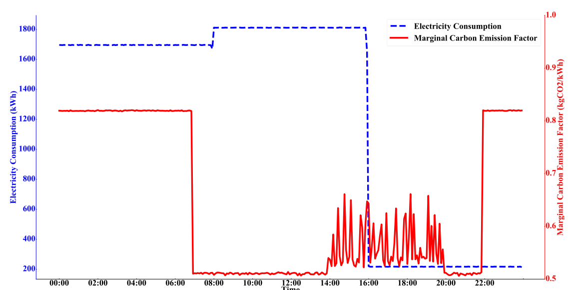

Fig.6. The electricity consumption and marginal carbon emission factor of a steel mill in one day

The time interval for load identification is five minutes, and the calculation time interval for the carbon emission is consistent with it. Therefore, the carbon emission of the steel mill is calculated every five minutes. In a calculation cycle, after obtaining the device states identified by load identification, the direct emission of the steel mill is calculated according to the carbon emission intensity in the corresponding states. Then calculate the indirect emission based on the electricity consumption in the period and the corresponding marginal carbon emission factor. In the experiment, the steel mill is located at a node of IEEE 118. The estimated carbon emission curve is shown in Fig.5. It can be seen that when the device that directly emits carbon dioxide is turned off at around 16:00, the direct emission is reduced to zero, the indirect emission and the total emission is also reduced accordingly. As can be seen from Fig.5 and Fig.6, the indirect emission and total emission drop rapidly after six o'clock in the morning, while the direct emission and electricity consumption remain unchanged. The reason is the rapid decline of the marginal carbon emission factor, resulting in a rapid decline in the indirect emission. The experiments have proved that the corporate carbon emission can be estimated in real-time by using the proposed framework, while the indirect emission and direct emission are clearly and accurately calculated. This achieves real-time monitoring of corporate carbon emission without the need to deploy additional sensors and communication networks, and greatly improves the accuracy of monitoring.

IV. CONCLUSIONS

In this paper, the real-time estimation framework of corporate carbon emission based on load identification is studied. A CNN-BLSTM based method is proposed for processing industrial load data, and monitoring device states. The proposed framework first processes the raw data into standard input data of the models through the data preparation part. The obtained standard data are then inputted into the CNN-BLSTM based load identification models for information learning and inference. The result integration part forms the outputs into a corresponding state vector of devices. Then the device states are calculated together with the device emission intensity to obtain the amount of direct emission. Meanwhile, the indirect carbon emission is obtained through multiplying the electricity consumption by the marginal carbon emission factor. Finally, the direct and indirect emission amounts are summed to get the total emission. Experimental results show that the identification accuracy of the proposed load identification method is close to 90%, which is significantly better than the five benchmarks. The proposed carbon emission estimation framework based on load identification can accurately estimate the direct and indirect carbon emission of a corporate in real-time. Through this framework, it is possible to better assess the sustainable development capabilities of a corporate, quantify its environmental impact, and help take measures to mitigate climate change.

REFERENCES

[1] S. Schaltegger, D. Zvezdov, I. A. Etxeberria, M. Csutora, and E. Günther, *Corporate carbon and climate accounting*. Springer, 2015.

[2] T. Wiedmann and J. Minx, "A definition of 'carbon footprint,'" *Ecological economics research trends*, vol. 1, pp. 1–11, 2008.

[3] I. S. O. 14067, "Greenhouse gases – Carbon footprint of products – requirements and guidelines for quantification and communication," *International Organization for Standardization, Geneva*, 2013.

[4] P. A. S. 2050, "Specification for the assessment of the life cycle greenhouse gas emissions of goods and services," *British Standards Institution*, 2011.

[5] W. R. I. Wbcsd and others, "The greenhouse gas protocol. A corporate accounting and reporting standard," *World Business Council for Sustainable Development and World Resources Institute: Geneva, Switzerland*, p. 116, 2004.

