Clustering Enabled Few-Shot Load Forecasting

Qiyuan Wang*, Zhihui Chen*
School of Data Science
The Chinese University of Hongkong, Shenzhen
Shenzhen, Guangdong, 518172 China
{qiyuanwang, zhihuichen}@link.cuhk.edu.cn

Chenye Wu†
School of Science and Engineering
The Chinese University of Hongkong, Shenzhen
Shenzhen, Guangdong, 518172 China
chenyewu@yeah.net

***Abstract*—While the advanced machine learning algorithms are effective in load forecasting, they often suffer from the low data utilization, and hence their superior performance relies on huge datasets. Unfortunately, such huge scale datasets may not be available for all tasks. In this paper, we consider the load forecasting for a new user in the system by observing only few shots (data points) of its energy consumption. We propose to utilize clustering to mitigate the challenges brought by the limited samples. Specifically, we first design a feature extraction clustering method for categorizing the historical data. Then, the load forecast for new users is conducted through a two-phase Long Short Term Memory (LSTM) model, which inherits prior knowledge from the clustering results. The proposed method outperforms traditional LSTM model, especially when the training sample size fails to cover a whole period (i.e., 24 hours in our task). Extensive case studies on two real world datasets and one synthetic dataset verify the effectiveness and efficiency of our method. We also numerically suggest the minimal number of shots to guarantee satisfactory forecast result.**

***Index Terms*—Load Forecasting, Few-Shot Learning, Time Series Analysis**

I. Introduction

Load forecasting, a classical procedure in the electricity sector, is the basis for efficient power system control as well as effective electricity market operation. As a time series forecasting task, its toolbox has been fundamentally reshaped by the rise of deep learning technologies. For example, dependency learning models such as recursive neural network (RNN) [1] and feature learning models like convolutional neural network (CNN) [2] are both capable of extracting complex statistics and learning representative features from huge datasets. However, such models are often not very data efficient, i.e., deep learning approaches suffer from poor sample efficiency in stark contrast to human perception. As shown in Fig. 1, when only provided with a short sequence of historical data, the prediction results of deep learning models such as Long Short Term Memory (LSTM) are far from satisfactory.

While such an observation is not surprising, we may utilize the patterns in load profiles to help LSTM improve its performance. Specifically, there are limited number of underlying daily energy consumption patterns [3]. Thus, we may first observe a short sequence of energy consumption profile from a new user (an unknown sample) and try to identify its consumption pattern. Once such identification is successful, the LSTM may utilize information in the pattern as rich historical data and the short sequence as short term memory to improve the performance. We term this task the clustering enabled few-shot load forecasting.

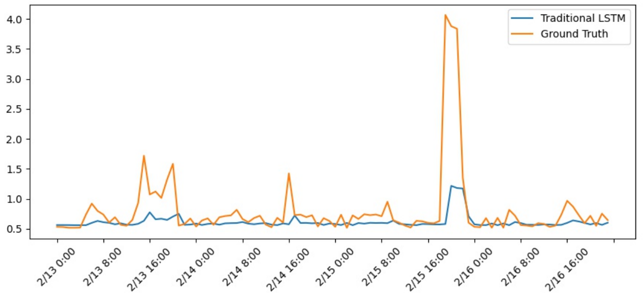

Fig. 1. 96 hours LSTM Forecasting on 12-shot Training Set

Specifically, this work integrates ensemble clustering and two-phase LSTM model in order to achieve better forecasting accuracy based on few-shot samples. As shown in Fig. 2, these limited samples will first be classified into similar clusters with base class data. Then the LSTM model will first be trained (this can be done offline and hence being a pretrained model) with the long-term denoised mean-averaging data of the specific cluster. By further fine-tuning the pretrained model with the few-shot samples, the resulting two-phase LSTM is able to utilize the prior knowledge of the cluster and the real time information of the new user. Together, they guarantee a remarkable performance.

The remainder of the paper is organized as follows. Section II reviews the literature on time series forecasting, clustering and few-shot learning (FSL). Then, Section III introduces our proposed two-phase LSTM model in detail. To validate the performance of few-shot forecasting, we introduce the performance metrics, dataset overview and case study design in Section IV. Comprehensive numerical studies are conducted in Section V. Finally, Section VI gives the concluding remarks and points out interesting future directions.

II. Related works

We identify three major streams of related works. The first one seeks to apply time series forecasting in the electricity

*The first two authors contributed equally to this work, and hence are co-first authors of this work.

†C. Wu is the correspondence author.

