Predicting the Household Power Consumption Using CNN-LSTM Hybrid Networks

Tae-Young Kim and Sung-Bae Cho(✉)

Department of Computer Science, Yonsei University, Seoul, Republic of Korea
{taeyoungkim, sbcho}@yonsei.ac.kr

Abstract. Prediction of power consumption is an integral part of the operation and planning of the electricity supply company. In terms of power supply and demand, For the stable supply of electricity, the reserve power must be prepared.

However, it is necessary to predict electricity demand because electricity is difficult to store. In this paper, we propose a CNN-LSTM hybrid network that can extract spatio-temporal information to effectively predict the house power consumption. Experiments have shown that CNN-LSTM hybrid networks, which linearly combine convolutional neural network (CNN), long short-term memory (LSTM) and deep neural network (DNN), can extract irregular features of electric power consumption. The CNN layer is used to reduce the spectrum of spatial information, the LSTM layer is suitable for modeling temporal information, the DNN layer generates a predicted time series. The CNN-LSTM hybrid approach almost completely predicts power consumption. Finally, the CNN-LSTM hybrid method achieves higher root mean square error (RMSE) than traditional predictive methods for the individual household power consumption data sets provided by the UCI repository.

Keywords: Power consumption prediction · Deep learning
Convolutional neural network · Long short-term memory

1 Introduction

A power system is a sophisticated system that simultaneously handles demand and supply, balancing demand and supply. Household energy consumption is also steadily increasing as population is increasing and citizens' standard of living improved [1]. However, since electric energy cannot be stored, it is necessary to forecast electricity demand [2]. Energy consumption forecasting is a multivariate time series problem with several variables that determine power consumption [3]. The variables vary according to the user's consumption pattern and affect the power consumption. Figure 1 shows a visualization of the individual household power consumption data set provided by the UCI repository. This data has a total of nine variables and is used as a benchmark for power demand forecasting. Figure 2 shows the trend components for the individual home power consumption data sets. The trend component was extracted using time series decomposition method. This trend is very complex and irregular. Therefore, it is difficult to forecast power demand using existing statistical techniques. In this paper, we propose a CNN-LSTM hybrid neural network that linearly connects CNN and

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11314, pp. 481–490, 2018.
https://doi.org/10.1007/978-3-030-03493-1_50

LSTM to automatically predict the household power consumption. Power consumption, which is a multivariate time series, includes spatial and temporal information. Therefore, the CNN-LSTM hybrid neural network can extract the space-time feature of the power consumption variable to predict the household power consumption.

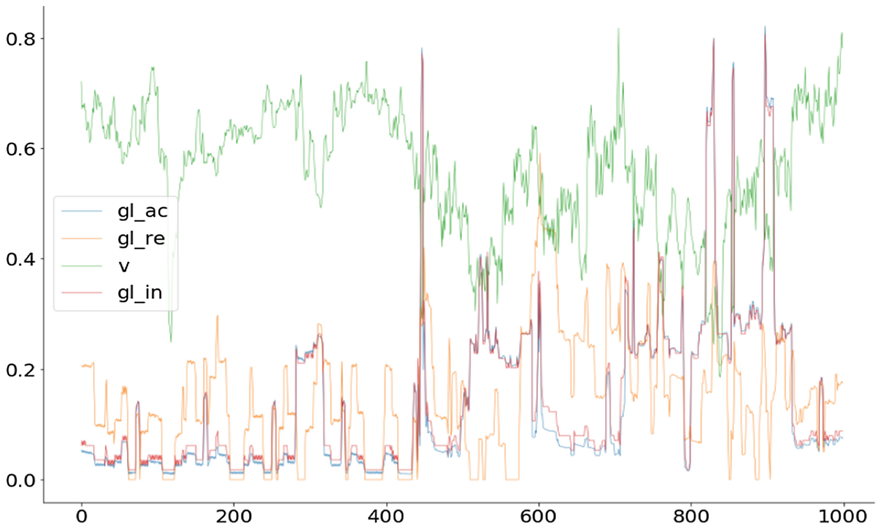

Fig. 1. Attribute of household power consumption data

The proposed CNN-LSTM method reduces the spectrum of time series data by using the convolution and the pooling layer. The output of this CNN layer is passed as input to the LSTM layer to model the temporal information. The output of the LSTM unit is used as an input to the fully connected layer to generate a time series that

