Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

MF-GCN-LSTM: a cloud-edge distributed framework for key positions prediction in grid projects

Shaoyuan Huang[1], Yuxi Zhang[2], Guozheng Peng[2], Juan Zhao[3], Keping Zhu[4], Heng Zhang[1] and Xiaofei Wang[1*]

Abstract

In this article, we solve the key positions prediction problem of engineering projects in smart grid, which pays more attention to the spatial-temporal distribution of projects. Many studies show that the projects are affected by multi-dimensional features such as time, space, correlation etc. However, few work can accurately predict the key positions of projects based on multi-dimensional features. In order to solve this problem, we propose the idea of multi-feature extraction, and make use of the real-world records trace to conduct multi-dimensional modeling. Then we introduce a multi-dimensional features extraction model: Multi-Feature-based GCN-LSTM (MF-GCN-LSTM) to take the effect of time, space and correlation for predicting the key positions of projects. Experiments on different datasets with various project types have proved that our model can complete the key positions prediction task efficiently. Compared with the other traditional method and non-linear models, our model shows higher prediction accuracy and robustness. Moreover, we show that the whole prediction framework MF-GCN-LSTM can be split and deployed in a distributed manner to accelerate the inference of the model under the cloud edge system.

Keywords: Key Positions Prediction, Projects Networks, Multi-Dimensional Feature Extraction, Graph Convolutional Networks

Introduction

The development and services of the grid are based on the implementation of various grid projects, e.g., the construction of a new power station in an urban area. In recent years, the management of grid projects is gradually moving from the rough-and-tumble development to refined management, where management based on manual decision and expert experience can no longer handle the entire cluster of projects with huge numbers. Introducing data analysis and machine-aided decision making into grid project management becoming an important way to reduce non-essential grid projects and improve project management effectiveness.

The analysis and prediction of the regional density of grid projects is an effective way to discover the essential projects and to assist the grid in scientific project management. In the cluster of grid projects, each project has real location attributes and different types of projects are interrelated, influenced by this, certain urban locations will concurrently propose a large number of similar projects, i.e., grid projects key position.

In this paper, we aim to predict the key positions of grid project accurately. Specifically, as shown in Fig. 1, the grid projects will be distributed throughout the city, as projects iterate, the spatial and temporal distribution of new projects will evolve, so that certain areas in the city will become key positions of concentrated project distribution (i.e., the red area in Fig. 1). Since information about the time and location of these projects from start to finish can be continuously recorded, the algorithms we deploy on the core cloud and edge servers can process

*Correspondence: xiaofeiwang@tju.edu.cn

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
Full list of author information is available at the end of the article

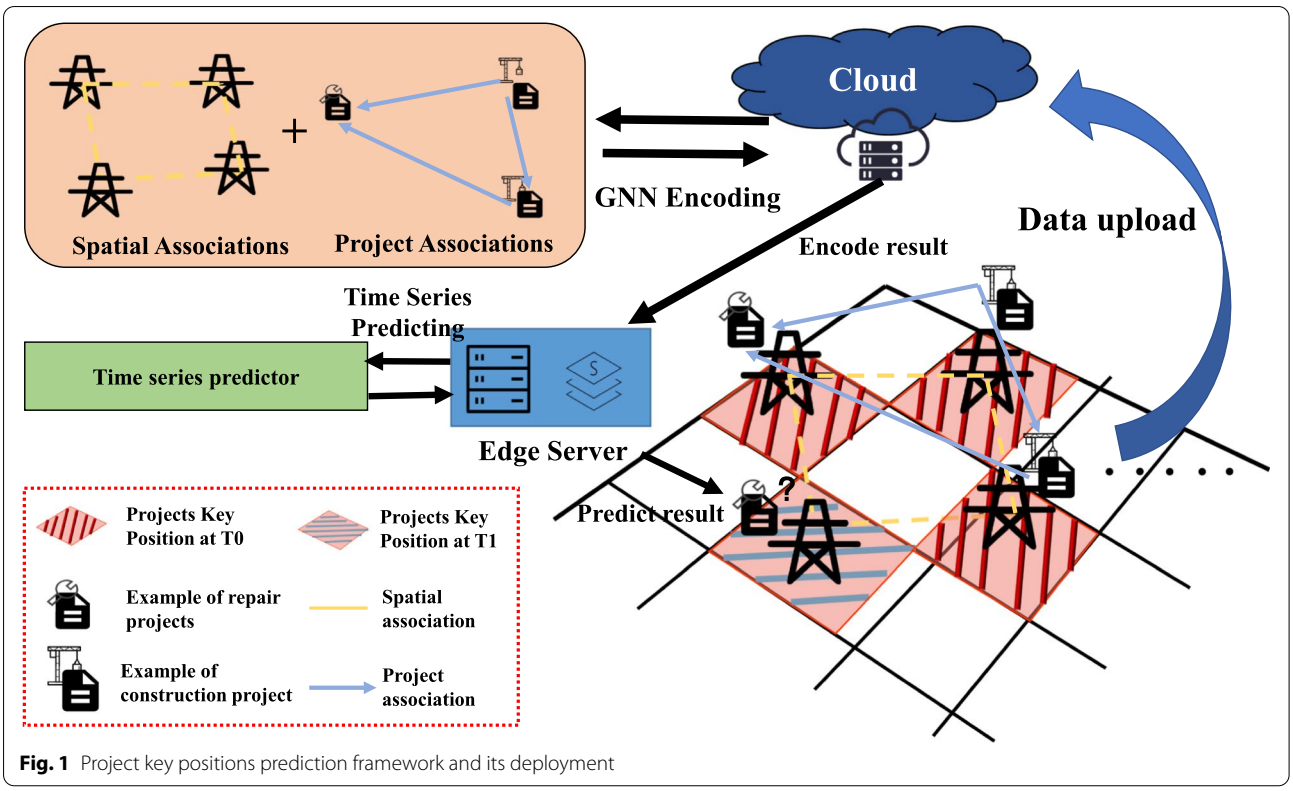


Fig. 1 Project key positions prediction framework and its deployment

the recorded data and predict what areas will be key positions for grid projects.

There have been many researches related to the project management optimization, such as project link modeling and prediction [1–3], project category modeling [4], project path optimization [5] etc. However, these researches still lack thinking about the spatial-temporal distribution of projects. Grid projects' spatial-temporal distribution is a necessary attribute to consider for project management. Especially, considering the complexity and specificity of grid projects, there are two major challenges in the prediction of project key positions:

- **Data abstraction method.** Due to the lack of location data and the limitation of prediction models, it's difficult for existing data modeling methods to simultaneously model the geographical distribution and project correlation of project transmission under multiple time slices [6]. Therefore, we propose a regional spatial-temporal modeling algorithm to model the features of space, time and correlation in smart grid.
- **Multi-feature extraction.** Studies [7, 8] have shown that the projects is closely related to the spatial, temporal and correlation features. Models that concern single feature can hardly achieve the best performance. To maximize the prediction ability of the model, we propose the spatial-temporal-correlation multi-feature extraction to solve the key positions prediction task.

To solve the above problems, we first propose a regional spatial-temporal modeling method to abstract grid project data into project networks. Besides, we propose a multi-feature fusion model: Multi-Feature-based GCN-LSTM (MF-GCN-LSTM), which can predict the key positions of grid projects based on the learning of project networks.

The MF-GCN-LSTM is consist of a GNN novel customized encoder and a time series predictor. The GNN encoder is designed to capture spatial features of grid projects based on the inter-project dependencies, and encode them to vector representation. After that, the learned network representation are input to the time series predictor e.g., long-short time memory network (LSTM) to capture the time-dependency of projects.