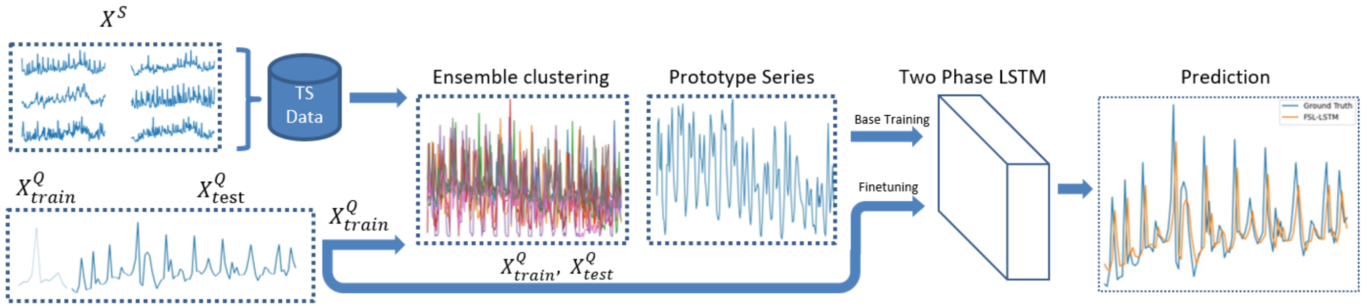

Fig. 2. The Framework of FSL-LSTM

sector. The second one investigates the time series clustering techniques, while the third one targets to advance FSL.

A. Time Series Forecasting in Electricity Sector

Time series forecasting is applied in the electricity sector to facilitate decision making [4]. Particularly, in electricity sector, load forecasting has long been an important research topic. Statistical and machine learning based methods are widely applied in load forecasting. In [5], Huang and Shih presented an Auto-regressive moving average (ARMA) procedure for load forecasting characterizing non-Gaussian process. The ARMA model can be extended to Auto-regressive Integrated Moving Average (ARIMA) model which is widely used in forecasting electricity load and market price [6].

Recently, machine learning techniques have become particularly popular in load forecasting. In [7], a support vector regression model with empirical mode decomposition method was proposed. In [8], Park $et\ al.$ presented a multi-layered perceptron artificial neural network (ANN) that interpolates among the load and temperature data. In [9], Elman neural network based forecast engine with empirical mode decomposition was proposed as a novel method for predicting load signal. Introduced by Hochreiter $et\ al.$ in [10], LSTM has received enormous attention in this area due to its capacity of capturing long-distance statistical regularities, e.g., in [11]–[13], LSTM based deep learning forecasting frameworks were used in load forecasting.

B. Time Series Clustering

Time series clustering has been a hot topic in data mining. Compared with the classical clustering method, time series clustering is more complicated due to the temporal dynamics. Therefore, on top of the common clustering methods, time series clustering also cares about the similarity measurement as well as the feature extraction.

The most classical time series clustering is based on temporal similarity metrics, such as Euclidean distance (ED) [14] and dynamic time warping (DTW) [15]. Although such distance metrics are easy to implement in practice, they suffer from fatal demerits: ED suffers from the dimensionality curse [16], while DTW is overly sensitive to locally changes.

To overcome the demerits of similarity based clustering, feature extraction based clustering methods are investigated. Such methods first extract the key features in the time series and then conduct the clustering in low dimensional feature space. Thus they can better capture the global feature of time series. The most fundamental feature extraction tools include Discrete Fourier transform (DFT) and discrete cosine transform (DCT) [17]. Another widely adopted feature extraction tool is the discrete wavelet transform (DWT) [18]. In this work, we follow the novel feature extraction workflow based on DWT in [19], where Hacine-Gharbi $et\ al.$ proposed wavelet cepstral coefficient (WCC) for feature extraction, and then utilized a hidden Markov model for electricity appliance identification. This procedure achieves a completeness ratio of 98.13% when the decomposition level is five.

C. Few-Shot Learning

The objective of FSL is to learn new tasks supported by only a few samples with supervised information. FSL enables the learning of rare cases and relieves the burden of large scale data collection. One approach is to constrain hypothesis space $\mathcal{H}$ by prior knowledge in the learning process. For example, Caruana proposed Multitask Learning [20], an inductive transfer mechanism to improve generalization performance by using domain information contained in training signals of related tasks.

Another approach is to alter search strategy in hypothesis space $\mathcal{H}$ by using prior knowledge to provide a good initialization or guiding the search steps [21]. To guide the search steps or alter the search strategy by prior knowledge, a popular approach is to apply meta-learning to continuously refine the parameters according to the learner's performance. One representative method is model-agnostic meta-learning (MAML), proposed in [22]. Many efforts have also been devoted to achieve FSL by fine-tuning the parameter from a good initialization, including those based on generated-adversarial network (GAN) [23] and convolutional neural network (CNN) [24], [25]. However, to our best knowledge, few attempts have been made to extend these approaches to LSTM for time series forecasting. In our work, we make use of historical data to provide a good initialization which enables LSTM to perform fast adaption to novel load forecasting tasks.

III. FSL for Load Forecasting

Our proposed FSL framework consists of two major components: the basic ensemble clustering, and a two-phase LSTM forecasting network. For the first component, we use compact selected features extracted from wavelet analysis and other statistic descriptors, as shown in Fig. 3. For the second component, as we have mentioned, we follow [10] to implement the LSTM, utilizing wavelet denoising and data enhancement.

