EAST: An Enhanced Automated Machine Learning Library for Spatio-Temporal Forecasting

Songyu Ke*
Shanghai Jiao Tong University
Shanghai, China
JD Intelligent Cities Research
Beijing, China
songyu-ke@outlook.com

Li Song*
JD iCity, JD Technology
Beijing, China
JD Intelligent Cities Research
Beijing, China
song200626@gmail.com

Kainan Bao*
Southwest Jiaotong University
Chengdu, China
JD Intelligent Cities Research
Beijing, China
baokainan123@gmail.com

Zheyi Pan
JD iCity, JD Technology
Beijing, China
JD Intelligent Cities Research
Beijing, China
zheyi.pan@outlook.com

Junbo Zhang†
JD iCity, JD Technology
Beijing, China
JD Intelligent Cities Research
Beijing, China
msjunbozhang@outlook.com

Yu Zheng
JD iCity, JD Technology
Beijing, China
JD Intelligent Cities Research
Beijing, China
msyuzheng@outlook.com

ABSTRACT

Huge spatio-temporal data (e.g., traffic flow, human mobility, and geographical data) are generated in modern cities. Accurately forecasting over spatio-temporal data enables many essential applications in intelligent cities, such as traffic management, public safety, and economy. To improve the ease of use, we propose an **E**nhanced **A**utomated machine learning library for **S**patio-**T**emporal forecasting, entitled **EAST**. In EAST, we mainly reconstruct three types of automated machine learning methods, namely, AutoSTPoint, AutoSTGrid, and AutoSTGraph for spatio-temporal point (STPoint) forecasting, spatio-temporal grid (STGrid) forecasting, and spatio-temporal graph(STGraph) forecasting, respectively. We adapt structure-aware algorithms using neural architecture search methods. The search space is elaborately designed according to the structures and characteristics of spatio-temporal data. We implement EAST with the popular deep learning frameworks. Finally, we conduct experiments on real-world datasets to demonstrate the effectiveness and superiority of our EAST.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Information systems → Spatial-temporal systems**.

KEYWORDS

Spatio-temporal data, automated machine learning

1 INTRODUCTION

Spatio-temporal data has been widely studied in academia and industry for its difficulties in capturing spatial and temporal dependencies. Accurately forecasting spatio-temporal data would benefit both managers and residents in the city. Researchers have proposed many deep neural network models to deal with spatio-temporal forecasting in rich real-world scenarios over the past few years [5, 10, 14]. However, spatio-temporal dependencies are always diverse from different tasks, leading to different neural network structures. Designing neural networks for spatio-temporal forecasting always requires huge human efforts, which may limit the development of real-world applications. Recently, automated machine learning approaches [1] have attracted considerable attention because of their tremendous potential for finding the best neural architecture. AutoST[3] and AutoSTG[8] have taken a slight step in the spatio-temporal data modeling. However, a significant gap remains between real-world applications and the easy way to design and deploy a model. A unified and practicable framework is eager to alleviate those problems.

To build a unified automated machine learning framework for spatio-temporal data, we divide spatio-temporal data into three main categories concerning the data structure. Then, neural architecture search (NAS) technologies are applied to find the optimal neural networks for each task. Specifically, the search space is tailored to adapt to different data structures. Our proposed framework uses the gradient-based search algorithm to find the optimal network. Further, we build the EAST based on the popular framework PyTorch. Without loss of generality, EAST consists of three AutoML methods for devise spatio-temporal forecasting tasks, *i.e.*, AutoSTPoint, AutoSTGrid, and AutoSTGraph.

Our contributions are three-fold: a) We propose a unified AutoML framework for ST forecasting, which classifies various ST forecasting problems into three main categories: STPoint, STGrid, and STGraph forecasting; b) We design the corresponding search spaces for three ST forecasting tasks and implement EAST by integrating gradient-based network architecture search algorithms, which can significantly reduce the human effort required for ST prediction model development; c) Extensive experiments on real-world datasets to show the effectiveness and superiority of our EAST.

2 FRAMEWORK

Traditionally, expert experience and human resources are indispensable for designing excellent deep neural network architectures for each task and dataset. NAS is a practical approach for reducing the heavy experimental burden by automatically searching for the

*Songyu Ke, Li Song, and Kainan Bao contributed equally to the paper.
†Junbo Zhang is the corresponding author.