Besides, we show that the whole prediction framework MF-GCN-LSTM can be split to accelerate the inference of the model under the cloud edge system, we can first place MF-GCN-LSTM in the cloud to complete the training, and then keep GCN as an encoder that can be further trained in the cloud, while the LSTM can be decentralized to the edge server to perform temporal inference to complete the prediction.

This combination gives MF-GCN-LSTM the ability to process complex features in the power grid system. Besides, the deployment splitting makes the application of the framework more flexible and increases the speed of project optimization at the edge measurement with guaranteed accuracy. In summary, this paper makes the following contributions:

- In this work, we propose and formalize a novel problem of key positions prediction using realistic large-scale projects data in smart grid. As far as we know, this is the first attempt to predict the projects key positions distribution in real-world.
- We firstly combine LSTM with GCN to predict the projects key positions in smart grid. The prediction model can simultaneously extract the spatial-temporal-correlation feature, with high accuracy and stability. Besides, we propose a regional spatial-temporal modeling algorithm to simultaneously encode the multi-dimensional features in smart grid.
- To verify the generalization ability and inference efficiency of our framework, we conduct extensive experiments on different datasets in smart grid, the results demonstrate that our model can significantly improve the effectiveness of prediction in terms of precision and recall metrics, i.e., promoting the precision by at most 1.18 times and the recall score by at most 0.70 times.

The rest of this paper is organized as follows. In Section II, we briefly introduce the related work. A formal definition of the prediction problem is given in Section III. Section IV is a preliminary data analysis of two data sets. Section V presents the proposed MF-GCN-LSTM model in details. In Section VI, we describe the experiments, including the performance evaluation and a case study of the proposed model's prediction effect. We finally conclude this paper in Section VII.

Related work

In this section, we review previous approaches regarding key positions prediction in smart grid and other related fields. Moreover, we review these prediction methods from single to multidimensional features based on the technical point of view.

Key positions prediction problem

A large number of efforts have explored projects prediction, such as link prediction [9–11], project activity prediction [12–14], community and news detection [14, 15]. These works focus more on the topological structure or attributes of the network, pay less attention to the prediction of real-world features of projects like geographical distribution.

Mededovic et al. [16] and Scellato et al. [7] both model and analyze the key positions of projects in real geographic space. However, these works only concern about identifying key positions, but do not propose the key positions prediction model.

There are some spatial-temporal analysis and prediction methods in position recommendation and traffic fields, Liu et al. [17] proposed a two stage destination prediction framework of shared bicycle based on geographical position recommendation, but it use feature engineering method to process the dataset and this method requires abundant expertise knowledge and cannot be easily transferred to other areas. Jianmei et al. [18] proposed a bus arrival time prediction method with time, gps position and traffic flow feature, but this method analyze the bus with fixed routine, cannot be applied to key positions prediction.

Single Feature Extraction Model

Similar to conventional spatial-temporal forecasting problems, prediction of grid projects also relies on the extraction of key impact features (e.g., temporal, spatial, and project associations). Traditional methods like [8, 19, 20], rely on pre-data law analysis, and feature selection in the model requires certain domain expertise, which leads to poor generalization ability of the model.

Machine learning methods are widely used in project management. Zhang et al. [21] proposed a investment probabilistic interval estimation model with the hybrid model of SVR and GWO. Mart' et al. [22] proposed a prediction model for the software project construction phase with SVR. Huang et al. [23] use a random forest and simple linear regression model to predicting BIM labor cost. These methods analyze the single project features with certain time, but ignore the relationship among projects and the geographical features. In fact, in the field of smart grid, there is a strong spatial correlation and continuity between projects, which cannot be accurately analyzed and predicted using traditional machine learning algorithms.

Some researches put forward non-linear neural network model to realize feature extraction and projects prediction of large amount of data [24–26]. Zhang et al. [24] describe a method to predict the link in a network with GNN, but it only use the relationship in the network. Huo et al. [25] propose a method to predict the link in network with personalized social influence. Li et al. [26] propose a method to generate security guaranteed image watermark. These studies improve the prediction accuracy and the generalization ability of models. However, the neural network models are often unable to capture multi-dimensional features

such as space, time dependency and correlation structure at the same time. Moreover, within narrow constraints or even complete absence of spatial attributes, the representative ability of these networks would be hindered seriously.

Multi-Feature Extraction Model

To further improve the accuracy and robustness of prediction for grid projects, there are some studies based on multi-dimensional methods in smart grid. Alazab et al. [27] propose a multi-dimensional LSTM model to predict the stability of smart grid. In [28], Yu et al. propose a spatial-temporal convolution model of GNN and GLU module, which captures the spatial-temporal correlation while improving the training speed. In [29], Geng et al. propose a Multi-Graph Convolution Network ST-MGCN, to encode the non-Euclidean pair-wise correlations among regions.

Although these methods effectively exploit the correlation between entities, these studies are all based on the traffic network, whose network structure is generally fixed, which is quite different from the flexible project network.

In projects networks prediction, Wang et al. present a hybrid deep learning model for spatial-temporal prediction, which includes an auto encoder-based deep model for spatial modeling and Long Short-Term Memory units (LSTMs) for temporal modeling [30]. In another article, a non-linear model GCN-GAN was proposed to improve the time prediction performance of links by combining the dynamics, topology structure and evolutionary model of Weighted Dynamic Networks [1]. These models focus on the temporal and spatial features of the projects network, but do not care about the projects and the correlation. Compared to these jobs, our work pays more attention to the complex spatial-temporal-correlation features in projects.

Problem definition

Firstly, we define the **key position** in our prediction problem.

Using the Geohash coding algorithm, we divide the areas covered by all the power grid projects into several judgment units of equal size, each of which will be further divided into λ regions. In a certain time window t, if the number of same grid project's raise in a region is greater than a threshold τ, the region is marked as a key position for this type of project. We use a binary matrix M to represent the distribution of key positions in a judgment unit.

$$M[x][y] = \begin{cases} 0, & num\_nodes[x][y] < \tau, \\ 1, & num\_nodes[x][y] \geq \tau. \end{cases} \tag{1}$$

We use $G = \{G_0, G_1, ..., G_\kappa\}$, i.e., a sequence of graph snapshots to represent the projects network in smart grid of time windows $(t_0, t_1, ...t_\kappa)$. Each snapshot $G_t = (V_t, E_t)$ has different node set V and edge set E, each node in V represents a record of project, and each edge in E represents the friendship between projects.

The grid projects are generally interdependent with each other within an order ways, for instance, maintenance projects are often the successors of an establish project, and cultivation projects are the predecessors of most establish projects. The correlation of these projects is closely related to the project categories, and the backward and forward correlation between large number of projects can be obtained automatically by extracting the project categories and constructing correlation rule sets.

Considering the order of projects' raise, we set the edges in E as directed edge. After that, we generate a geographic distribution matrix $W_t \in \mathbb{R}^{|V| \times \lambda}$ for snapshot of time t according to the projects' location attribute.

For any given judgment unit, with the graph snapshots sequence $G = \{G_{\kappa-T+1}, ..., G_\kappa\}$ and geographic distribution feature matrix $W = \{W_{\kappa-T+1}, ..., W_\kappa\}$ of T time windows, the goal of the key positions prediction is to predict the key positions matrix M of the time window $(\kappa + 1)$:

$$[G_{\kappa-T+1} W_{\kappa-T+1}, ..., G_\kappa W_\kappa] \xrightarrow{f(\cdot)} M_{\kappa+1} \tag{2}$$

where $f(\cdot)$ is the model that we need to construct in this paper while $M_{\kappa+1}$ represents the prediction result. After model training and fitting, our target is to minimize the difference between prediction key positions matrices and the ground-truth.