A. Feature Extraction for Clustering

1) Discrete wavelet analysis: The whole procedure starts with an ensemble clustering where few-shot samples are clustered with historical data. The historical data are segmented according to the length and the time stamps of the few-shot samples (k-shot) in order to represent the same period of time in a day. To reduce the dimensionality of the sequence set, wavelet analysis is adopted to project the original data onto a lower dimensional feature space. We compute three descriptors, namely discrete wavelet energy (DWE), log wavelet energy (LWE) and WCC.

As proposed in [19], we follow a feature extraction workflow based on wavelet analysis (shown in Fig. 3). Instead of applying DWT, we use discrete wavelet package transform (DWPT) [26] to decompose original time series into a balanced tree structure. In each level j, the total number of wavelet samples is equal to 2^j, where each leaf node represents a set of wavelet coefficients either in high or low frequency.

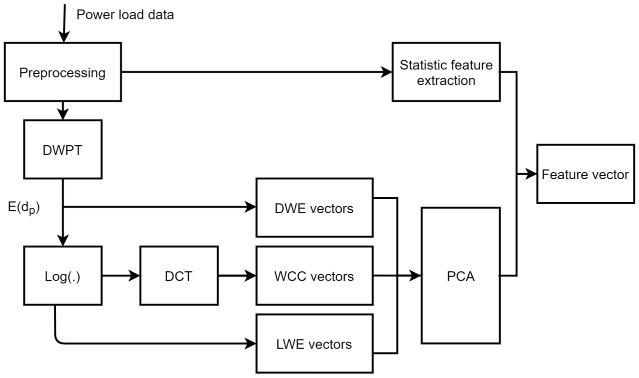

Fig. 3. Feature Extraction Workflow

Consider a DWPT balanced tree with total L levels of decomposition, the DWE value of a specific set of wavelet coefficient at level j, denoted by $E(d_j)$, with N_j number of detailed coefficients within the level, is calculated as:

$$DWE(d_j) = \frac{1}{E}\sum_{n=1}^{N_j} \| d_j[n]\|_2^2, 1 \leq j \leq L \tag{1}$$

where the l_2-norm of each wavelet coefficient d_j is scaled to the total energy E of all levels. The LWE is then calculated by applying log_{10} to DWE feature vectors, in order to achieve decorrelation of the energy values between different levels, which is defined as:

$$LWE(d_j) = \log\left(\frac{1}{E}\sum_{n=1}^{N_j} \| d_j[n]\|_2^2\right) \tag{2}$$

Based on the result of LWE, we further calculate the WCC feature vectors by applying DCT:

$$WCC(d_j) = DCT\left[\log\left(\frac{1}{E}\sum_{n=1}^{N_j} \| d_j[n]\|_2^2\right)\right] \tag{3}$$

After the derivative of WCC, we combine DWE, LWE and WCC feature vectors into one feature vector and apply Principle Component Analysis (PCA) to reduce the dimensionality of the feature space.

2) Other statistical feature: To represent time series data in a more comprehensive way, we further introduce several statistical features directly extracted from the time domain.

- Seasonal and trend indicators: According to [27], seasonal and trend decomposition based on loss (STL) suggests that any time series $X_t = \{x_1, x_2, \cdots, x_N\}$ can be decomposed in to three additive components: $X_t = T_t + S_t + E_t$, where T_t is the tendency component, S_T is the seasonal component, while E_t stands for residual component. To measure the trend and periodical behavior of the original series, we define the following indices respectively:

$$\begin{aligned} s_{deg} &= 1 - \frac{\text{var}(E_t)}{\text{var}(X_t - T_t)} \\ t_{deg} &= 1 - \frac{\text{var}(E_t)}{\text{var}(X_t - S_t)} \end{aligned} \tag{4}$$

- Skewness: The skewness is used to represent the heavy tail (asymmetric) phenomenon of a probability distribution. For a normal distribution, the skewness is equal to 0. In this perspective, the skewness can be used as a measure of non-Gaussian property. The skewness of the random variable X is defined as:

$$skew(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{5}$$

- Sample entropy: As stated in [28], sample entropy is a metric measuring the non linearity of time series. For a time series $X_t = \{x_1, x_2, \cdots, x_N\}$, we sample the original series into $N - m + 1$ segments with a template vector of length m defined as:

$$X_m(i) = \{x_i, x_{i+1}, \cdots, x_{i+m-1}\}, 1 \leq i \leq N - m + 1 \tag{6}$$

We further compute the distance between segments i, j, $i \neq j$ as:

$$d[X_{\mathrm{m}}(i), X_{\mathrm{m}}(j)] = \max_{k=0,\ldots,m-1} \|x_{i+k} - x_{j+k}\| \tag{7}$$

For a given threshold r, we count the number of segments pairs with $d[X_m(i), X_m(j)] < r$ as N_m, and the number of pairs with $d\left[X_{m+1}(i), X_{m+1}(j)\right] < r$ as N_{m+1}. For finite number N, the sample entropy is then calculated as:

$$SampEn = -\ln \frac{N_m}{N_{m+1}} \tag{8}$$

Considering the extreme few-shot scenario (i.e., 12 shots), where the total number of segments may be limited for large m, we directly set $m = 2$ and $r = 0.2 \times std(X_t)$.