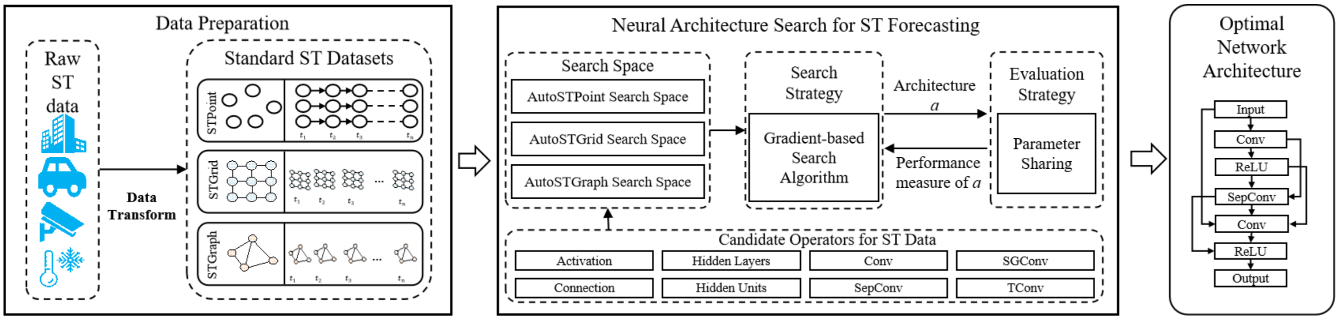

Figure 1: The framework of EAST. a) Data Preparation is used to transform raw ST data into STPoint, STGrid, STGraph datasets. b) Neural Architecture Search(NAS) for ST Forecasting is used to find the optimal network architecture.

optimal architecture. Generally, NAS consists of two main components, i.e., search algorithm and search space. Specifically, the neural architecture is organized as a direct acyclic graph (DAG) where the node indicates the data or features and the edge indicates an operator. The search space consists of available operators, while the search algorithm determines how to explore the search space to find the optimal architecture. We design and implement EAST, an Enhanced Automated machine learning library for Spatio-Temporal data. The framework is illustrated in Figure 1.

EAST consists of two main modules, i.e., data preparation and NAS for ST forecasting. The data preparation module is used to standardize data. In detail, different types of raw spatio-temporal data collected from IoT systems and mobile devices are divided and formatted into three main structures: the point-structured ST data (STPoint), grid-structured ST data (STGrid), and graph-structured ST data (STGraph). The NAS module aims at finding the optimal network architecture, consisting of four components: candidate operators for ST data, search space, search strategy, and evaluation strategy. We assembled three automated machine learning approaches: AutoSTPoint, AutoSTGrid, and AutoSTGraph for STPoint, STGrid, and STGraph forecasting, respectively. One can select the corresponding operators from the candidate operators, and generate the search spaces for AutoSTPoint, AutoSTGrid, and AutoSTGraph, respectively. These approaches employ the same search strategy (e.g., gradient-based search algorithm) and evaluation strategy to find the optimal network structure for a specific spatio-temporal forecasting task from the search space.

3 METHODOLOGY

Without loss of generality, we focus on spatio-temporal forecasting tasks. Formally, we give the problem definition as follows.

Problem Statement: Given historical observed region measurements $X_1, X_2, ..., X_{in}$, and optional external features E, we aim to predict the future measurements $\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_{out}$ for all locations.

X can be instantiated as STPoint, STGrid, and STGraph datasets according to real-world applications.

3.1 Search Space

Since devise data sources exist in various ST applications, it is non-trivial to design one model for all applications. To build a unified framework, we first group spatio-temporal data into three main

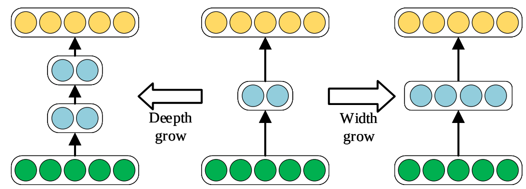

Figure 2: Automated network capacity growing in AutoST-Point

categories: STPoint, STGrid, and STGraph. Then, three different structure-aware search spaces are delicately designed for the corresponding categories.

3.1.1 AutoSTPoint for STPoint Forecasting. STPoint forecasting tasks, e.g., air quality prediction, are widespread in the real world. But sometimes, determining the number of layers in a neural network and the number of neurons in each layer can be tricky. Therefore, we try to use a neural architecture search algorithm to determine these architecture parameters to reduce manual structure adjustment costs.