Methodology

Regional spatial-temporal modeling for gird projects data abstraction

In order to maximize the retention of the multi-dimensional features of projects in smart grid, we have defined the data structure of Regional Spatial-Temporal Graph (for the convenience of discussion, we call it as spatial-temporal graph later) to reflect the trend of the geographical distribution of projects.

As discussed in section 3, the spatial-temporal graph is not a topological graph, but a sequence of graphs consisting of a certain number of static raise graphs. In the construction process, we first establish a complete raise graph G_c for a certain type of project under all observation windows $(t_0 - t_\kappa)$, and then perform sub-graph extraction on this basis to generate spatial-temporal graphs.

After merging the raise records of the same project category (will be further discussed in section. 5), we establish a complete raise graph of this project category. Firstly, we set a node for each projects record, then we

establish the edge between nodes according to the friendship between projects who participated in the twice projects (node). Since there is no explicit friendship in some records, we consider the two projects who have shared file in a PFR transmission record as friends.

The complete raise graph can't reflect the changes of the projects over time, at the same time, the complete raise graph covers the global geographical scope, which is not favorable for us to extract the interaction between nodes and predict the final key positions.

For this reason, we divide the whole time slice of projects into several $(\kappa + 1)$ equal segments (the observation time window length of each segment was about one week) and reduce the geographical scope of the graph by dividing the key positions judgment units.

We obtain the spatial-temporal graph from the complete raise graph by selecting the central node for region focusing and time division.

MF-GCN-LSTM for multi-feature extraction

In order to illustrate the impact of different features on projects in smart grid, we established the Static GCN, Time Series LSTM and our Multi-Feature-based GCN-LSTM (MF-GCN-LSTM) model on the basis of above two models to solve the the projects.

Static GCN for Spatial Feature Extracting

The Static GCN model is designed to learn the complete raise graph G_C under all time windows $(t_0 - t_\kappa)$. We use a spatial-based method to establish the Static GCN, that is, the GCN layer is aimed to broadcast, aggregate, and update nodes' location attribute through edge relationship between projects in the graph until it is stable, the process of node attribute updating can be briefly described as follows:

$$Node_v^{l+1} = \sum_{\forall u \ edge_{uv} \in E} Node_u^l \tag{3}$$

where $Node_v^l$ represents the attribute vector of $Node_v$ in the l-th layer, $\forall u \ edge_{uv}$ represents all nodes that have an directed edge points to v. Thus, each node in the graph will update its spatial distribution feature under the influence of its neighbors that point to it, i.e. the future features of an grid project are influenced by its predecessors. The greater the entry of a node, the more predecessors the project has, and the more information the node updates to refer to, and conversely the more stable the project is.

Static GCN has a strong feature-extraction capability for static projects networks. Through the information broadcast and update between nodes, it can aggregate the location information of the nodes and characterize the overall structure of the network. Here we use it to extract the spatial feature based on the grid projects networks in smart grid.

We have established a GCN model with two convolutional layers and an average-pooling layer. The specific model structure is shown in the Fig. 2.

We input complete raise graph and the geographic distribution W into GCN for feature extraction and synthesis. At the end of each forward propagation of GCN, the average-pooling algorithm $v_G = \frac{1}{n}\Sigma_i^n v_i$ is applied to transform the embedded information of all nodes in the graph into the representation of the entire graph. In this way, each graph will eventually get a unique embedding vector to represent the graph structure and the overall features of the nodes. The layer-wise propagation rule in vector form can be defined as:

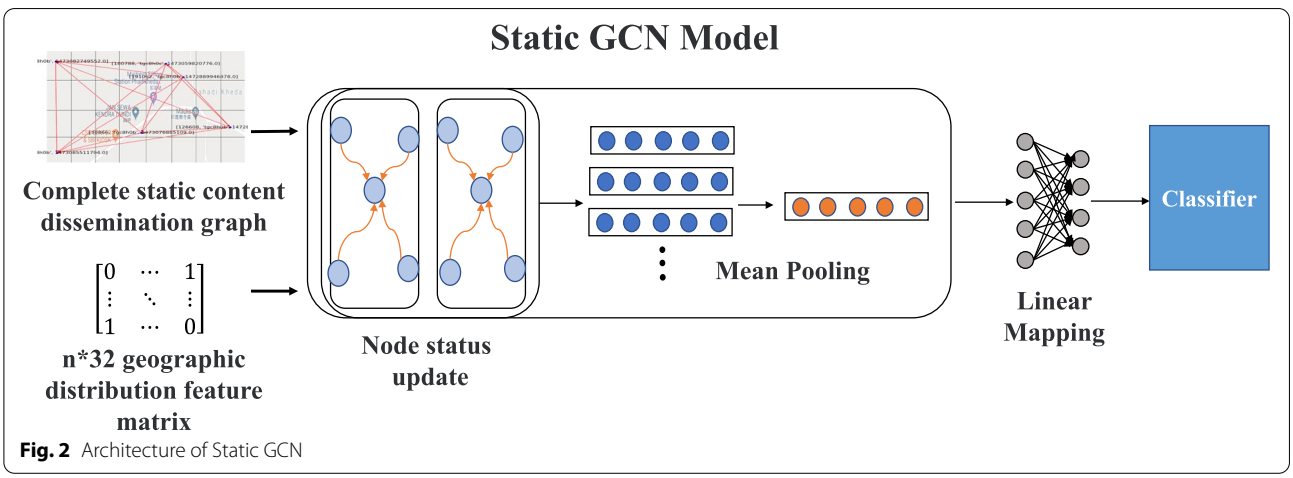


Fig. 2 Architecture of Static GCN

$$h_i^{l+1} = \sigma(\sum_j w_{ji} h_j^l W^l) \tag{4}$$

where h_i is the feature vector of node i, j are node i's neighbors that points to it, w_{ji} is the connection relationship between j and i, W^l is the the weight matrix of the l-th layer and σ is a non-linear activation layer.

At the end of the model, we use the *Sigmoid* activation layer to map the vector representation of the graph (v_G) to the key position probability matrix $\hat{y}$. This process can be formulated as Eq. 5.

$$\hat{y} = sigmoid(W_c v_G + b_c) \tag{5}$$

Time Series LSTM Model for Time Feature Extracting

The raise of grid projects are often periodic. For instance, inspection-type projects are often proposed after construction-type projects and will occur several times on a periodic basis. This temporal dependence leads to the relevance of project networks under different time, and GCN cannot capture this distinction and connection.

To extract the time feature of grid projects, we adopt the Time Series LSTM, which is used to capture the long-term time-dependency of regional projects in different time windows. The compute process for each layer in LSTM can be described as follows [31]:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_t - 1, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_t - 1, x_t] + b_i) \\ c_t &= f_t * c_t - 1 + i_t * tanh(W_c \cdot [h_t - 1, x_t] + b_c) \\ i_o &= \sigma(W_f \cdot [h_t - 1, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{6}$$

where t is the time step in terms of days. h_t, c_t, x_t are the hidden state, cell state, and previous layer hidden state at time t. f_t, i_t, o_t are respectively the forget gate, input gate and output gate.

LSTM calculates the parameters law of the input sequential data, adjusts the ratio of memory and forget through the input gate and forget gate to obtain the hidden state of the next layer. During this period, LSTM can perform parameter optimize through back raise to determine the impact of historical information in different time periods.