- Hurst exponent: As a coefficient describing autocorrelation, Hurst exponent is a non linear metric for long term dependency of a sequence [29]. We denote the standardized series as

$$X^{'}(t) = \frac{X(t) - mean(X(t))}{std(X(t))}, \tag{9}$$

and calculate the cumulative sum sequence as

$$Y(t) = \sum\nolimits_{i}^{k=1} x_i^{'} \tag{10}$$

The Hurst exponent is then calculated as:

$$K = \frac{2}{N} \log(\max(Y(t)) - \min(Y(t))) \tag{11}$$

3) Ensemble clustering: :Note that clustering models such as K means, Gaussian mixture model (GMM-EM), etc. have high sensitivity to initial values. To acquire stable clustering results, we follow a clustering ensemble method based on hypergraph algorithm introduced in [30], namely clustering based similarity partition algorithm (CSPA). To ensemble the clustering results generated by multiple models and attempts, binary similarity matrices H are formulated to capture the pairwise similarity between clustering results, while co-association matrix are computed as $S = HH^T$. Then a hypergraph is generated based on co-association matrix, where vertex represents time series sample, and edge represents the similarity between objects. Finally, METIS [31] algorithm based on graph theory is used to obtain the final clustering results. The structure of ensemble clustering is visualized in Fig. 4.

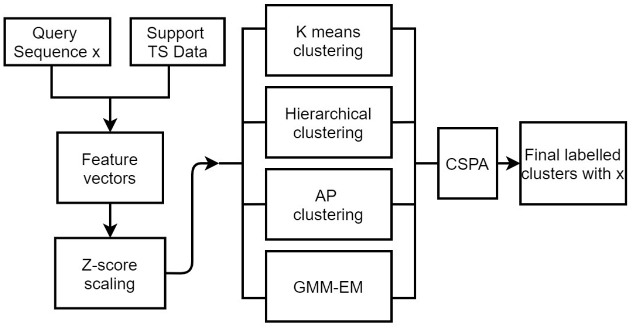

Fig. 4. Ensemble clustering

B. LSTM based Few-shot Forecasting

1) Wavelet denoising: To achieve FSL, we attempt to acquire prior knowledge about the characteristics of few-shot series, in order to generate a pre-trained model. By averaging all historical data from the clustering results, we obtain one sample series for each cluster, namely prototype series. The model then obtains a set of denoised prototype series and few-shot time series by performing DWT with hard threshold. The continuous wavelet transform (CWT) is given by:

$$H(x) = \frac{1}{|\sqrt{\zeta}|} \int x(t) \cdot \overline{\psi} \left(\frac{t - \tau}{\zeta} \right) dt \tag{12}$$

where signal $x(t)$ has a wavelet transform result as a function of time (t). ψ is a mother wavelet continuous in both time and frequency domain and $\overline{\psi}$ represents the complex conjugate of ψ. ζ is the scale parameter. τ is the transitional parameter. The DWT of the signal $x(t)$ is calculated by passing it through high and low pass filters. The decomposition of DWT is chosen to stop when the coefficients in the output are corrupted by edge effects caused by signal extension, where l_x is the length of signal and l_f is the length of filter.

$$level = \left\lfloor log_2 \left(\frac{l_x}{l_f} \right) \right\rfloor \tag{13}$$

The hard threshold is implemented with T denoted as the given threshold.

$$\rho_{\mathrm{T}}(x) = \begin{cases} x + T & x \leq -T \\ 0 & |x| \leq T, \\ x - T & x \geq T \end{cases} \tag{14}$$

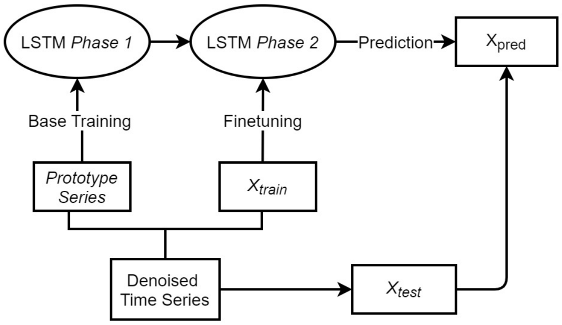

Fig. 5. Two-Phase LSTM

2) Two-phase LSTM: The model is designed to make full use of prior knowledge extracted from unsupervised ensemble clustering. Allocated in the same cluster c_a, a set of historical data $X^{S_1}, X^{S_2}..., X^{S_n}$ with abundant data points and few-shot time series $X^Q = (x_1, x_2..., x_m)$ share similar features that can be learnt as prior knowledge by two-phase LSTM (structure shown in Fig. 5).