The method to determine the number of network layers refers to the strategy of AutoGrow[11], which grows from the minimal network until the model's performance on the validation set drops to a certain threshold. Besides, the method to determine the number of neurons in each layer is to find an appropriate subnetwork in a super-network using the search strategy based on gradient.

3.1.2 AutoSTGrid for STGrid Forecasting. AutoSTGrid model can adaptively adjust the network structure to capture spatial dependencies (e.g., spatial distance) and temporal dependencies (i.e., closeness, periodicity, and trend) of specific STGrid data. AutoSTGrid reduces the workload of manual adjustment, achieving the purpose of rapid application.

Candidate Operators. The selection of candidate operators has a critical effect on the performance of architecture search results. For STGrid forecasting, convolution operation plays a significant role in capturing the correlation of nearby and distant regions. Therefore, we take the standard convolution and separable convolution into consideration. Moreover, local correlations captured by low-level convolution and global correlations captured by high-level convolution are important to STGrid forecasting. Therefore, skip connection, which fuses multi-level correlations, is also necessary.

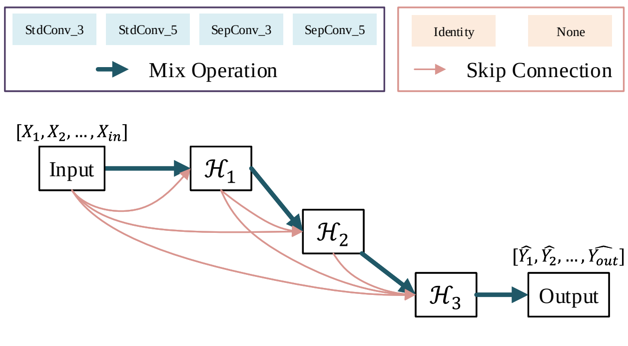

Figure 3: Search space of AutoSTGrid [3]

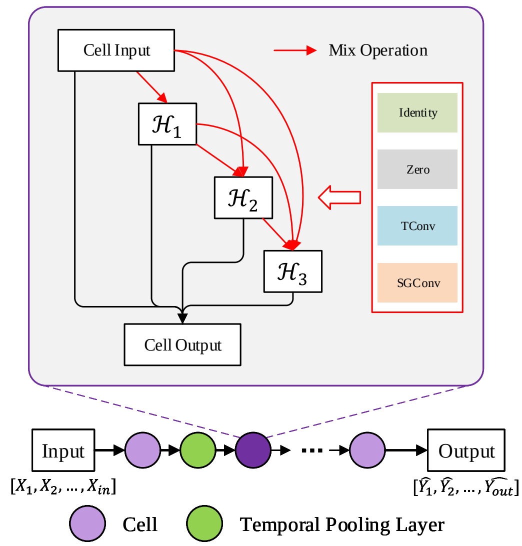

Figure 4: Search space of AutoSTGraph[8]

Therefore, we decide to include six operators into the search space of AutoSTGrid, i.e., convolution operations including 3×3 or 5×5 standard convolution, 3×3 or 5×5 separable convolution, skip connection operations consisting of none operation (do not connect) and identity operation (connect). Figure 3 shows the entire search space.

3.1.3 AutoSTGraph for STGraph Forecasting. The STGraph is also an important data structure to describe the ST data in real-world applications. Many hand-crafted graph neural networks are proposed to predict urban traffic, air quality, etc., accurately. However, no model can handle all different prediction tasks due to differences in data types, fields, and distributions. Therefore, there is a rising demand for neural architecture searching in predicting over spatio-temporal graphs[8].

AutoSTGraph[8] adopts a cell-based search space (as Figure 4 shown). The cell employs spatial and temporal convolution to capture spatial and temporal correlations, and the pooling layer in AutoSTGraph increases the temporal receptive fields. In particular, AutoSTGraph uses meta-learning techniques to model each node and edge in the graph individually so that the final network can flexibly use the common knowledge and unique knowledge of nodes. AutoSTGraph finally concatenates the outputs of all the cells and pooling layers and feeds them into a fully connected layer to generate the final prediction.

