Detection of Malfunctioning Smart Electricity Meter

Ming Liu\textsuperscript{a,b,c}, Dongpeng Liu\textsuperscript{c}, Guangyu Sun\textsuperscript{c}, Yi Zhao\textsuperscript{c}, Duolin Wang\textsuperscript{c}, Fangxing Liu\textsuperscript{c}, Xiang Fang\textsuperscript{d}, Qing He\textsuperscript{e}, Dong Xu\textsuperscript{c}\textsuperscript{*}

\textsuperscript{a}Key Laboratory for Applied Statistics of MOE and School of Mathematics and Statistics, Northeast Normal University, Changchun, 130024, China.
\textsuperscript{b}Department of Applied Mathematics, School of Mathematics and Statistics Sciences, Changchun University of Technology, Changchun, 130012, China.
\textsuperscript{c}Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, Missouri 65211, USA.
\textsuperscript{d}Qinghu Rising Sunshine Data Technology (Beijing) Co., Ltd, Beijing 100084, China;
\textsuperscript{e}Division of Electromagnetic Metrology, National Institute Ltd, Beijing 100084, China;

Abstract

In this paper, a method for malfunctioning smart meter detection, based on Long Short-Term Memory (LSTM) and Temporal Phase Convolutional Neural Network (TPCNN), is proposed originally. This method is very useful for some developing countries where smart meters have not been popularized but in high demand. In addition, it is a new topic that people try to increase the service life span of smart meters to prevent unnecessary waste by detecting malfunctioning meters. We are the first people complete a combination of malfunctioning meters detection and prediction model based on deep learning methods. To the best our knowledge, our approach is the first method that achieves the malfunctioning meter detection of specific residential areas with their residents’ data in practice. The procedure proposed creatively in this paper mainly consists of four components: data collecting and cleaning, prediction about electricity consumption based on LSTM, sliding window detection, and single user classification based on CNN. To make better classifying of malfunctioned user meters, we combine recurrence plots as image-input and combine them with sequence-input, which is the first work that applies one and two dimensions as two paths CNN’s input for sequence data classification. Finally, many classical methods are compared with the method proposed in this paper. After comparison with classical methods, Elastic Net and Gradient Boosting Regression, the result shows that our method has higher accuracy. The average area under the Receiver Operating Characteristic (ROC) curve is 0.80 and the standard deviation is 0.04. The average area under the Precision-Recall Curve (PRC) is 0.84.

Keywords: Type your keywords here, separated by semicolons;

\textsuperscript{*}Corresponding author: Dong Xu. Tel.:+1-573-8822299; e-mail: xudong@missouri.edu
1. Introduction

With the wide installation of smart meters, the electricity information collection system being more and more popular. The communication between power company and users is strengthened gradually. Smart meter is one of the most important electrical metering equipment in the construction of a smart grid, because the reliability of smart meters directly affects the stability and safety of smart grid. According to the latest report of Navigant Research, China has a high share in global smart meter market in the first season of 2018, with more than 496 million smart meters installed, occupying 68.4% of the world’s share. There will be almost 2 billion smart meters installed by 2020, and the smart grid will cover nearly 80% population in the world [1]. According to regulations, the general service life span of smart meters is 8 years in China, but most of them are still in good condition after having used for 8 years. If we can merely replace the malfunctioning smart meters, a huge amount of money and human resources would be saved. For example, if the smart meters increase average service life span into 12 years because of this system, it can save about 12 million dollars per year (assuming each smart meter is 60 dollars). Besides, there are hundreds of suppliers of different scales working for national grid, most of them have different production processes, making smart meters with various service life spans. However, checking and replacing all users’ smart meters in a constant time period is a labor-intensive task. Therefore, it is more reasonable to design and use an intelligence system, rather than human, to find malfunctioning smart meters.

Smart-meter data analysis means using statistical analysis methods to collect raw data for processing, modeling and calculating to extract useful information and get conclusions. The commonly used statistical analysis methods related to smart meters can be summarized as follows: correlation analysis, cluster analysis, malfunction analysis and tendency analysis. Among these, malfunction analysis plays a major role in equipment failure and abnormal power consumption (such as power theft) diagnosis. This paper mainly focuses on malfunction analysis, which refers to an analysis method that traces the cause of anomalies that deviate from the general electricity usage behavior.