- $Phase$ 1: The prototype series of historical data in c_a, X^c, is used to train the basic LSTM's network weights to θ_0, where the network possesses the ability of fast adaption to novel forecasting task in phase 2.

- $Phase$ 2: The few-shot time series X^Q are split into $(X^Q_{train}, X^Q_{test})$, where $\left|X^Q_{train}\right| \ll \min_i \left|X^{S_i}\right|$; X^Q_{train} fine-tunes θ_0 to θ_1; X^Q_{test} is used in the testing of few-shot task.

IV. Setup for Case Study

In this section, we introduce the performance metrics and overview the datasets for our case study.

A. FSL Task Formulation

The experiment tries to discover the performance of the proposed FSL under different levels of data shortage, namely trained with 12, 24, 48, 96, 192 shots of training data. For few-shot time series X^Q in k-shot learning scenario, $(x_1, ., x_k)$ is used in unsupervised clustering together with historical data. In two-phase LSTM fine-tuning, the prototype series of clustering results supports the base training of LSTM model. The k-shot data is used in the second phase to fine-tune LSTM. A fixed section of X^Q with length 72, $(x_{n+1}, ., x_{n+72}), n > k$ is used as ground truth in testing.

B. Metrics

Root Mean Square Error (RMSE) is one of the most used performance evaluation factors for forecasting or analyzing time series [10]. For n testing data, denote p_x as the ground truth and $\hat{p_x}$ as the corresponding forecast value, such that $x = 1$ to N, the RMSE is given as,

$$RMSE = \frac{1}{n}\sum_{x=1}^{n}\sqrt{(p_x - \hat{p_x})^2} \tag{15}$$

In our FSL settings, to describe model's overall performance of multiple predictions on different time series in c_a, Mean Root Mean Square Error (MRMSE) is introduced. For M time series, the MRMSE is given as,

$$MRMSE = \frac{1}{Mn}\sum_{i=1}^{M}\sum_{x=1}^{n}\sqrt{(p_{ix} - \hat{p_{ix}})^2}, i \in c_a \tag{16}$$

To eliminate outliers in our result, we cover the 95% confidence interval by adding or subtracting the MRMSE by two standard deviations and deleting values outside the interval. The mean and standard deviation of the remaining RMSE are then recalculated, and we use $MRMSE \pm std(RMSE)$ as our final metric to represent forecasting performance.

C. Training Details

The clustering model is trained on an ensemble clustering model consisting of K-means, GMM-EM, hierarchy clustering and affinity propagation, where the maximum level of DWPT $L = 5$. The following LSTM network applies Adam optimizer with 50 dimension of inner cells.

Firstly, an ablation experiment is conducted on the two real-world datasets in order to compare our model with traditional LSTM. Then, our model is applied on the synthetic dataset to verify a theoretical lower bound of shots. Lastly, we perform a sensitivity analysis on the proposed model based on the experiment, which investigates the influence of cluster compactness on forecast accuracy. For LSTM, we employ the same network structure as the two-phase LSTM in the proposed method, which adopts 50 units with Adam optimizer. All the tests are performed on a Linux server with an Intel Xeon E5-2620@2.10 GHz and 128GB of RAM.

D. UMass Smart Dataset

UMass Smart Dataset (2017 release) [32] includes minute-level electricity usage data from more than 400 anonymous homes. The dataset is sliced to have the time span from Jan. 1, 2016 to Mar. 10, 2016. During this period 114 homes' records are available. The granularity is set to be 20 minutes, 1 hour, 2 hours by averaging over data:

$$y_a[m] = \frac{1}{k}\sum_{i=mk}^{mk+k-1} m[i] \tag{17}$$

The FSL-LSTM is trained with 12, 24, 48, 96, 192 shots. A fixed section of X^Q with length of 72 is used for testing. Fig. 7 visualizes the UMass electricity load.