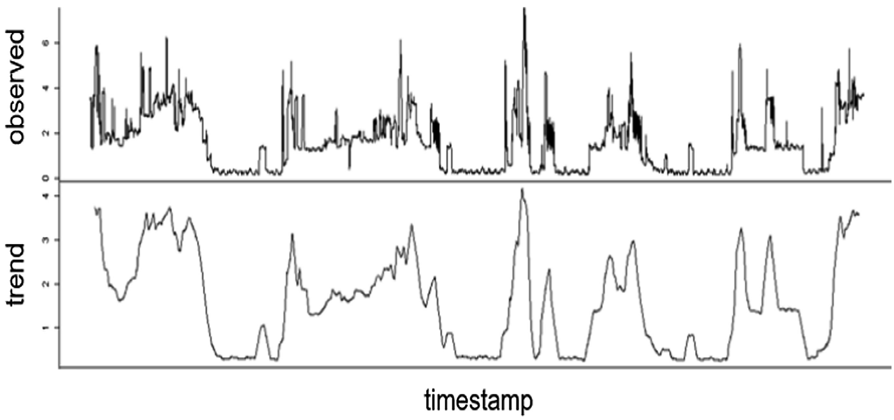

Fig. 2. Trend component for individual home power consumption dataset

predicts power demand. Finally, we tuned some parameters to see if the performance of CNN-LSTM improves. In particular, we analyze the internal output process of the proposed method of extracting spatial features from time series data using CNN. We also compare performance with the LSTM model.

The remainder of this paper is organized as follows. Section 2 discusses related work on time series data prediction. Section 3 describes the proposed CNN-LSTM hybrid network architecture. Section 4 presents experiment settings and results. Section 5 concludes this paper.

2 Related Works

Table 1 shows the related work for processing time series data. Many researchers have studied various feature extraction methods to predict time series data. Methods for extracting features from a time series can be divided into three categories: statistical modeling, spatial information modeling, and temporal information modeling.

Table 1. Related work on time series preocessing

Category	Author	Year	Method	Description
Statistical modeling	Nychis [4]	2008	Entropy based	Entropy based correlation analysis
	Münz [5]	2007	K-means clustering	Classification using cluster
	Zhang [6]	2006	Random forest	Outlier pattern extraction
Spatial information modeling	Ince [7]	2016	1D-CNN	Sliding window-based feature extraction
	Kiranyaz [8]	2015	1D-CNN	ECG feature classification
	Souza [9]	2014	SVM	Sequence texture mapping
Temporal information modeling	Bontemps [10]	2016	LSTM	Normal signal prediction
	Taylor [11]	2016	LSTM	Prediction and exception calculation
	Malhotra [12]	2015	Stacked LSTM	Using prediction error distribution

Münz et al. use k-means clustering algorithm in time series data to predict time series of irregular patterns. They calculated the center value of the cluster and classified the time series into regular and irregular trend according to the distance [5]. Zhang and Zulkernine used a random forest algorithm to detect outliers in time series data. They attempted to predict irregular features using unsupervised learning [6]. These methods provide high performance when estimating the value of time series data. However, a value having the same distribution as the normal sequence cannot be properly predicted.

Kiranyaz et al. tried to predict the signal by extracting the feature from the patient's electrocardiogram signal. They split the ECG signal into time fragments and extracted features using 1D CNN [8]. Souze et al. created a local feature by mapping the time series data to a texture. They created texture features using SVM [9]. This method efficiently extracts spatial information from a sequence of complex patterns with noises. Compared to previous research, they get better predictive performance. However, the time information of the time series data is lost by the convolution and the pulling operation.

Bontemps et al. attempted to predict the time-series using LSTM [10]. Malhotra et al. classify irregular signals into sensor data from various equipment using a stacked LSTM structure [12]. They learned the normal signal to the LSTM and then calculated the actual data and error distribution using the predicted signal. These methods can easily predict in time series data with periodicity. However, if the time series data does not have a period, it cannot predict the actual power consumption.