Since the popularization of smart meters, massive data about the users are collected every moment. The excessive data gives related researchers a huge possibility to improve the accuracy of prediction, but at the same time, it also brings an excessive amount of computational effort. Silipo et al. [2] use K-Means algorithm and Auto-Regressive model to predict the usage of energy in the future. However, their approach does not contain a detection model since they do not have records of meter meters. In addition, they do not use deep learning method, which makes their result not accurate enough. The data set that Silipo et al used is gathered from Customer Behavior Trials (CBTs) conducted by the Irish Commission for Energy Regulation (CER) in 2007. Similarly, Cosmo et al [3] use the same data set with Silipo. They are trying to find out the relationship between electricity consumption and time-of-use tariffs by studying the historical data. Wu et al. [4] implement and integrate CUSUM and Shewhart methods to identify smart meter malfunction. Their dataset only contains summer season, but the electricity usages usually have differences in different seasons. Hence, their method may not perform well in a long-term dataset.

The deep learning methods are often used to do malfunction detection. Gajowniczek et al. [30] proposed an approach to predict the electricity load. Their approach can be used for 24 hours ahead electricity load prediction. However, their approach is designed for short-term forecasts of individual users. Nikovski et al. [5] proposed a predictive model for technical losses in electrical distribution networks to detect power theft. However, the data sampling rate is not high enough to detect power theft definitely. Zhang et al. [6] employed a deep learning method to detect traffic accidents from social media data. They investigated two deep learning methods—Deep Belief Network (DBN) and LSTM—to find the relationship inherent in the accident-related social media data. Kim and Cho [7] proposed a C-LSTM method combining Long Short-Term Memory, convolutional neural network, and deep neural network to extract complex features. Their C-LSTM method also achieves abnormal detection for web traffic data. Zafferey et al. [8] introduced the how to use Artificial Neural Networks to do short-term load and Photovoltaic (PV) predictions of smart meter profiles.

To find out the smart meter's malfunctioning type, one way is collecting historical data before the malfunction and then sampling and modeling it to
predict malfunction of smart meters. By analyzing malfunction of electricity big data, Sheng et al. [9] came up with a big data modeling method for power system identifying abnormal data based on matrix theory. He [10] analyzed smart meters’ history abnormal data to study the malfunction occurrence probability and malfunctioning type of smart meter, then explored how to store the data reasonably and used neural networks to predict the malfunction of the smart meters, but the accuracy was not good enough to put into commercial use. Yang et al. [11] proposed a smart meter malfunction identification model based on abnormal analysis model of smart meter measurement. They used the malfunction prediction model to predict whether the measuring process has a non-measurement malfunction or not. Yao et al. [12] introduced the difficulties of smart meter maintenance and faults classification. They also proposed three reliability prediction methods for smart meter life based on big data and their advantages and disadvantages briefly. However, they have not delved deeper to get a complete method of predicting the service life of smart meters like this paper. At present, the problem of smart meters malfunction diagnosis based on deep learning is still in a relatively initial stage, especially when combined with the problem of smart meters malfunction detection. This paper is arranged as follows: Section 2 introduces data collecting method, data cleaning method, and detection and diagnosis strategy, where we use LSTM and CNN methods to build malfunctioning smart meter prediction and detection models. Section 3 describes and analyzes the experiments and results. We achieve the combination of smart meter detection and prediction model. Finally, Section 4 reveals the conclusion.
Fig. 1. Working process of detection and diagnosis
2. Method

2.1. Overview of the method

Our method contains four parts: Data Collecting, Malfunction detection, Diagnosis and Result. The work process is shown like Fig. 1.

Current, voltage and energy usage of every single user are recorded by submeters, while the whole residential area’s average current, voltage and energy usage are recorded by the master meter. Since the master meter is unique and important, it is under careful maintenance. Therefore, we can assume that it is always accurate. Based on this assumption, the system is described in Fig. 1. Our system collects these data from submeters and the master meter respectively. In Malfunction detection section, our system uses LSTM method to predict the reading of the Master meter based on the data collected from Submeters. The system compares the predicted value with the actual Master meter reading. If the predicted value is quite different from actual Master meter reading for a period of time, the detection system will consider there are malfunctioning meters in the residential area, and the Diagnosis part will begin to work. The Diagnosis system will utilize CNN model to classify every single user into abnormal user or normal user. At last, the system will find the malfunctioning meters.