In order to use LSTM model to predict the projects key positions, we refer to the methods of traditional correlations to discretize the projects in smart grid [32]. For the static graph under each observation window, we generate a key position matrix M. In this way, the sequence type graph G is converted into sequential matrices, then we input it into the LSTM model to extract its time-dependency feature. Finally, we add one sigmoid layer to map the final output of LSTM to the key position prediction matrix in $t_{\kappa+1}$. The prediction process of Time Series LSTM can be formally described as:

$$[M_0, M_1 ....., M_\kappa] \xrightarrow{\mathcal{L}(\cdot)} M_{\kappa+1} \tag{7}$$

where M_t is the key positions distribution matrix in time window t and $\mathcal{L}(\cdot)$ represents the fitting calculation of LSTM.

LSTM can successfully capture the time-dependency of regional projects. We further combined it with other features to design a more robust prediction model.

The Multi-Feature-based GCN-LSTM

With the aforementioned feature extraction models, we finally introduce our Multi-Feature-based GCN-LSTM model (MF-GCN-LSTM), which is used to simultaneously extract the spatial-temporal features of the projects in smart grid. With combining of the topological graph representation and the time-dependency capturing of sequential data. The system structure of the MF-GCN-LSTM is shown in Fig. 3.

To fuse GCN with LSTM, we first remove the classifier of the static GCN, however, this is far from enough. The fusion of GCN with LSTM is not a simple mapping of model output to input, the static GCN model encodes graph based on the stable network structure, it assumes fixed structure among networks and computes the graph structure feature once and set the adjacency matrix as constant coefficients. Therefore it is not suitable for encoding dynamic spatial-temporal graphs.

To solve the problem, we reconstruct the static GCN. Referring to the inductive learning of GraphSage [33], we design the node information transfer and aggregation process of GCN as a nonlinear function with learnable parameters, and adjust the information update among nodes by the change of parameters. In this way, GNN can learn the inter-influence property between nodes based on the projects networks under different time windows, and thus encode the heterogeneous spatial-temporal graphs in a uniform way.

The information update process of GCN for a single node in MF-GCN-LSTM is as follows:

$$\begin{aligned} m_{ij}^l &= W_\phi(h_j^l) + b_\phi \\ h_i'^l &= W_\theta(h_i^l) + b_\theta \\ h_i^{l+1} &= \max_{j \in \mathcal{N}(i)} \mathrm{ReLU}(m_{ij}^l + h_i'^l) \end{aligned} \tag{8}$$

where m_{ij} represents the information passed from node j to node i, $h_i'^l$ represents the information update of node i and $h_i^{(l+1)}$ represents the final feature of node i after one round information update. The learnable parameters W_ϕ and W_θ enable the GCN to adaptively adjust the update

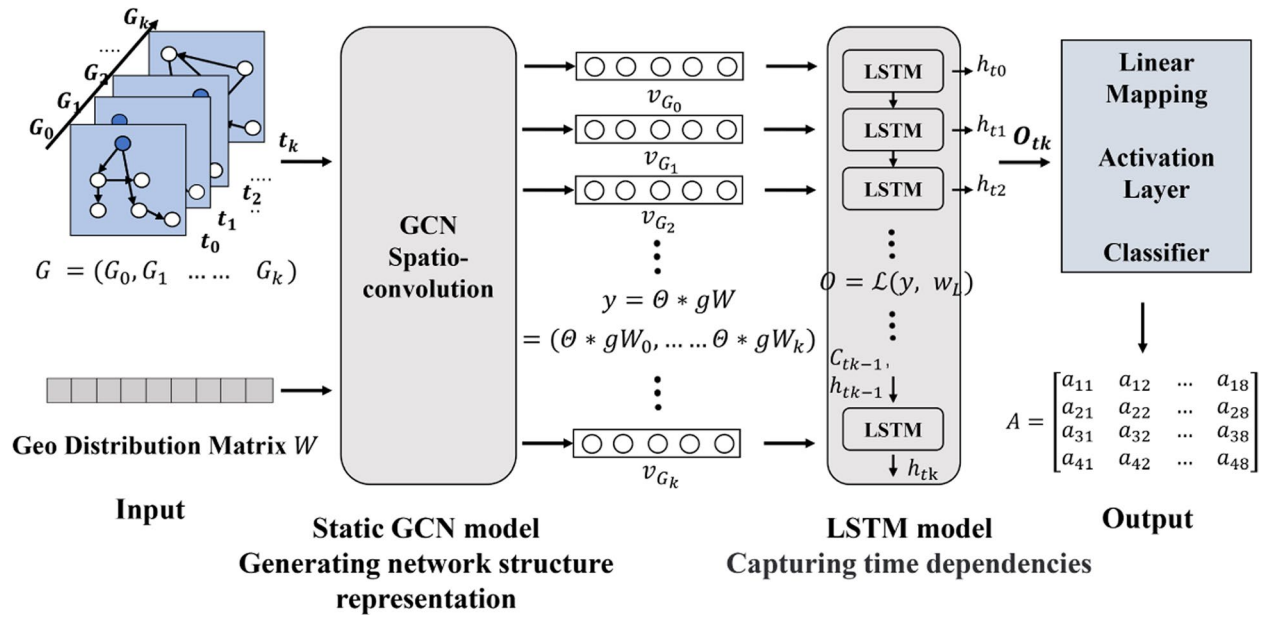


Fig. 3 Architecture of the Multi-Feature-based GCN-LSTM model

rules of the nodes, thus eliminating the need to memorize the fixed structure of the network. Therefore, the GCN in MF-GCN-LSTM can handle dynamic projects networks.

The spatial-temporal graph G and the geographic distribution feature matrix W are used as the total input, the GCN model is used to encode each static graph G_t in spatial-temporal graph, capture the correlation topological structure feature and update the spatial feature of nodes through the spatial-based graph convolution, then the learned network representation $v_G = (v_{G_0}, v_{G_1}, ..., v_{G_\kappa})$ is input to the LSTM network to capture the time-dependency feature of the projects in smart grid. Finally, the output of the LSTM's last hidden state is put into activation layer and classifier to generate the final key position prediction matrix.

We use $*g$ to represent the spatial-based convolution operation, and $\mathcal{L}$ to represent the compute process of the LSTM neural network for forward propagation. The following Eq. 9 is the formal process of the overall calculation of the model: Θ represents the convolution kernel of the graph convolution (whose parameters like the receptive field and weight matrix are modified according to the input network), W represents the geographic distribution matrix while W_L represents the parameter matrix of the LSTM neural network. Therefore, the calculation process of MF-GCN-LSTM can be formulated as:

$$\begin{aligned} y =& \Theta * gW = (\Theta * gW_0, \Theta * gW_1, .., \Theta * gW_k) \\ O =& \mathcal{L}(y, W_L) \\ \hat{y} =& sigmoid(W_c O_i + b_c) \end{aligned} \tag{9}$$

Since the problem of key positions determination is essentially a binary classification problem, the above three models all use the BCELoss function as the loss function. The BCELoss function calculates the difference between each predicted key position matrix and the real key position matrix.

$$\begin{aligned} &loss(pre(G), M) = BCELoss(pre(G), M) \\ &= -w[Mlog(pre(G)) + (1 - M)log(1 - pre(G))] \end{aligned} \tag{10}$$

During the training process, we use the image batch function of the DGL library to aggregate all graphs and sequential matrix data into a large dataset.

Preliminary data analysis

We use two typical real-world large-scale datasets in smart grid: project file records (PFR) and project places data (PPD).

In order to solve the key positions prediction problem concretely, we use Geohash encoding algorithm to encode the GPS information of all data. Geohash was proposed by Gustavo Niemeyer, which is used to encode a geographic location into a short string of letters and digits [34]. Geohash use Base-32 alphabet encoding (characters can be 0 to 9 and A to Z, excl "A", "I", "L"

and "O"). Using Geohash to establish spatial index can improve the efficiency of latitude and longitude inspection of data.