3 Proposed Method

3.1 CNN-LSTM Hybrid Neural Network

The proposed CNN-LSTM hybrid neural network consists of a linear structure of CNN and LSTM layers. Figure 3 shows the structure for predicting household power consumption using the proposed CNN-LSTM. CNN-LSTM model uses time-series data

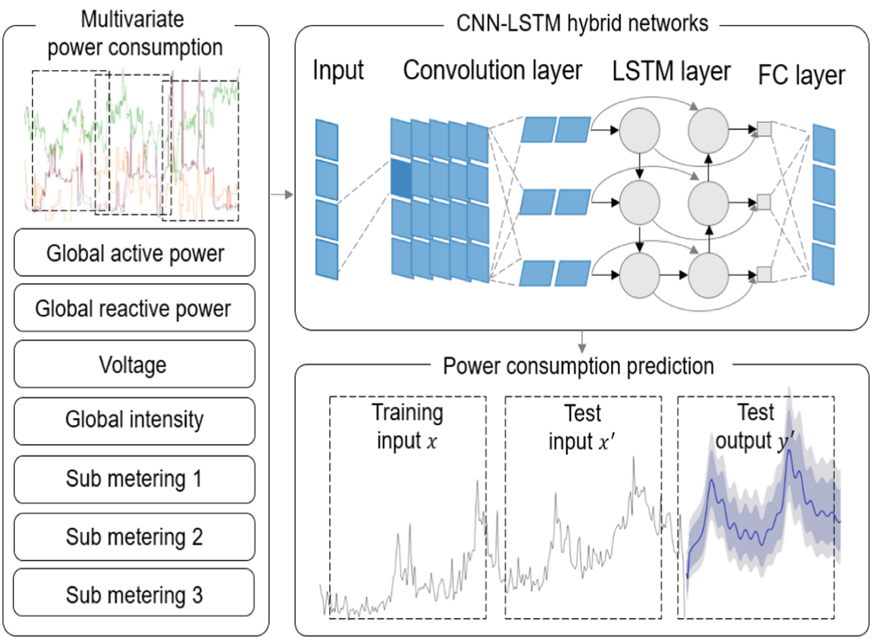

Fig. 3. The proposed household power prediction structure

preprocessed with sliding window algorithm. The preprocessed window extracts spatial features through convolution and pooling operations [13]. Then, temporal feature of the data is modeled by the LSTM. The training model performs household power consumption prediction on the test data in the DNN layer

Assume that $x_i^0 = \{x_1, x_2, \cdots, x_n\}$ is a input vector preprocessed by the sliding window algorithm and n is the number of normalized values. Equation 1 passes x_i to the first convolutional layer to derive the output vector y_{ij}^l. This can be calculated using x_{ij}^l, which is the output vector of the previous layer. b_j^l is the bias for the j^{th} convolution kernel, w is the weight of the convolution kernel, m is the index value of each kernel filter, and σ is the activation function. we used ReLu as an activation feature.

$$y_{ij}^l = \sigma\left(\sum_{m=1}^{M} w_{m,j}^l x_{i+m-1,j}^0 + b_j^l\right) \tag{1}$$

The pooling layer uses max-pooling and is applied independently for each depth slice to reduce the spatial size. Power Consumption Reduces the number of parameters and computation by reducing the space size of the time series. Using the Eq. 2, a pooling layer operation can be performed. R is the input vector whose resolution is reduced by the convolution operation, is a pooling size smaller than the size of y, and T is a stride that determines how far to move the pooled area.

$$p_{ij}^l = \max_{r \in R} y_{i \times T + r, j}^{l-1} \tag{2}$$

In the LSTM layer, multiple memory cells are used to store historical power consumption. Using these cells allows us to remember power consumption for a long time to make it easier to understand. The output value previously calculated on the CNN layer is used as input to the LSTM layer. Each LSTM unit consists of gates and updates the cell according to the active state and is controlled to a continuous value between 0 and 1. The LSTM has three gates, which are inputs, outputs, and forgetting gates. The hidden value of the LSTM cell, h_t is updated every time interval t.

$$i_t = \sigma\left(W_{pi} p_t + W_{hi} h_{t-1} + W_{ci} \circ c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{pf} p_t + W_{hf} h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \tag{4}$$

$$o_t = \sigma\left(W_{po} p_t + W_{ho} h_{t-1} + W_{co} \circ c_t + b_o\right) \tag{5}$$

Equations 3, 4, and 5 are denoted i, f, and o, respectively, and are used to calculate input, forget, and output gate activation values. Equations 6 and 7 show the process of calculating the cell state and the hidden state, using notation c and h, respectively. These two values are determined by the input, forget and output gate enable values. σ is calculated using the tanh activation function. The term p_t is used as an input to the memory cell and is the output generated at time t in the previous CNN layer. W is the weight matrix of the LSTM unit, b is the bias vector, and $\circ$ represents Hamad product for cell state calculation.