2.2. Data Collecting and Brief Analyze

Firstly, the data our system uses is collected by smart meters from two residential areas. The smart meter being studied has five parameters: rated power of 1100 W, rated voltage of 220V, rated current of 5 A, rated frequency of 50Hz and error rate of 2%. There are two kinds of smart meters: master meter and submeter. The submeters which are installed for all users can collect data from them and send the collected data to the master meter, while the master meter can record the data of the entire residential area. There is only one master meter in a residential area so that it is under careful maintaining and monitoring, we could assume its measurement is always accurate. Malfunction detection for a residential area is accomplished by comparing the error between master meter and sum of sub meters. When the estimated error is extremely high in a residential area, one or more sub meters in that residential area would have failed. Then, our model will find malfunctioning meters among meters in that residential area using classification. Therefore, our method is divided into two parts: detection and classification, as described in Eq. 1.

Silipo et al. [2] aim to predict electricity usage for different groups of meter IDs. They firstly import all the information for every user and clean it. They define some behavioral measures of electricity usage that could cause influence on the calculation. Then, they cluster all users according to the behavioral measures, which reduces the analysis space from 6000 meters to at most 30 clusters. Finally, they create predictive models for every cluster.

There are two difference between their data set and ours. First, their data set does not record instantaneous current and voltage value. Instead, their data set collects the questionnaire completed by users. Second, unlike their data set, our data set not only records smart meter (submeters) for each user but also smart meters (master meters) for specific residential area. Therefore, their approach is not suitable for specific residential areas. Based on that, our project does the similar data processing except we do not need cluster the users and we also create malfunction detection model because of the existence of master meter. Our system does not need to analyze all users’ submeters if the detection model shows that the result is normal.

After concatenating all the data and renaming the columns, we get tables like Table 1 (a), (b) and Table 2. RID, UID, SID in Table 1 (a) and (b) stand for Residential area ID, User ID and System Id respectively. The master meter has three phases and submeters only have one phase. Magnification means how many times the apparent size is enlarged. The meter’s magnification is 160 and the submeter’s is 1. For a single user, three values are collected: voltage, current and electricity usage. Therefore, the data set is given by three tables. The original voltage and current table have 102 columns and 3,038 rows.

Voltages and current are collected every 15 minutes in the master meter and collected every hour in submeters. Meanwhile, columns of everyday usage table are shown in Table 2. The original usage table have 6 columns and 65534 rows.

Everyday voltage and current of every user are recorded in Table 1 and everyday electricity usage is
recorded in Table 2. Since each user has a unique user ID and system ID, [User ID, Date] and [System ID, Date] are candidate keys, and finally [User ID, Date] is chosen as the primary key of this data set.

2.3. Data Cleaning and Preprocessing

Fill in the missing values

The submeters do not record values every 15 minutes as the master meter does. In order to format the data, the missing values are replaced by the values recorded every hour.

Clean the invalid values

There are three kinds of invalid data in our data sets: redundant data, unscientific data and non-numeric characters. For a single ID, the data with same date time which appears multiple times are redundant data. We keep one row of them and delete the others. Unscientific data includes the data whose sum of submeters is more than the master meter. According to technical specifications of the smart meters, master meter should always larger than the sum of submeters due to the loss on the circuit. During cleaning, all non-numeric characters are translated into string for the convenience.

Features encoding

In order to explore the weekday, month and year’s influence on electricity usage, date is converted into three kinds of formats in 22 dimensions: The week, month and year are presented in one-hot encoding. Considering our prediction is time-based, this process can provide more features to the prediction system.

2.4. Data Analysis

The desensitized data we used are collected from two residential areas called Hua Yuan (residential area A) and Dong Hui (residential area B). Residential area A and B collect the voltage readings and the electric current readings of every hour from August 2014 to August 2016. In addition, the master meter of residential area A records voltage and current every 15 minutes.

After data cleaning, the formula used to calculate the measurement error is defined as follow equation:

\[ E = W_{\text{sub}} - \sum_{i=1}^{n} W_{\text{sub}_i} \]  \hspace{1cm} (1)

where \( E \) represents residue error between master meter and sum of the submeters, \( W_{\text{master}} \) means usage of master meter and \( W_{\text{sub}} \) means the daily usage of submeters. The errors between master meter and submeters are small. Most of the relative errors do not exceed 0.02, which reveals the high accuracy of collected data. Considering the massive data, the data is considered acceptable.

To detect the measurement error, we draw and visualize the error distribution. Error distribution of normal users roughly follows normal distribution. If the distributions are similar, we can try to introduce abnormal data into it and to detect malfunction by observing if the distribution will change. However, the distributions are changed as time, which is shown on Fig.2. Therefore, we can’t use above method to solve it directly.