According to the actual precision of GPS information, the encoding length of Geohash and the error meter relationship, We set the radius of key positions judgment unit to 610m (which is the region represented by 6-bit geohash code, close to the coverage of the city base station), then we can divide the judgment unit into $32(\lambda = 32)$ equal size regions (7-bit geohash code), so that the regions' key positions prediction results in a judgment unit can be directly applied to the project caching strategy based on the city base station.

Affected by project interest preference and project popularity, different types of project have great differences in the spatial-temporal law of raise. Therefore, we choose to classify the projects in the dataset of PFR and PPD before prediction.

PPD provides category for each check-in place, and the transfer records in the PFR only provide the name of the project. Thus, we classify and merge them by the category statistics of Google project store.

Experimental evaluation

Dataset and Experiment Setup

We selected 4 different categories of project file under PFR dataset and 3 categories of check-in place under PPD dataset for experiment. After data merging and spatial-temporal modeling, there are 120140 nodes and 168340 edges in all spatial-temporal graphs under PFR, and 230996 nodes and 491752 edges in all spatial-temporal graphs under PPD. Table 1 shows some key parameters in the data after spatial-temporal modeling:

where C represents the type of raise project, G represents the number of spatial-temporal graphs, N represents the average node number of a spatial-temporal graph in a judgment unit, THR represents the key position threshold, whose size varies with the change of N, and hot/all is the proportion of key positions in all regions.

Table 1 Information about chosen datasets

DataSets	C	G	N	THR	hot/all
PFR	PFR0	1244	17	3	0.17%
	PFR1	710	26	5	0.42%
	PFR2	2292	26	5	0.22%
	PFR3	1047	20	3	0.27%
PPD	PPD0	766	98	3	7.3%
	PPD1	883	88	3	3.9%
	PPD2	687	123	5	15.3%

In the later experiments, we can see that the baseline models have a great difference in the prediction performance of different project modes, but our model has always maintained a high prediction accuracy, which confirms the stability of the multi-feature-based prediction model.

We use DGL Library and PyTorch to implement the Static GCN, Time Series LSTM and Multi-Feature-based GCN-LSTM. The model uses sigmoid and linear classifier as the final prediction layer. For the Static GCN model, we adopted a two-layer convolution configuration with an intermediate embedding layer size of 48, and applied average pooling at the last layer to obtain a single vector representation for each graph. The LSTM network is configured with 2 layers, and the size of the middle embedded layer is 64. To avoid overfitting the models, we apply a dropout layer with a ratio of 0.5 to both LSTM and GCN.

In the model training process, we use batch gradient descent, and all data are used for one training. Random shuffle sets 80% of the data as the training set and the remaining 20% as the test set. The learning rate of the Static GCN model and Multi-Feature-based GCN-LSTM is set to 0.01, and the learning rate of the time-series LSTM model is designed to be 0.001. We use Adam as the optimizer.

Evaluation Metrics

According to the relevant evaluation functions of classification tasks [13]. We use three evaluation metrics including precision score, recall score and F1 score to evaluate the model's key positions prediction ability. Precision score and recall score present the model's ability to predict the future key positions distribution and to find the real key positions respectively. F1 score is the harmonic mean of precision and recall, its precision is on the basis of considering the influence of false positive (fp) and false negative cases (fn), can provide a more comprehensive evaluation for the model's ability.

$$\begin{aligned} precision &= tp/(tp+fp) \\ recall &= tp/(tp+fn) \\ F1 &= 2*(precision*recall)/(precision+recall) \end{aligned} \tag{11}$$

In the experiment, we use the F1 score function under the Sklearn metrics library. In order to balance the situation where the key positions number of negative is more than the number of positive, the function is adopted to the principle of averaging.

Result and performance analysis

In order to evaluate the effectiveness of our model on the task of key positions prediction, we compared the

MF-GCN-LSTM with SVM [35, 36], the Static GCN [12] and the Time Series LSTM model [32]. Table 2 represents the feature extraction ability of different models for projects in smart grid.

Among them, the SVM model classifies the future key positions matrix by capturing the linear feature of key positions distribution in projects data.

The Static GCN model uses the complete raise network and geographical distribution matrix throughout the whole raise cycle to predict the future key positions distribution, which represents the feature extraction of the correlation structure (*SNSTR*) and geographic distribution feature (*Geo.Dist.*) of the projects.

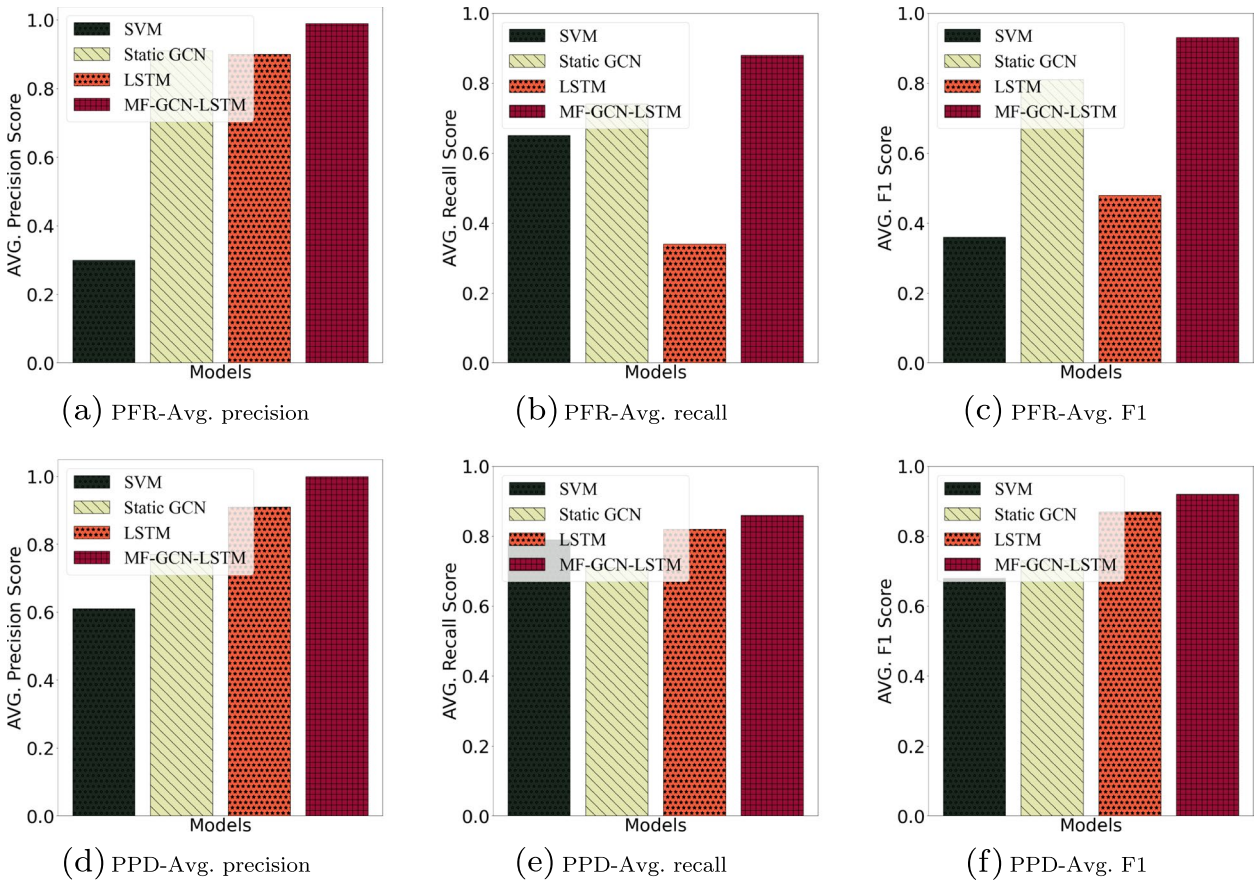
The Time Series LSTM model predicts the future key position matrix by capturing the time-dependency of the sequential key position matrices, which represents the learning of the time-dependency feature (*Time-Dept.*) of the projects.