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma(W_{pc}p_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$h_t = o_t \circ \sigma(c_t) \quad (7)$$

Fully-connected layer can help you to predict your home power consumption more easily. These are the last link layer of CNN-LSTM. The output of the LSTM layer is flattened to the vector $h^l = \{h_1, h_2, \ldots h_l\}$, and transferred to the fully-connected layer. Where l is the total number of units in the output of the LSTM. Equation 8 is used in a fully-connected layer. σ is the activation function, w is the weight of the i^{th} node in layer $l-1$ and j^{th} node in layer l, and b_i^{l-1} is the bias. We can generate predicted time series in a fully-connected layer.

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma(W_{pc}p_t + W_{hc}h_{t-1} + b_c) \quad (8)$$

3.2 CNN LSTM Hybrid Architecture

The CNN-LSTM include various structures depending on the number of convolution, pooling, lstm, and dense layers [14]. It can also change the kernel size, stride size, activation function. By appropriately adjusting the parameters, more features can be extracted from the learning data, affecting the learning speed and can have a significant impact on performance. To change the parameters and determine the CNN-LSTM architecture, we need to understand the characteristics of univariate time series data. We can generate a predicted time series from the input power consumption. The input of the CNN-LSTM is a time slice of length 60. Therefore, 2×1 kernel is used to extract the most suitable features minimizing the loss of information. The convolution and pooling layers are stacked in two layers to update and learn more weights. LSTM uses 64 units to model temporal feature. Hyperbolic tangent is used as an activation function to increase the convergence rate. Table 2 represents the proposed CNN-LSTM architecture.

Table 2. The Proposed CNN-LSTM architecture

Layer	Kernel size	Stride size	# parameter
Convolution	2×1	1	192
Activation (ReLu)	-	-	0
Pooling	2×1	2	0
Convolution	2×1	1	8.256
Activation (ReLu)	-	-	0
Pooling	2×1	2	0
TimeDistributed	-		0
LSTM (64)	-		180,480
Activation (tanh)	-		0
Dense (32)	-		2,080
Dense (60)	-		1,980

4 Experiment and Results

4.1 Individual Household Electric Power Consumption Dataset

In this paper, we use the individual household electric power consumption dataset provided by UCI [15]. This data is a set of data for one generation of power consumption with a one-minute sampling rate over the long term from 2006 to 2010. The data is composed of 9 attributes and the Date (2006/12/16 ~ 2010/11/26), Time (minute), Global_active_power (kW), Global_reactive_power (kW), Voltage (V), Global_intensity (Wh). We use the "Global Active Power" variable among the nine attributes for power demand forecasting. This variable is the average total active power in kWh (kWh). Raw data was not ready to construct the prediction model because some of the values were missing and the recorded time frame was inappropriate. Because of the lack of information, the prediction efficiency of the predictive model may deteriorate. Because the input value for CNN-LSTM is between 0 and 1, we had to pre-process the power station congestion. The values were normalized using Eq. 9. The data consists of a total of 2,075,259 time series data and there are 25,979 missing values. We removed all missing values in the data preprocessing process. We pre - processed multivariate time - series data by the sliding window algorithm.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{9}$$

4.2 Comparison of Loss Between LSTM and CNN-LSTM Model

To verify the usefulness of the proposed method, we compared the learning loss graphs of the LSTM and CNN-LSTM models. The proposed CNN-LSTM hybrid model showed stable learning compared to the existing LSTM method. The left side of Fig. 4 shows that the validation loss of the LSTM model is very unstable. The right side shows the loss of the CNN-LSTM model. It shows unstable in the middle of learning but can be stabilized again.