2.5. Detection and Diagnosis Strategy

The main objective of this work is to design an efficient and accurate process, based on LSTM and CNN, to detect the malfunction in the smart meters and diagnose which smart meter is the abnormal one.

Deep Neural Network (DNN) is an artificial neural network with many hidden layers between the input and output layers[13]. Recurrent Neural Network (RNN) is a variant of DNN that is suitable for sequence data. RNN is a very efficient sequence model, but sometimes it is difficult to train because of
exploding and vanishing gradients. They are two common problems of RNN because of its iterative nature [14]. The exploding gradient problem can be solved by enforcing a hard constraint over the norm of the gradient, while the vanishing gradient problem can be overcome by the LSTM [15].

Table 1(a) Data set of current (Master meter)

| RID | UID  | SID | Magnification | Date       | Phase | 0:00 | 0:15 | …   | 23:30 | 23:45 |
|-----|------|-----|---------------|------------|-------|------|------|-----|-------|-------|
| 01365 | 24663 | 20544 | 160          | 8/4/2014   | A     | 0.407 | 0.328| …   | 0.467 | 0.308 |
| 01365 | 24663 | 20544 | 160          | 8/4/2014   | B     | 0.603 | 0.581| …   | 0.818 | 0.681 |
| 01365 | 24663 | 20544 | 160          | 8/4/2014   | C     | 0.62  | 0.629| …   | 0.839 | 0.811 |
| 01365 | 24663 | 20544 | 160          | 8/5/2014   | A     | 0.335 | 0.365| …   | 0.506 | 0.578 |
| 01365 | 24663 | 20544 | 160          | 8/5/2014   | B     | 0.597 | 0.727| …   | 0.557 | 0.642 |
| 01365 | 24663 | 20544 | 160          | 8/5/2014   | C     | 0.604 | 0.618| …   | 0.898 | 0.939 |

Table 1(b) Data set of current (Submeter)

| RID | UID  | SID | Magnification | Date       | Phase | 1:00 | 2:00 | …   | 22:00 | 23:00 |
|-----|------|-----|---------------|------------|-------|------|------|-----|-------|-------|
| 01365 | 16407 | 18897 | 1        | 8/1/2016   | A     | 4.082 | 0.132| …   | 1.329 | 1.453 |
| 01365 | 16414 | 24942 | 1        | 8/1/2016   | A     | 3.33  | 1.582| …   | 4.809 | 2.87  |
| 01365 | 16430 | 18903 | 1        | 8/1/2016   | A     | 5.733 | 0.12 | …   | 6.038 | 5.519 |
| 01365 | 16407 | 18897 | 1        | 8/2/2016   | A     | 4.012 | 1.041| …   | 1.039 | 0.768 |
| 01365 | 16414 | 24942 | 1        | 8/2/2016   | A     | 2.687 | 2.703| …   | 3.662 | 2.273 |
| 01365 | 16430 | 18903 | 1        | 8/2/2016   | A     | 4.865 | 4.673| …   | 5.152 | 1.178 |

Table 2 Data set of usage (Master meter and submeter)

| Residential ID | User ID | System ID | Date       | Usage |
|---------------|---------|-----------|------------|-------|
| 01365         | 24663   | 20544     | 8/3/2014   | 1110.4|
| 01365         | 16476   | 17061     | 8/3/2014   | 6.32  |
| 01365         | 16465   | 18778     | 8/3/2014   | 9.99  |
| 01365         | 16469   | 18779     | 8/3/2014   | 6.24  |
| 01365         | 16443   | 18783     | 8/3/2014   | 5.03  |
| 01365         | 16450   | 18784     | 8/3/2014   | 6.58  |
LSTM is a specific RNN architecture that is much easier to train. LSTM is well-suited to process and predict long time-series data. Hochreiter and Schmidhuber [16] proved that LSTM is more effective than other recurrent neural networks when the input sequences were long. The main idea behind the LSTM architecture is a memory cell, which can maintain its state over time, and nonlinear gating units, which regulate the information flow into and out of the cell [17].

CNN is a class of Feedforward Neural Network that has been widely used in image recognition and sequence task like natural language processing [18]. The layers of CNN have neurons with three dimensions. CNN can use spatial locality between neurons of adjacent layers, which produces the strongest response to a spatially local pattern. Because of shared weight, CNN generally has a lower number of parameters than DNN and many other neural networks. Therefore, it can save the memory to train more powerful networks.