The Multi-Feature-based GCN-LSTM model comprehensively combines the GCN and LSTM model, representing the feature extraction of correlation structure, geographic distribution and time-dependency of the projects at the same time.

The prediction performance of the baseline models and MF-GCN-LSTM model is shown in Tables 3, 4, 5, 6, in which Tables 3, 4, 5 represent the prediction performance of different models to 4 types of project in PFR dataset, and Table 6 represents the prediction performance to 3 types of project in PPD dataset. Besides, we plot the average metrics of different models in Fig. 4.

Table 2 Feature capture of different models

Model Name	SN STR	Geo. Dist.	Time Dept.
SVM	×	×	×
Static GCN	✓	✓	×
LSTM	×	×	✓
MF-GCN-LSTM	✓	✓	✓



(a) PFR-Avg. precision (b) PFR-Avg. recall (c) PFR-Avg. F1

(d) PPD-Avg. precision (e) PPD-Avg. recall (f) PPD-Avg. F1

Fig. 4 The average performance of three non-linear models on PFR and PPD

Table 3 Precision score results-PFR

Models	Project Categories				AVG.
	PFR0	PFR1	PFR2	PFR3	
SVM	0.43	0.26	0.16	0.37	0.30
Static GCN	0.64	1.00	1.00	**1.00**	0.91
LSTM	0.67	1.00	0.92	**1.00**	0.90
MF-GCN-LSTM	**1.00**	**1.00**	**1.00**	0.90	**0.99**

The boldface in the tables represents the optimal values among the four models, and the underline represents the suboptimal values

Table 4 Recall score results-PFR

Models	Project Categories				AVERAGE
	PFR0	PFR1	PFR2	PFR3	
SVM	0.17	0.76	0.79	0.90	0.65
Static GCN	0.64	0.96	0.57	0.79	0.74
LSTM	0.36	0.41	0.45	0.14	0.34
MF-GCN-LSTM	**0.67**	**0.97**	**0.98**	**0.90**	**0.88**

The boldface in the tables represents the optimal values among the four models

Comparison of models' prediction accuracy

Firstly, we compare the performance of different models on PFR dataset:

From Table 3, it can be seen that the three non-linear models have a high prediction precision for the key positions of the four types of PFR project, but for the project of PFR0, the precision of MF-GCN-LSTM model is much higher than the three baseline models. Judging from the average value of the four types of projects in 4(a), although the precision is not much different, the precision score of the MF-GCN-LSTM model is basically 100%.

Considering the sparsity of the key positions distribution in the data, the value of finding a real key position is higher than that of correctly judging the non-key position. Therefore, in the comparison, we pay more attention to the performance of key position prediction recall and F1 score.

As shown in Table 4, it can be seen that the key position prediction recall score of different models varies greatly. The recall score of the MF-GCN-LSTM model performs better on the four types of PFR project than the three baseline models, with a maximum of 0.98. From the average point of view in Fig. 4(b), it is also much higher than three baseline models.

Table 5 F1 score results-PFR

Models	Project Categories				AVG.	Diff.
	PFR0	PFR1	PFR2	PFR3		
SVM	0.24	0.39	0.26	0.53	0.36	0.29
Static GCN	0.64	0.98	0.73	0.88	0.81	0.34
LSTM	0.47	0.58	0.60	0.25	0.48	0.35
MF-GCN-LSTM	**0.80**	**0.99**	**0.99**	**0.92**	**0.93**	**0.19**

The boldface in the tables represents the optimal values among the four models

Table 6 Three metrics results-PPD

Metrics		Models			
		SVM	Static GCN	LSTM	MF-GCN-LSTM
Pre	PPD0	0.53	0.67	0.91	**1.00**
	PPD1	0.70	0.81	0.94	**1.00**
	PPD2	0.59	0.82	0.84	**1.00**
	AVG.	0.61	0.77	0.91	**1.00**
Rec	PPD0	0.81	0.63	**0.85**	0.80
	PPD1	0.90	0.71	0.88	**0.94**
	PPD2	0.66	0.78	0.75	**0.84**
	AVG.	0.79	0.71	0.82	**0.86**
F1	PPD0	0.64	0.65	0.87	**0.88**
	PPD1	0.79	0.75	0.91	**0.97**
	PPD2	0.63	0.79	0.83	**0.91**
	AVG.	0.68	0.73	0.87	**0.92**
	Diff.	0.16	0.14	**0.08**	0.09

The boldface in the tables represents the optimal values among the four models, and the underline represents the suboptimal values

At last, we compare the performance of different models by F1 score. As shown in Table 5, it can be seen that the MF-GCN-LSTM model has a unique brilliant performance. A vertical comparison shows that the MF-GCN-LSTM model has the highest F1 score for four types of PFR project. Through horizontal comparison, we can find that the F1 score of the MF-GCN-LSTM in different file categories is the most stable, with an extreme value difference (Diff.) of 0.19, while the Diff. of two baseline models are about 30 percentage points.

In order to reduce the repeated discussion, we describe all the evaluation metrics values of models on PPD in Table 6. The most important difference between the two datasets is the average number of nodes in the network (as can be seen from Table 1), which leads to the high

proportion of key positions in PPD dataset. From Table 6, it can be seen that the MF-GCN-LSTM model still has the most outstanding performance in different evaluation indexes.

Performance analysis from the perspective of models

Based on the above analysis, the MF-GCN-LSTM is better than the other three baseline models in terms of all three evaluation metrics in the key positions prediction task.

At the same time, the MF-GCN-LSTM model is robust to predict different PFR project types. As shown in Fig. 5, the MF-GCN-LSTM predicts the smallest (or second smallest) F1 extreme value difference for different item categories on both datasets. The difference in prediction performance among different models mainly comes from the feature extraction ability.

Since LSTM model can not deal with the network data of topological structure, it is necessary to discretize the data before model training, which leads to the neglect of the important relationship feature between projects in the correlation. GCN can handle the network structure, which allows the model to capture the mutual relationship between projects, but it ignores the periodic changes in projects. So the prediction performance of the GCN model on different types of project is not stable and robust enough.

Compared with the Static GCN, the MF-GCN-LSTM model refines the network structure feature in the time dimension, and improves the prediction stability through the LSTM gating mechanism. Compared with LSTM, it avoids discretization, integrates the structure of correlation into the process of feature extraction and greatly improves the accuracy of prediction. In addition, the introduction of a convolution mechanism reduces the data dimension and improves the training speed.

The above analysis, illustrates the necessity of multi-dimensional feature extraction model, and on the other hand, inspires us to explore the interaction between projects and different features in the future.

Validation of framework's distributed training and inference

In this section, we demonstrate that MF-GCN-LSTM is a framework that supports distributed training and inference to improve model's inference efficiency through experiments.

In Fig. 6, we plot the average training loss curves of three non-linear models for different types of project under PFR and PPD dataset. As shown in Fig. 6, It can be seen that the training loss of our model and static GCN has a fast descent and convergence rate, while LSTM takes a long time to reach convergence due to the bloated multi-layer gating mechanism.