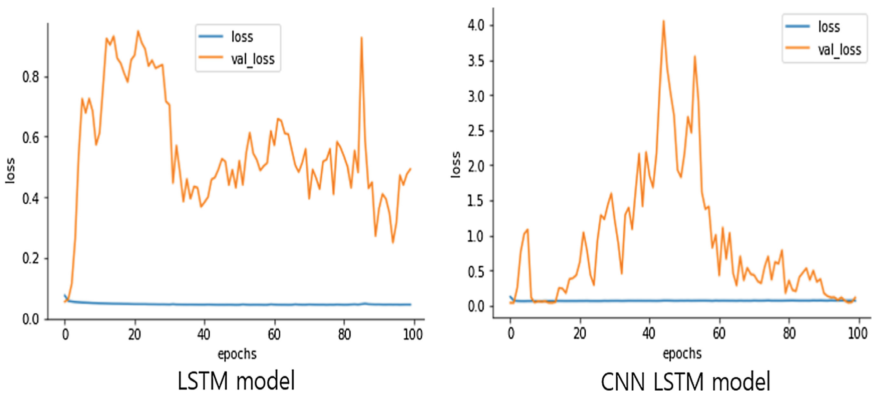

Fig. 4. Loss and validation loss per epoch

4.3 Performance Comparison of Deep Learning Model

Figure 5 shows different machine learning methods and performance comparisons. The proposed CNN-LSTM method achieves higher performance than other models. Root mean square error (RMSE) was used as an evaluation metric of the learning model.

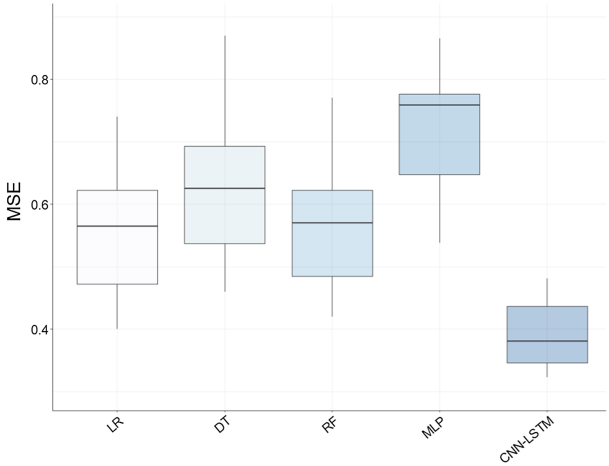

Fig. 5. The RMSE of 10-fold cross validation using other methods

4.4 C-LSTM Model Internal Analysis

We confirmed the operation of the CNN-LSTM through internal visualization. In particular, we can see how the power consumption input changes through the CNN layer. The intermediate outputs were analyzed in C-LSTM neural network using Individual household electric power consumption dataset. Figure 6 shows the outputs

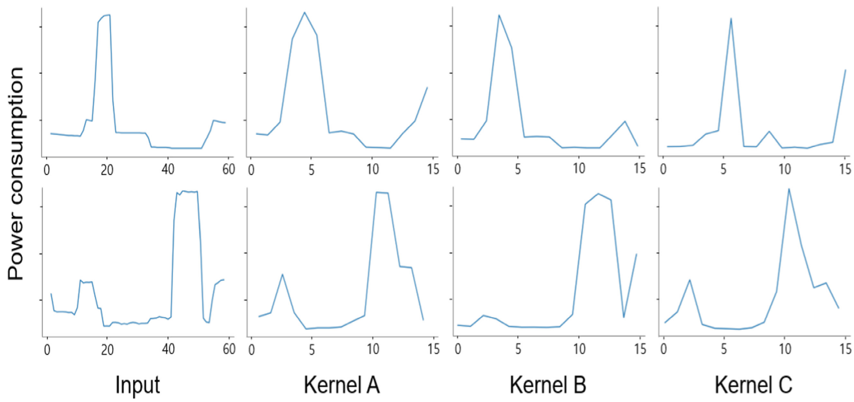

Fig. 6. The CNN-LSTM internal output according to kernel

of the CNN-LSTM layers. Each CNN layer learns to extract the most appropriate features while changing kernel weights. Each layer of CNN reduces the overall length of the input sequence by convolution and pulling operations. However, Fig. 6 confirms that the intermediate output maintains the spatial and temporal feature.

4.5 Visualize Forecasting Results Through CNN-LSTM Hybrid Model

Figure 7 shows a graph that visualizes the predicted results using the CNN-LSTM model. We have confirmed that the power consumption prediction in the individual household electric power consumption dataset has been successful. Predictive results can be visually confirmed similar to the ground truth. In addition, it can be seen that excellent prediction performance is achieved even in situations where the periodicity is not observed. We can visually confirm that the proposed CNN-LSTM model performs well in power consumption prediction. We can confirm that the proposed CNN-LSTM model predicts local features well in power consumption prediction.