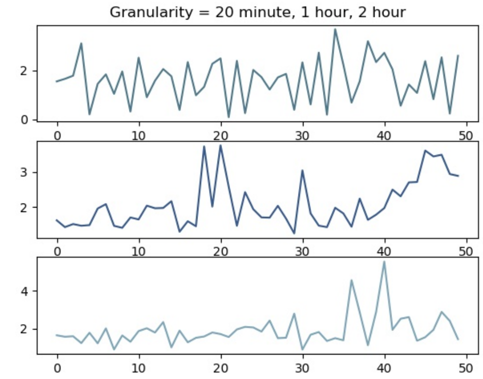

Fig. 6. UMass Smart Dataset with Different Granularity

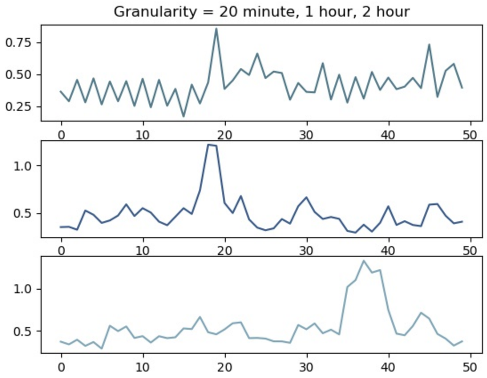

Fig. 7. Pecan Street Dataset with Different Granularity

TABLE I

COMPARISON OF PREDICTION ACCURACY ON UMASS SMART DATASET IN K-SHOT LEARNING SCENARIO

Dataset		Umass				
Granularity	Methods	12shot	24shot	48shot	96shot	192shot
20 minutes	FSL-LSTM(Ours)	0.883±0.317	0.999±0.366	0.959±0.296	0.931±0.336	1.004±0.377
	LSTM	1.177±0.483	1.240±0.443	1.499±0.211	1.096±0.340	1.004±0.339
1 hour	FSL-LSTM(Ours)	0.693±0.306	0.738±0.314	0.423±0.228	0.551±0.330	0.317±0.123
	LSTM	0.748±0.400	0.844±0.404	0.510±0.155	0.437±0.205	0.434±0.179
2 hours	FSL-LSTM(Ours)	0.528±0.233	0.339±0.179	0.347±0.146	0.308±0.086	0.308±0.197
	LSTM	0.695±0.352	0.283±0.161	0.754±0.277	0.335±0.127	0.321±0.144

TABLE II

COMPARISON OF PREDICTION ACCURACY ON PECAN STREET DATASET IN K-SHOT LEARNING SCENARIO

Dataset		Pecan Street				
Granularity	Methods	12shot	24shot	48shot	96shot	192shot
20 minutes	FSL-LSTM(Ours)	0.388±0.190	0.308±0.161	0.324±0.185	0.218±0.104	0.312±0.164
	LSTM	0.662±0.294	0.675±0.420	0.396±0.174	0.314±0.141	0.338±0.194
1 hour	FSL-LSTM(Ours)	0.418±0.187	0.363±0.154	0.512±0.188	0.515±0.198	0.466±0.314
	LSTM	0.624±0.287	0.521±0.230	0.670±0.212	0.543±0.214	0.593±0.282
2 hours	FSL-LSTM(Ours)	0.422±0.226	0.533±0.260	0.337±0.217	0.416±0.203	0.382±0.143
	LSTM	0.536±0.316	0.465±0.263	0.617±0.283	0.527±0.296	0.394±0.129

E. Pecan Street Texas Dataset

Pecan Street Dataport [33] includes minute-level electricity usage data from 310 units in Texas. The dataset is sliced to have a time interval from Jan. 1, 2016 to Mar. 10, 2016. The granularity is set to 20 minutes, 1 hour, 2 hours by averaging over data. The FSL-LSTM is trained with 12, 24, 48, 96, 192 shots. A fixed section of X^Q is used for testing. Fig. 8 shows the visualization of Pecan Street electricity load.

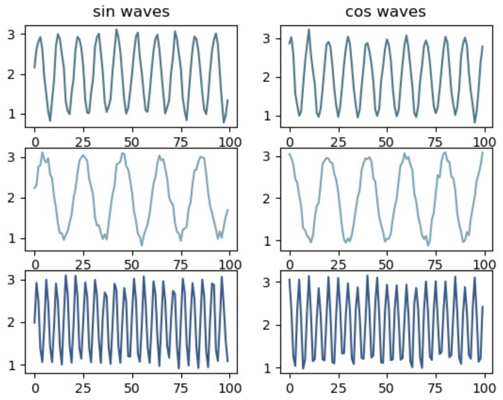

Fig. 8. Synthetic Dateset with Different Periods

F. Synthetic Dataset

Since the real-world power load data of users is not always based on a 24-hour cycle, we designed a synthetic dataset, which consists of sinusoidal waves where Gaussian noise is constructed to explore the influence of data cycle and training length on the model performance. The periods of time series are set to be 10, 15 and 20 sample points.

V. NUMERICAL RESULTS AND ANALYSIS

The experiment conducted on Pecan Street and Umass dataset suggests that FSL-LSTM outperforms traditional LSTM in most FSL scenarios. The detailed MRMSE results in Table I and II show significant improvements in precision and variance for forecasting 20 minute, 1 hour and 2 hour-level energy load in FSL. As shot length increases, the proposed method is followed more closely by traditional LSTM.

A. Influence of k Shot

To study the effect of number of training sample points for novel time series, we consider k = 12, 24, 48, 96, 192 for X^Q_{train} and measure the overall performance using MRMSE. The results are shown in Fig. 9 and Fig. 10.

When considering a very small number of training sample points in time series forecasting, e.g., for k = 12, we observe a large gap between the proposed method and traditional LSTM, yet FSL-LSTM also experiences a decrease in precision and variance. This is expected as given just 12 data points with granularity ranges from 20 minutes to 2 hours, it is difficult to learn the seasonality of power load of households or facilities with efficiency. The large gap in extreme few-shot scenario shows the high efficiency of FSL-LSTM to combine prior knowledge and specific k-shot samples during training.