To save the convergence time of the model and make the prediction framework support grid projects decisions faster, we propose a distributed training inference mechanism of MF-GCN-LSTM, where we keep the stable GCN model which has been trained under a large amount of data in the projects data center (e.g., cloud) for incremental learning and encoding the project networks, and place the LSTM at the edge of the project decisions

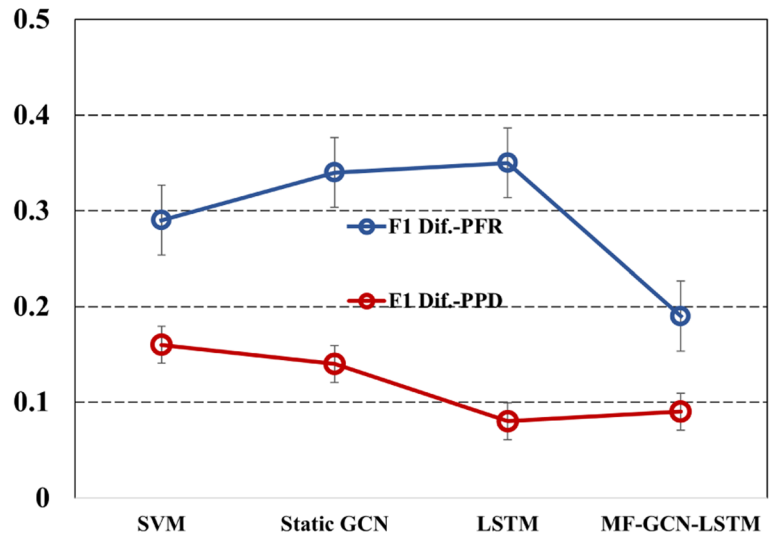


Fig. 5 Extreme f1 difference of different models

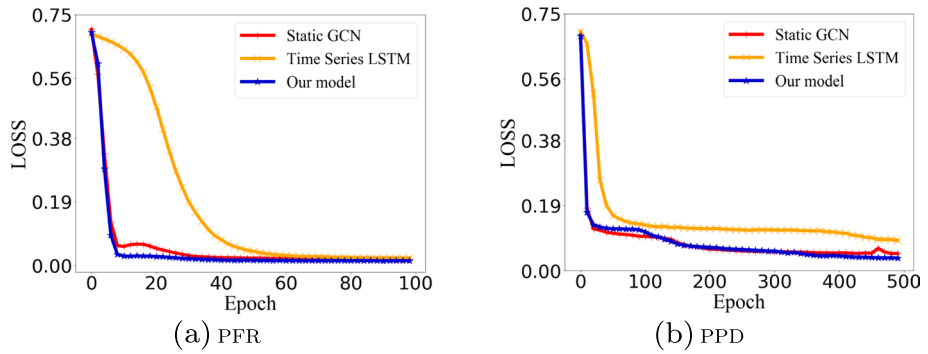


Fig. 6 The average training loss curve of three non-linear models on PFR and PPD

for result output based on the encoding results (without further learning).

To illustrate the feasibility of the above scheme, we split the MF-GCN-LSTM and test the inference efficiency on the new data distribution of PFR. Specifically, we use the model trained under partial dataset PFR and then make the GCN in MF-GCN-LSTM for incremental learning under new data distribution, but the LSTM keeps the parameters of the previous model training, and compare the inference efficiency with two baselines that are also pre-trained and completed, and the results are as Fig. 7.

It is worth mentioning that due to the MF-GCN-LSTM and Static GCN infer faster than Time Series LSTM, its inference accuracy converges at 100 epoch, while the LSTM model, although pre-trained, still requires a retraining process of 3000-4000 epochs to convergence. In order to compare the inference efficiency between models intuitively, we extended the f1/auc-epoch curves of MF-GCN-LSTM and Static GCN with the values of 1000 epoch as the benchmark, i.e., in the real case MF-GCN-LSTM and Static GCN were only tested for inference of up to 1000 epoch (inference termination cut off).

It can be seen that from Fig. 7 both MF-GCN-LSTM and Static GCN have high inference efficiency, and they converge after retraining within 100 epochs. This is due to the high inference efficiency of GNN, similar to the parameter sharing mechanism of CNN, GCN only needs a small number of parameters to achieve the information convolution of the whole network. Compared with LSTM, in MF-GCN-LSTM, the retrained GCN encodes the projects network, after which the encoding results are directly input to the pre-trained LSTM. The mechanism can both improve the overall inference accuracy and reduces the data dimensionality of the LSTM input, thus reducing the training parameters of the LSTM.

Conclusion

In this paper, we explore and analyze the projects key positions in smart grid and propose a spatial-temporal-social multi-feature-based model: Multi-Feature-based GCN-LSTM to solve the key positions prediction task. In order to realize the integration of various complex features of projects in smart grid, we propose a regional spatial-temporal modeling algorithm and implement the MF-GCN-LSTM model for the prediction of key

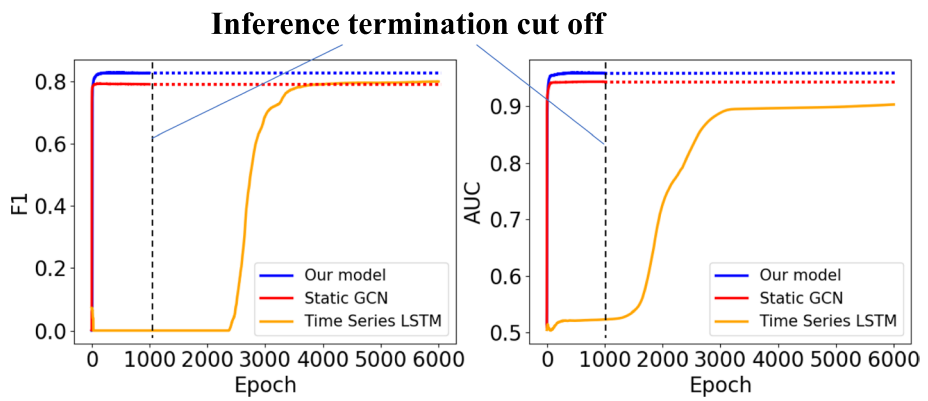


Fig. 7 Comparison of different models inference efficiency

positions. The experiment results of 7 types of project under 2 large-scale datasets show that the MF-GCN-LSTM model has significant improvement than the other three baseline models to extract the multi-dimensional features of projects in smart grid and can effectively predict the key positions. Moreover, MF-GCN-LSTM has a more stable prediction ability when the project category or project mode of the key positions change in heterogeneous scenarios.

In future work, We will explore the degree of importance of temporal, spatial, and correlation features in grid project forecasting to enhance the overall guidance of the work to relevant staff. Besides, we will also actively try the applicability of MF-GCN-LSTM in other fields to apply this multi-feature capture and prediction idea to more relevant prediction fields. Regarding the practical deployment of magic, we will first try to build the basic cloud-side collaborative system and deploy the model in a distributed manner to test the actual runtime of MF-GCN-LSTM.

Acknowledgements
The authors would like to thank to anonymous reviewers for their valuable comments on the manuscript.

Authors' Contributions
All authors took part in the discussion and analysis of the work described in the paper. Shaoyuan Huang initiated the research and led the entire work. Guozheng Peng, Yuxi Zhang and Juan Zhao contributed to the methodology and design of this paper, Keping Zhu and Heng Zhang carried out the experimental work and drafted the manuscript. All authors read and approved the final manuscript.

Funding
This research is supported by the Science and Technology Project of State Grid Corporation of China under grant No. 1400-202055132A-0-0-00.