2.5.1. Master Meter’s Error Prediction Based on LSTM

In order to accomplish the prediction of time series, we use a two-layer LSTM to get the predicted value. The input is the history error and date presented in several forms. The output is the predicted error for expected date in the future. The architecture of LSTM is described as Fig. 1. Both LSTM-1 and LSTM-2 have 30 dimensions. Prediction is calculated after a dense layer. The details are described as following:

LSTM calculates a hidden state $h_t$ in following formulas:

$$f_t = \sigma(x_t U_f^f + h_{t-1} W_f^f)$$  \hspace{1cm} (2)

$$o_t = \sigma(x_t U_o^o + h_{t-1} W_o^o)$$  \hspace{1cm} (3)

$$\tilde{C}_t = \tanh(x_t U_{\tilde{C}} + h_{t-1} W_{\tilde{C}})$$  \hspace{1cm} (4)

$$C_t = o_t \cdot \tilde{C}_t + i_t \cdot C_{t-1}$$  \hspace{1cm} (5)

$$h_t = \tanh(C_t) \cdot o_t$$  \hspace{1cm} (6)

where: $i, f, o$ are called the input, forget and output gates respectively. $W$ is the recurrent connection between the previous hidden layer and the current hidden layer. $U$ is the weight matrix that connecting the inputs to the current hidden layer. $\tilde{C}$ is a “candidate” hidden state that is computed based on the current input and the previous hidden state. $C$ is the internal memory of the unit [19].

The inputs are shown in Table 3. The first dimension represents the batch size and the second dimension represents the number of time steps, in other words the sequence length. The third dimension represents the number of cells in an input sequence. We build our sequence training dataset by stride=1. As a result, each sample gives the features of the days in the window. We have 730 samples totally and each sample has rows number equaling to sequence length. Finally, we split 703 samples as the train set and 27 samples as the test set. Meanwhile, we compare different sequence length and select the most efficient one in Section 3.2.

Table 3 Description of inputs

| Features      | Notes                                      |
|---------------|--------------------------------------------|
| error: float  | Error between sum of the usage of submeters and usage of master meter |
| master: float | Usage of master meter                      |
| com_date: int | Length between current date and base date after setting a base date as 0 |
| week: list    | 7 dimensions one-hot code                  |
| month: list   | 12 dimensions one-hot code                 |
| year: list    | 3 dimensions one-hot code                  |
| numbers: int  | Number of users in current residential area |

2.5.2. The Classification of Single-user’s Smart Meters Based on CNN

According to the data analysis, finding the malfunctioning meters is a classification problem. An efficient solution to classification problem is CNN. CNN considers the partial features and the data continuity, so it has high accuracy in image classification as mentioned before. Therefore, in order to make full use of data and show the data characteristics better, we convert the everyday usage data of every user into recurrence plots (RP).

Table 4 Architecture of CNN for series inputs

| Layer   | Parameters                        |
|---------|-----------------------------------|
| Conv1D  | Filters numbers: 32, kernel size: 5, Weight regularizer: 11 |
Batch Normalization -

**Activation** function="relu"

**Conv1D** Filters numbers: 32, kernel size: 3, Weight regularizer: l1

**Activation** function="relu"

**Flatten** -

**Dense** Units: 64

**Batch Normalization** -

**Dense** Units: 32

**Dense** Units: 1

| **Batch Normalization** | - |
|------------------------|---|
| **Activation**        | function="relu" |
| **Conv1D**             | Filters numbers: 32, kernel size: 3, Weight regularizer: l1 |
| **Activation**        | function="relu" |
| **Flatten**            | - |
| **Dense**              | Units: 64 |
| **Batch Normalization**| - |
| **Dense**              | Units: 32 |
| **Dense**              | Units: 1 |

Related work by Nima Hatami [20] employed 2 dimensional (2D) CNN for RP plot classification. Instead using their architecture, our 2D CNN model is based on transfer learning [21] from VGG16, pretrained by ImageNet dataset. We add a Dense layer in the end to get one-dimensional output. On top of it, we compare the result of series as the input and image as the input, using the. The 1D CNN architecture we use is shown in Table 7 (“-“ means that there are no parameters in that layer). A RP is a tool for nonlinear data analysis. It visualizes a square matrix; whose elements correspond to those times at which a state of a dynamical system recurs [22]. The main advantage of RP is that it can show useful information even with some short data [23]. In this project, the functions used are from the class library called SciPy [24] to implement the RP conversion with eps=0.1 and steps=100, which control the size of the plot.