B. Influence of Granularity

Since we extracted only short segments of sequence from historical dataset in order to match with the length and time stamps of few-shot time series, when the length of the few-shot series fails to cover a whole period, namely T, of the ground truth series, clustering results at first stage does not necessarily guarantee the following trends are similar to each other. Theoretically, to avoid mis-labelling, the length of few-shot series for fine-tuning, denoted by N, is expected to be $N \geq \frac{T}{M}$ for a fixed number of granularity M. This lower

TABLE III
COMPARISON OF FSL-LSTM PREDICTION ACCURACY WITH

k-shot	12	24	48	96	192	S-score
Kmeans	0.534±0.371	0.424±0.403	0.350±0.359	0.345±0.314	0.336±0.282	0.1385
Agglomerative	0.509±0.175	0.414±0.307	0.338±0.274	0.326±0.178	0.315±0.286	0.3843
GMM-EM	0.522±0.427	0.426±0.311	0.344±0.328	0.347±0.266	0.354±0.185	0.1230
Affinity Propagation	0.519±0.293	0.426±0.276	0.345±0.326	0.345±0.314	0.345±0.271	0.2138
Ensembling	0.500±0.242	0.407±0.143	0.314±0.172	0.306±0.220	0.316±0.197	0.3622

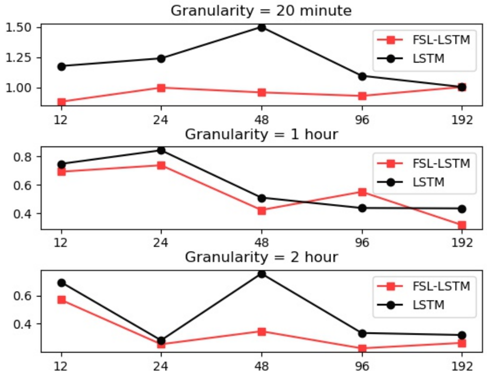

Fig. 9. Case 1: Normalized Quantile Loss on Umass Dataset

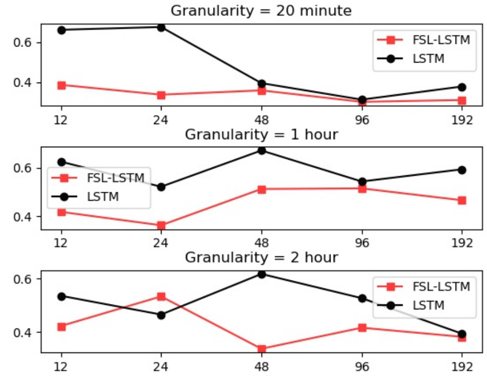

Fig. 10. Case 2: Normalized Quantile Loss on Pecan Street Dataset

bound is particularly phenomenal in our synthetic dataset, while not violating the observation in real-world dataset.

When granularity is small, the ideal length of few-shot samples that yield acceptable MRMSE is significantly larger than those of the large ones. Furthermore, the granularity and few-shot length pairs reach the most benign model performance when their products fully contain one or multiple periods of the historical dataset. This phenomenon is much more significant on our synthetic dataset. As shown in Fig. 12, the model reaches the lowest MRMSE when $N = PT/M$, where P denotes any positive integer. The MRMSE then remains relatively steady after N reaching the threshold, which means that our theoretical assumptions do not violate empirical observation.

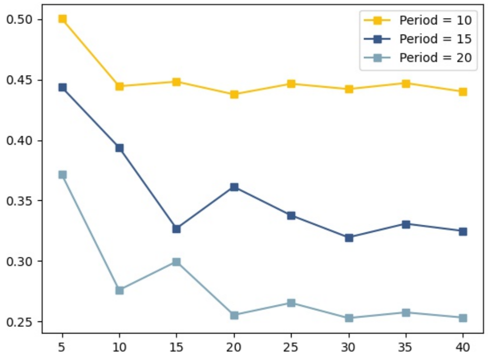

Fig. 11. Case 3: Normalized Quantile Loss on Synthetic Dataset

C. Influence of Cluster Compactness

As an FSL forecasting model, the prediction accuracy of the fine-tuned model depends on the quality of the prior knowledge. One rational intuition is that the compactness of clustering results is positively correlation with MRMSE. To investigate the hypothesis, we conduct single factor sensitivity analysis by changing different clustering models on UMass Smart dataset with 1 hour granularity. To quantify the compactness of clusters, Silhouette score (S-score) is introduced. The results are shown in Fig. 13.