Availability of data and materials
Due to signed confidentiality agreements, we are actively disclosing data with data providers, and our other materials and codes are publicly available.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China. [2]Institute of Biology, China Electric Power Research Institute, Beijing, China. [3]Institute of Biology, State Grid Economic And Technological Research Institute, Beijing, China. [4]Institute of Biology, State Grid Zhejiang Electric Power Company, Zhejiang, China.

Received: 30 November 2021 Accepted: 10 August 2022
Published online: 29 September 2022

References

1. Lei K, Qin M, Bai B, Zhang G, Yang M (2019) GCN-GAN: A non-linear temporal link prediction model for weighted dynamic networks. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, p. 388-396. https://doi.org/10.1109/INFOCOM.2019.8737631
2. Lu Z, Sagduyu Y, Shi Y (2016) Friendships in the air: Integrating social links into wireless network modeling, routing, and analysis. 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), p. 322-327. https://doi.org/10.1109/INFCOMW.2016.7562095
3. Memos VA, Psannis KE, Ishibashi Y, Kim BG, Gupta BB (2018) An Efficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework. Future Gener Comput Syst 83:619–628. https://doi.org/10.1016/j.future.2017.04.039
4. Dong Y, Cai Z, Yu M, Sturer M (2011) Modeling and simulation of the communication networks in Smart grid. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, p 2658–2663. https://doi.org/10.1109/ICSMC.2011.6084073
5. Pan T, et al (2019) INT-path: Towards optimal path planning for in-band network-wide telemetry. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, p 487–495. https://doi.org/10.1109/INFOCOM.2019.8737529
6. Wang X, Li X, Pack S, Han Z, Leung VCM (2020) STCS: Spatial-temporal collaborative sampling in flow-aware software defined networks. IEEE J Sel Areas Commun 38(6):999–1013. https://doi.org/10.1109/JSAC.2020.2986688
7. Scellato S, Noulas A, Lambiotte R, Mascolo C (2021) Socio-spatial properties of online location-based social networks. Proc Int AAAI Conf Web Soc Media 5(1):329–336. Retrieved from https://ojs.aaai.org/index.php/ICWSM/article/view/14094
8. Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11). Association for Computing Machinery, New York, p 1082–1090. https://doi.org/10.1145/2020408.2020579
9. Du D, Wang H, Xu T, Lu Y, Liu Q, Chen E (2017) Solving link-oriented tasks in signed network via an embedding approach. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), p 75–80. https://doi.org/10.1109/SMC.2017.8122581
10. Liben-Nowell D, Kleinberg J (2003) The link prediction problem for social networks. In: Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03). Association for Computing Machinery, New York, p 556–559. https://doi.org/10.1145/956863.956972
11. Huang Z, Lin DKJ (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS J Comput 21(2):286–303. https://doi.org/10.1287/ijoc.1080.0292
12. Qiu J, Tang J, Ma H, Dong Y, Wang K, Tang J (2018) DeepInf: social influence prediction with deep learning. In:Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18). Association for Computing Machinery, New York, p2110–2119. https://doi.org/10.1145/3219819.3220077
13. Liu Y, Shi X, Pierce L, Ren X (2019) Characterizing and forecasting user engagement with in-app action graph: a case study of snapchat. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, p 2023–2031. https://doi.org/10.1145/3292500.3330750
14. Sahoo SR, Gupta BB (2021) Multiple features based approach for automatic fake news detection on social networks using deep learning. Appl Soft Comput 100:106983. https://doi.org/10.1016/j.asoc.2020.106983
15. Liu L, Xu L, Wangy Z, Chen E (2015) Community detection based on structure and content: a content propagation perspective. 2015 IEEE International Conference on Data Mining, p 271–280. https://doi.org/10.1109/ICDM.2015.105
16. Mededovic E, Douros VG, Mähönen P (2019) Node centrality metrics for hotspots analysis in telecom big data. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), p 417–422. https://doi.org/10.1109/INFCOMW.2019.8845204
17. Liu Y, Jia R, Xie X, Liu Z (2019) A two-stage destination prediction framework of shared bicycles based on geographical position recommendation. IEEE Intell Transp Syst Mag 11(1):42–47. https://doi.org/10.1109/MITS.2018.2884517
18. Jianmei L, Dongmei C, FengXi L, Qingwen H, Siru C, Lingqiu Z, et al (2017) A bus arrival time prediction method based on GPS position and real-time traffic flow. In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and

Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), p 178–184. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.42
19. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G (2016) Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16). JMLR.org, 2071–2080
20. Li H, Sun C, Li X, Xiong Q, Wen J, Wang X, et al (2020) Mobility-aware content caching and user association for ultra-dense mobile edge computing networks. In: GLOBECOM 2020 - 2020 IEEE Global Communications Conference, p 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9348257
21. Zhang H, Wang X, Chen J, Wang C, Li J (2020) D2D-LSTM: LSTM-based path prediction of content diffusion tree in device-to-device social networks. AAA I:34. https://doi.org/10.1609/aaai.v34i01.5363
22. Martín C (2021) Effort prediction for the software project construction phase. J Softw Evol Process 06:33. https://doi.org/10.1002/smr.2365
23. Huang CH, Hsieh SH (2020) Predicting BIM labor cost with random forest and simple linear regression. Autom Constr 118:103280. https://doi.org/10.1016/j.autcon.2020.103280
24. Zhang M, Chen Y (2018) Link prediction based on graph neural networks[J]. Adv Neural Inf Process Syst 31
25. Huo Z, Huang X, Hu X (2018) Link prediction with personalized social influence. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11892
26. Li D, Deng L, Bhooshan Gupta B, Wang H, Choi C (2019) A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. Inf Sci 479:432–447. https://doi.org/10.1016/j.ins.2018.02.060
27. Alazab M, Khan S, Krishnan SSR, Pham QV, Reddy MPK, Gadekallu TR (2020) A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid. IEEE Access 8:85454–85463. https://doi.org/10.1109/ACCESS.2020.2991067
28. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18). AAAI Press, p 3634–3640
29. Geng X, Li Y, Wang L, Zhang L, Yang Q, Ye J, Liu Y (2019) Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. Proc AAAI Conf Artif Intell 33(01):3656–3663. https://doi.org/10.1609/aaai.v33i01.33013656
30. Wang J, et al (2017) Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, p 1–9. https://doi.org/10.1109/INFOCOM.2017.8057090
31. Liu H, Xu H, Yan Y, Cai Z, Sun T, Li W (2020) Bus arrival time prediction based on LSTM and spatial-temporal feature vector. In: IEEE Access, vol. 8, p 11917–11929. https://doi.org/10.1109/ACCESS.2020.2965094
32. Yang C, Shi X, Jie L, Han J (2018) I know you'll be back: interpretable new user clustering and churn prediction on a mobile social application. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18). Association for Computing Machinery, New York, p 914–922. https://doi.org/10.1145/3219819.3219821
33. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17. Curran Associates Inc, Red Hook, p 1025–1035
34. GeoHash (2022) Tips and Tricks. [EB/OL]. http://geohash.org/site/tips.html
35. Wang Y, Zhang Z, Ma L, Chen J (2014) SVM-based spectrum mobility prediction scheme in mobile cognitive radio networks. ScientificWorldJournal 2014:395212. https://doi.org/10.1155/2014/395212
36. Müller KR, Smola AJ, Rätsch G, Schölkopf B, Kohlmorgen J, Vapnik V (1997) Predicting time series with support vector machines. In: Gerstner W, Germond A, Hasler M, Nicoud JD (eds) Artificial Neural Networks — ICANN'97. ICANN 1997. Lecture Notes in Computer Science, vol 1327. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0020283

Publisher's Note