![Fig. 3 Transfer sequence data to recurrence plot](image1)

**Fig. 3 Transfer sequence data to recurrence plot**

We choose a sigmoid function as the activation function, after the element-wise add layer. Our model has two kinds of input. We send image-input to VGG model and send sequence-input to CNN model. Then,
we merge the outputs together after flatten layer and get the final result. At first, we replace sequence-input with image-input. The accuracy of result improves but it still not meets our expectations. Then we propose a model combine these two inputs with two paths respectively, shows in Fig. 4. By comparison with single input model, the accuracy of combined inputs model has been greatly improved. This maybe because the image signal can be treated as a new feature of our raw data, more effective feature will help the deep network learn better. When sequence-input is converted into image-input, some new features are found but meanwhile some features are lost. Because of that, only using image-input does not significantly improve the results. Recent work by Yongming Han [25] used fast Fourier transform as a layer in Deep Network. The 1D sequence inputs are converted to 2D after the multi-frequency decomposition (MFD), which is different with our new input feature model, TS-RP CNN. TS-RP CNN is the first to apply combine time series and corresponding PR plot as two input for 1D and 2D CNN respectively.

3. Experiments and Result

3.1. Noisy Sequence Generation

Since there is only one master meter for a residential area, it is easy to detect the status of master meter by the manager of the residential area. Therefore, master meters don’t need noisy generations. Instead, we should add noise to some of the sub meters, then the sum of them also have an increase pattern. However, some datasets without records of master meters cannot use this way to test. For example, the data set generated from CBTs does not record master meters. The measurement shift for a malfunctioning sub meter will increase as time goes according to the technical specifications of the smart meters. The increment can be considered as linearly or exponentially. Based on these rules, randomly select 30% users with random starting time(s) as malfunction, and modify every row of their daily usage since the starting date following this formula:

$$Usage_{new}(i) = (1 + \alpha \cdot (i - s)) \cdot Usage(i)$$  \hspace{1cm} (8)

where: $i$ means date of this row, $\alpha$ means error percentage per day. We also test the performance of our model under another kind of noisy generation formula. We choose sigmoid function as the generation formula, which is described in equation 9 [26].

$$Usage_{new}(i) = Usage(i) \cdot (1 + \frac{0.75 + \frac{i}{\sqrt{225+i^2}}}{2})$$  \hspace{1cm} (9)

Following the manually changes of everyday usage, recurrence plots for this user, sum of the submeters, and error (from Equation 1) change as well. Therefore, we can test our system by detecting the changes and classifying them into the abnormal class.

3.2. Result of Prediction of Error Based on LSTM

The standard to evaluate the result of prediction in this experiment is sensitivity, which is defined as the length between the detected date and the starting date. The detected date is computed by a sliding-window detection system, which is described in Fig. 1.

Our sliding-window detection system has two parameters called threshold($t$) and window length($l$) respectively. For each step, if all the measurement errors between predicted value and test value in the window are less than the threshold, move the window forward with one-day stride. Keep sliding until all the errors in the window exceed the threshold and return the right edge of the window as the sensitivity. The sensitivity increases with decreasing threshold and decreasing window length. The sensitivity can be used to adjust the accuracy of detection system. If the sensitivity is high, some normal users might be classified as abnormal user by mistake; if it is low, some abnormal users might be missed by the detection system.

The prediction is based on our LSTM model described in 2.3.1. Mean squared error (MSE) is selected as the loss function as our problem can be treated as a regression problem.
In order to evaluate the result more conveniently, the data are normalized. Most of the error between truth and predicted value is accurate. The line chart of the result of prediction result are shown in Fig. 5, which shows that the predicted value is close to the true value.

In order to find the most efficient sequence length for training for our LSTM model, we test the different sequence length for 10 times and find that sequence length from 40 days to 80 days have lower MSE. When sequence length is 80 days, our model will have a comparatively lower MSE which is less than 0.005, and a comparatively lower standard deviation, as shown in Fig. 6.

Predicted value and true value are sent to the detection process for CNN classification. Fig. 7 Malfunction detection examples (a) when the threshold is 0.5 and the window length is 4, one abnormal user’s meter is detected as malfunctioning meter. (b) with the same parameter setting, another normal user’s meter is not detected as malfunctioning meter.
where $p$ means the predicted value, and $t$ means the threshold of the system.