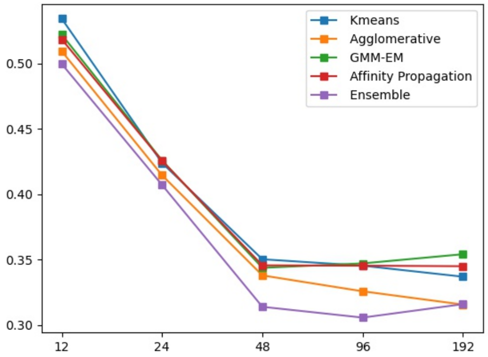

Fig. 12. Case 4: Normalized Quantile Loss for Different Clustering Models

Table III suggests that the S-score of ensemble clustering is higher than those of traditional clustering models due to elimination of some edge samples. Moreover, the standard deviation of RMSE has a negative correlation with S-score. This means that the larger the S-score, the more likely the denoised prototype can capture most of local features inside the cluster. In addition, MRMSE reduces slightly when S-score improves. However, the difference of MRMSE for a fixed shot between different clustering models is not significant.

VI. Conclusion and Future work

The ability to quickly adapt to time series forecasting tasks with limited customized samples is an important property for electricity load forecasting and other practical applications. We contribute to this field by proposing the FSL time series forecasting based on LSTM. The proposed method leverages the existing power load records through ensemble clustering to gather an ability to efficiently solve few-shot forecasting tasks on previously unseen time series. Numerous studies suggest that, the proposed method is able to largely outperform its baseline on 2 major electricity load datasets. Moreover, we empirically interpret FSL-LSTM's performance from two aspects, k-shot setting and granularity of data.

In the future, it would be interesting to explore more sophisticated few-shot learning techniques such as [22], [34] for load forecasting. Besides, by combining FSL with incremental learning [35], a robust AI blue print can be provided to power grid system, such that models can be swiftly generated through FSL when data scale is small, and be fine-tuned locally as data scale increases.

References

[1] P. Coulibaly and C. K. Baldwin, "Nonstationary hydrological time series forecasting using nonlinear dynamic methods," *Journal of Hydrology*, vol. 307, no. 1-4, pp. 164–174, 2005.

[2] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *stat*, vol. 1050, p. 16, 2017.

[3] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*. IEEE, 1999, pp. 4–9.

[4] G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, 2016, pp. 1–8.

[5] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via arma model identification including non-gaussian process considerations," *IEEE Transactions on power systems*, vol. 18, no. 2, pp. 673–679, 2003.

[6] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE transactions on power systems*, vol. 18, no. 3, pp. 1014–1020, 2003.

[7] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy load forecasting using empirical mode decomposition and support vector regression," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 549–556, 2013.

[8] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.

[9] Y. Liu, W. Wang, and N. Ghadimi, "Electricity load forecasting by an improved forecast engine for building level consumers," *Energy*, vol. 139, 07 2017.

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

[11] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[12] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018.

[13] Y. Wang, D. Gan, M. Sun, N. Zhang, C. Kang, and l. Zongxiang, "Probabilistic individual load forecasting using pinball loss guided lstm," *Applied Energy*, vol. 235, pp. 10–20, 02 2019.

[14] S. Rani and G. Sikka, "Recent techniques of clustering of time series data: a survey," *International Journal of Computer Applications*, vol. 52, no. 15, 2012.

[15] D. T. Anh and L. H. Thanh, "An efficient implementation of k-means clustering for time series data with dtw distance," *International Journal of Business Intelligence and Data Mining*, vol. 10, no. 3, pp. 213–232, 2015.

[16] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International work-conference on artificial neural networks*. Springer, 2005, pp. 758–770.

[17] V. Bettaiah and H. S. Ranganath, "An analysis of time series representation methods: data mining applications perspective," in *Proceedings of the 2014 ACM Southeast Regional Conference*, 2014, pp. 1–6.

[18] F. Mörchen, "Time series feature extraction for data mining using dwt and dft," 2003.

[19] A. Hacine-Gharbi and P. Ravier, "Wavelet cepstral coefficients for electrical appliances identification using hidden markov models." in *ICPRAM*, 2018, pp. 541–549.

[20] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[21] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[22] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[23] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[24] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta r-cnn: Towards general solver for instance-level low-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9577–9586.

[25] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8420–8429.

[26] E. Pavez and J. F. Silva, "Analysis and design of wavelet-packet cepstral coefficients for automatic speech recognition," *Speech Communication*, vol. 54, no. 6, pp. 814–835, 2012.

[27] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[28] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[29] R. Weron, "Estimating long-range dependence: finite sample properties and confidence intervals," *Physica A: Statistical Mechanics and its Applications*, vol. 312, no. 1-2, pp. 285–299, 2002.

[30] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

[31] G. KARYPIS, "Metis, a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices version 4.0," *http://glaros. dtc. umn. edu/gkhome/metis/metis/download*, 1997.

[32] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, J. Albrecht *et al.*, "Smart*: An open data set and tools for enabling research in sustainable homes," *SustKDD, August*, vol. 111, no. 112, p. 108, 2012.

[33] P. Street, "Pecan street dataport," Website, 2016, https://dataport.pecanstreet.org.

[34] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1842–1850.

[35] S. W. Yoon, D.-Y. Kim, J. Seo, and J. Moon, "Xtarnet: Learning to extract task-adaptive representation for incremental few-shot learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 852–10 860.