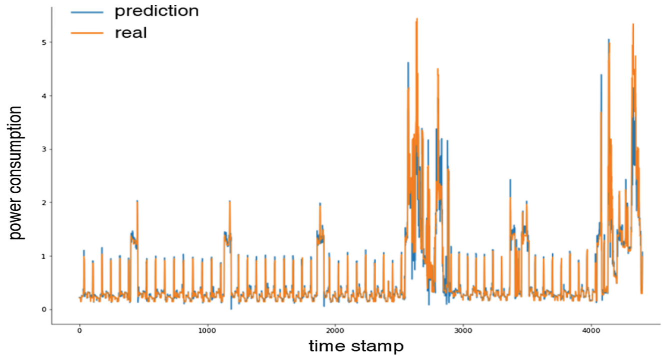

Fig. 7. Graph of predicted results using CNN-LSTM

5 Conclusion

We proposed a CNN-LSTM hybrid architecture to predict power consumption. We have demonstrated usability and excellence by comparing the proposed model with other machine learning methods. We also found an optimal CNN-LSTM prediction model through performance experiments and parameter tuning. This method can quickly and accurately predict the irregular trend of energy consumption in household power consumption dataset. We use the CNN-LSTM method to automatically extract correlations and time information from variables in multivariate time series. We also checked how time series data changes to predict power consumption through CNN-LSTM internal analysis. We also analyzed how time series data predicts power consumption through CNN-LSTM internal analysis. The proposed CNN-LSTM neural network predicts time-series characteristics that were difficult to predict with conventional machine learning methods. However, since the proposed method is preprocessed by the sliding window algorithm, there is a delay in predicting the actual data. This problem remains a challenge.

Acknowledgement. This research was supported by Korea Electric Power Corporation. (Grant number: R18XA05).

References

1. Cho, S.-B., Yu, J.-M.: Hierarchical modular Bayesian networks for low-power context-aware smartphone. Neurocomputing (2017)
2. Kant, G., Sangwan, K.S.: Prediction and optimization of machining parameters for minimizing power consumption and surface roughness in machining. J. Clean. Prod. **83**, 151–164 (2014)
3. Hernandez, L., et al.: A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. IEEE Commun. Surv. Tutor. **16**(3), 1460–1495 (2014)
4. Nychis, G., Sekar, V., Andersen, D.G., Kim, H., Zhang, H.: An empirical evaluation of entropy-based traffic anomaly detection. In: Proceedings of the 8th ACM SIGCOMM Internet Measurement Conference, pp. 151–156 (2008)
5. Münz, G., Li, S., Carle, G.: Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet (2007)
6. Zhang, J., Zulkernine, M.: Anomaly based network intrusion detection with unsupervised outlier detection. In: IEEE International Conference on Communications, vol. 5, pp. 2388–2393 (2006)
7. Ince, T., Kiranyaz, S., Eren, L., Askar, M., Gabbouj, M.: Real-time motor fault detection by 1-D convolutional neural networks. IEEE Trans. Ind. Electron. **63**(11), 7067–7075 (2016)
8. Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patient-specific ECG classification by 1-D convolutional neural networks. IEEE Trans. Biomed. Eng. **63**(3), 664–675 (2016)
9. Souza, V.M., Silva, D.F., Batista, G.E.: Extracting texture features for time series classification. In: 22nd International Conference on Pattern Recognition, pp. 1425–1430 (2014)
10. Bontemps, L., Cao, V.L., McDermott, J., Le-Khac, N.-A.: Collective anomaly detection based on long short-term memory recurrent neural networks. In: Dang, T.K., Wagner, R., Küng, J., Thoai, N., Takizawa, M., Neuhold, E. (eds.) FDSE 2016. LNCS, vol. 10018, pp. 141–152. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48057-2_9
11. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: Analytics IEEE International Conference on Data Science and Advanced, pp. 130–139 (2016)
12. Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long short term memory networks for anomaly detection in time series. In: European Symposium on Artificial Neural Networks, Computational Intelligence, p. 89 (2015)
13. Ronao, C.A., Cho, S.B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. **59**, 235–244 (2016)
14. Kim, T.Y., Cho, S.-B.: Web traffic anomaly detection using C-LSTM neural networks. Expert Syst. Appl. **106**, 66–76 (2018)
15. Loginov, A., Heywood, M.I., Wilson, G.: Benchmarking a coevolutionary streaming classifier under the individual household electric power consumption dataset. In: IEEE International Joint Conference on Neural Networks, pp. 2834–2841 (2016)