In Fig. 7(a), the parameters of the system are $t=0.5$ and $l=4$. The result is the malfunction is detected with a sensitivity of 65. Fig. 7(b) shows the result that there are no malfunctioning smart meters detected under the same parameter settings.

3.3. Result of Classification Based on CNN

We have trained our networks with the Keras [27] machine learning system on 2 Geforce 1080ti GPU. The parameters and architecture of the network are shown in Table 4.

We used Receiver Operating Characteristic (ROC) curve and Precision Recall Curve (PRC) to evaluate our result. We did a 5-fold cross validation and got the mean Area Under Curve (AUC) as our evaluating standard. The result for dataset generated by Eq. 8 is shown in Fig. 11, whose mean AUC of the ROC is 0.80±0.04 and the mean AUC of the PRC is

\[
UB = p + t
\]

\[
LR = p - t
\]
The result for dataset generated by Eq. 9 is shown in Fig. 9, which has a $0.69 \pm 0.11$ mean AUC of ROC and $0.66 \pm 0.20$ mean AUC of PRC. Therefore, for different formula of noisy generation. Our model’s performance is acceptable.

We also exclude the influence of different proportion of positive samples. The proportion of malfunctional meters ranged from 0.5 to 0.8. The mean AUC of ROC is stable, as shown on Fig. 10.

3.4. Comparison with Classical Methods

As we are the first to combine electric meter and deep learning, we compared our methods with the classical methods to test the improvement. In order to test our method, we used the same data as input and compare it with different models. We have tried many classical methods to do master meter’s error prediction as well. The result of the comparison is shown in Fig. 8. In Fig. 8, the red line stands for the true value of master meter’s error, and other lines are the predicted values of master meter’s error of using Bayesian Ridge [28], Elastic Net [29], Gradient Boosting Regression [30], and LSTM methods respectively. The range of the pink part is from the true value minus the threshold to the true value plus the threshold. The predicted values within this area are considered as acceptable predictions, so we call this pink area as the “Acceptable”. From Fig. 8, we can find that the values predicted by LSTM method has a higher probability to find malfunctioning meters as the pink part than other methods (the threshold of Fig. 8 is 8).

In order to match our detection method mentioned above, we create a new standard to evaluate the result of prediction. Table 6 shows that the number of days in range from the lower bound (true value minus threshold) to the upper bound (true value plus the threshold), which is also the pink part in Fig. 8. We tried different sizes of thresholds to get different results. In conclusion, LSTM and GBR outperform others in most situations. LSTM performs better than GBR in most cases. LSTM gets 95.8% accuracy when
threshold equals 8. This value is normalized and used in the detection system.

Table 5 Evaluation of the prediction

| Threshold | Classic Methods | LSTM |
|-----------|----------------|------|
|           | Elastic Net    | GBR  | 5 (6.9%) |
| 0.5       | 1 (1.4%)       | 5 (6.9%) |
| 1         | 1 (1.4%)       | 13 (18.1%) | 17 (23.6%) |
| 4         | 13 (18.1%)     | 52 (72.2%) | 42 (58.3%) |
| 6         | 52 (72.2%)     | 58 (80.6%) | 65 (90.3%) |
| 8         | 66 (91.7%)     | 59 (81.9%) | 69 (95.8%) |

To compare our combined model with the single input models. We compute the AUC of ROC for image-input and sequence-input. After testing, the best result shows that AUC is 0.61 when sequence data are taken as input and AUC can get 0.79 when images data is taken as input. It turns out that combining(mean AUC gets 0.80) these two inputs can get a more accurate result.

Table 6 Comparation between different models

| AUC of ROC | Model       |
|------------|-------------|
|            |             |
| Fold 1     | 0.29        | 0.47        | 0.75 |
| Fold 2     | 0.62        | 0.71        | 0.97 |
| Fold 3     | 0.46        | 0.17        | 0.79 |
| Fold 4     | 0.68        | 0.41        | 0.80 |
| Fold 5     | 0.55        | 0.79        | 0.80 |
| Mean ±1 std. | 0.52±0.14 | 0.51±0.22 | 0.80±0.04 |

4. Conclusion

This work has proposed a system for smart meter malfunction detection. The main strategy of this system is to judge whether there are any malfunctioning meters, if so, then use the CNN model to identify which smart meter is failed. In the first stage, we use the LSTM model to make a prediction for the electricity based on collected data. Then the sliding window is used to detect the existence of malfunctioning smart meters. As mentioned before, if all the errors in the window exceed the threshold, the system will consider that there are malfunctioning smart meters in the residential area. Once the system finds existence of malfunctioning meters, it will use the CNN model to classify all the users in this residential area. To achieve this goal, we proposed a novel 1D and 2D combined CNN model, called TS-RP CNN. When using the CNN model, the system will convert the matrix of data into recurrence plots and use both of them in order to get a better result.