[6] A. L. Radu, M. A. Scrieciu, and D. M. Caracota, "Carbon footprint analysis: towards a projects evaluation model for promoting sustainable development," *Procedia Economics and Finance*, vol. 6, pp. 353–363, 2013.

[7] M. Csutora and G. Harangozo, “Twenty years of carbon accounting and auditing--a review and outlook,” *Society and Economy*, vol. 39, no. 4, pp. 459–480, 2017.

[8] A. P. Fernandes *et al.*, “Emission factors from residential combustion appliances burning Portuguese biomass fuels,” *Journal of Environmental Monitoring*, vol. 13, no. 11, pp. 3196–3206, 2011.

[9] S. Ozgen *et al.*, “Emission factors from small scale appliances burning wood and pellets,” *Atmospheric environment*, vol. 94, pp. 144–153, 2014.

[10] T. Cáceres, H. Soto, E. Lissi, and R. Cisternas, “Indoor house pollution: Appliance emissions and indoor ambient concentrations,” *Atmospheric Environment (1967)*, vol. 17, no. 5, pp. 1009–1013, 1983.

[11] D. DECC, “Guidelines to Defra/DECC’s GHG conversion factors for company reporting,” *London: Department for Environment, Food and Rural Affairs and Department for Energy and Climate Change*, 2011.

[12] EPA, “Emissions & Generation Resource Integrated Database (eGRID 2018).”

[13] IEA, “Statistics by country/region for electricity/heat.”

[14] WRI(2011), “GHG Protocol Tool for Emissions from Purchased Electricity – version 4.2,” 2011.

[15] C. Szigeti and G. Harangozó, “Online carbon calculators-Corporate carbon footprint analysis in practice,” *Proceedings of MAC-MME 2016*, p. 262, 2016.

[16] M. Brander, A. Sood, C. Wylie, A. Haughton, and J. Lovell, “Electricity-specific emission factors for grid electricity. Ecometrica,” *Edinburgh, United Kingdom*, 2011.

[17] M. M. Rogers, Y. Wang, C. Wang, S. P. McElmurry, and C. J. Miller, “Evaluation of a rapid LMP-based approach for calculating marginal unit emissions,” *Applied energy*, vol. 111, pp. 812–820, 2013.

[18] K. Siler-Evans, I. L. Azevedo, and M. G. Morgan, “Marginal emissions factors for the US electricity system,” *Environmental science & technology*, vol. 46, no. 9, pp. 4742–4748, 2012.

[19] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[20] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp. 6645–6649.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] J. Martens and I. Sutskever, “Learning recurrent neural networks with hessian-free optimization,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1033–1040.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *International conference on machine learning*, 2013, pp. 1139–1147.

[24] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural networks*, vol. 18, no. 5–6, pp. 602–610, 2005.

[25] S. Zhang, D. Zheng, X. Hu, and M. Yang, “Bidirectional long short-term memory networks for relation classification,” in *Proceedings of the 29th Pacific Asia conference on language, information and computation*, 2015, pp. 73–78.

[26] F. A. Gers and E. Schmidhuber, “LSTM recurrent networks learn simple context-free and context-sensitive languages,” *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[27] A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional LSTM,” in *2013 IEEE workshop on automatic speech recognition and understanding*, 2013, pp. 273–278.

[28] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Fifteenth annual conference of the international speech communication association*, 2014.

[29] P. Chen, Z. Sun, L. Bing, and W. Yang, “Recurrent attention network on memory for aspect sentiment analysis,” in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 452–461.

[30] Y. K. Jain and S. K. Bhandare, “Min max normalization based data perturbation method for privacy protection,” *International Journal of Computer & Communication Technology*, vol. 2, no. 8, pp. 45–50, 2011.

[31] G. Liu, J. Gu, J. Zhao, F. Wen, and G. Liang, “Super Resolution Perception for Smart Meter Data,” *Information Sciences*, 2020.

[32] G. Hinton, “Dropout : A Simple Way to Prevent Neural Networks from Overfitting,” vol. 15, pp. 1929–1958, 2014.

[33] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.