In fact, detection of malfunctioning smart meters is very important, especially for some countries that have great demand for smart meters, but there are few researchers studying on this topic. Our study can give other researchers a direction to solve this problem and similar problems.

Reference

[1] C. Liu. Smart Meter Development Prospects and Market Capacity Analysis. Market Research. (2015) 36-7.
[2] R. Silipo, P. Winters. Big data, smart energy, and predictive analytics. Time Series Prediction of Smart Energy Data. 1 (2013) 37.
[3] Y. DeCusno, S. Lyons, A. Nelen. Estimating the impact of time-of-use pricing on Irish electricity demand. The Energy Journal. (2014) 117-36.
[4] D. N. Nikovski, Z. Wang, A. Esenther, H. Sun, K. Sugihara, T. Muso, et al. Smart meter data analysis for power theft detection. International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer2013. pp. 379-89.
[5] Z. Zhang, Q. He, J. Gao, M. Ni. A deep learning approach for detecting traffic accidents from social media data. Transportation research part C: emerging technologies. 86 (2018) 500-96.
[6] T-Y. Kim, S. B. Cho. Web traffic anomaly detection using C-LSTM neural networks. Expert Systems with Applications. 106 (2018) 66-76.
[7] T-A. Zafferey, Ulbig; Stephan, Koch; Gabriela, Hug Forecasting of smart meter time series based on neural networks. International Workshop on Data Analytics for Renewable Energy Integration (2016) 10-21.
[8] S. Wamim, L. Keyan, N. Huauna, W. Yulu, Z. Jingxiang. The anomalous data identification study of reactive power optimization system based on big data. Probabilistic Methods. Applied to Power Systems (PMAPS), 2016 International Conference on. IEEE2016. pp. 1-5.
[9] N. He. Research on Big Data Analysis of Smart Meter Fault. Management & technology of SME. (2016) 142-5.
[10] J. Yang, P. Jiang, G. Chen, T. Yuan, F. Xue. Study on Smart Meters Fault Identification Model. Electrical Energy Management Technology. (2017) 30-6.
[11] H. Yao, X. Wang, L. Wu, D. Jiang, T. Liao, D. Liang. Prediction Method for Smart Meter Life Based On Big Data. Procedia engineering. 211 (2018) 1111-4.
[12] Y. Bengio. Learning deep architectures for AI. Foundations trends® in Machine Learning. 2 (2009) 1-127.
[13] Y. Bengio, P. Simard, P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks. 5 (1994) 155-67.
[14] D. Monner, J.A. Reggia. A generalized LSTM memory. Neural Networks. 25 (2012) 70-83.
[15] S. Hochzeiter, J. Schmidhuber. Long short-term memory. Neural computation. 9 (1997) 1735-40.
[16] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber. LSTM: A search space odyssey. IEEE transactions on neural networks learning systems. 28 (2017) 2222-32.
[17] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. Proceedings of the IEEE conference on computer vision and pattern recognition workshops 2014. pp. 806-13.
[18] H. Zhang. LSTM and GRU – Formula Summary. 2017.
[19] N. Marwan. Recurrence Plots and Cross Recurrence Plots. 2016.
[20] N. Marwan, M.C. Romano, M. Thiel, J. Kurths. Recurrence plots for the analysis of complex systems. Physics reports. 438 (2007) 237-329.
[21] E. Breslau. Scipy and Numpy: an overview for developers. * O'Reilly Media, Inc.* 2012.
[22] S.J. Pan, Q. Yang. A survey on transfer learning. IEEE Transactions on knowledge data engineering. 22 (2010) 1345-59.
[23] Keras Documentation. 2016.
[24] Q. Shi, M. Abdel-Aty, J. Lee. A Bayesian ridge regression analysis of congestion's impact on urban expressway safety. Accident Analysis Prevention. 88 (2016) 124-37
[25] H. Zou, T. Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B. 67 (2005) 301-20.
[26] C. Li. A Gentle introduction to gradient boosting. (2016).