3.2 Search Strategy & Evaluation Strategy

We employ a gradient-based search algorithm and a weight-sharing model evaluation strategy. Specifically, since the candidate operators are in a discrete space, which can not be optimized through backpropagation, the gradient-based search algorithm[7] makes a relaxation to generate a continuous space. Assuming the set of candidate operations is O which may consist of universal and grid-structure operators, the relaxation operator over all candidate operators between two adjacent nodes is defined as $\sum_{o \in O} \frac{exp(\alpha_o)}{\sum_{o' \in O} exp(\alpha_{o'})} o(x)$, where α is the structure parameter that determine the weight of each operator, x in the output features of previous layer. To find the best parameter, a bi-level optimization problem should be solved over train and validation set as,

$$\begin{aligned} &\min_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ s.t.\ \ & w^*(\alpha) = \arg\min_w \mathcal{L}_{train}(w, \alpha), \end{aligned} \tag{1}$$

where w and α are the model parameters and architecture parameters seperately. The above optimization problem can be effeciently solved by an approximation[7].

4 EXPERIMENTS WITH EAST

4.1 Library Overview

We implement EAST with `PyTorch 1.8.2`[1] and `PyTorch-Lightning 1.5.6`[2]. PyTorch is one of the most popular deep learning frameworks worldwide. PyTorch-Lightning provides a flexible interface and powerful plugins for training and inference, greatly reducing the difficulty of implementing network architecture searching algorithms.

The mainstream deep learning workflow usually includes data processing, model implementation, and training. Therefore, we design and implement three main modules in EAST, i.e., general data interfaces for three types of ST data, the automatic search space construction, and the network architecture search algorithm.

To adapt to different spatiotemporal prediction tasks and improve the reusability of EAST, we design a set of spatio-temporal data sample specifications according to the characteristics of spatio-temporal data and use it in the implementation of candidate operators and the network architecture search algorithm:

- For *STPoint forecating*, we define the input sample as $X_{point} \in \mathbb{R}^{C \times N \times T}$, where C is the number of input features, N is the number of locations to predict, and T is the number of input timestamps.
- For *STGrid forecasting*, we take $X_{grid} \in \mathbb{R}^{C \times H \times W \times T}$ as the model input, where C is the number of input features, T is the number of input timestamps, and H, W are the height and width of the grid, respectively.
- For *STGraph forecasting*, we have dynamic temporal input $X_{graph} \in \mathbb{R}^{C \times N \times T}$, static node features $\mathcal{V} \in \mathbb{R}^{N \times F_v}$, and static edge features $\mathcal{A} \in \mathbb{R}^{N \times N \times F_A}$, where N is the number

[1]https://pytorch.org/
[2]https://www.pytorchlightning.ai/

Algorithm 1: Pseudocode of EAST's workflow

Data: RawSTData, TargetDataCategory(i.e., point, grid, or graph), SearchSpaceConfig
Result: OptimalModel
StandardSampleDataset ← DataTransform(RawSTData, TargetDataCategory);
`/* EAST integrates data transformation interfaces for common types of ST data, e.g., geo-sensory time series */`
SearchSpace ← GenerateSearchSpace(SearchSpaceConfig);
OptimalModel ← NetworkSearchAndTrain(StandardSampleDataset, SearchSpace);

of locations to predict, C is the dimension of dynamic input, T is the number of input timestamps, F_v and F_A are the dimension of node features and edge features, respectively.

In EAST, we have implemented several spatial operators (e.g., `Conv`, `SepConv`, `GraphConv`) and temporal operators (e.g., `TConv`, `TCN`). Moreover, we reproduce the search space described in [3, 8, 11], and a search space construction mechanism is developed to facilitate users to adjust the search space and add custom operators. Furthermore, we use PyTorch-Lightning to implement a gradient-based network search algorithm, which makes the searching and re-training process easier. Algorithm 1 shows the pseudocode of EAST's workflow.

4.2 Reproducibility

To verify the correctness of the reproduced models in EAST, we conduct experiments on real-world datasets, including AirBJ[5], TaxiBJ[14], TaxiGY[14], METR-LA[4], and PEMS-BAY[4]. Table 1 shows the experimental results. On STPoint forecasting, EAST achieves higher accuracy than expert-designed networks. And for the spatiotemporal grid prediction task, AutoST (EAST) adopts the same search space setting as the original one in [3] and achieves a comparable result, which shows the correctness of our reproduction. Moreover, AutoSTG (EAST) uses a reduced search space for STGraph prediction tasks and achieves even higher prediction accuracy on the two datasets.

5 CONCLUSION

In this paper, we propose a unified AutoML framework for spatio-temporal forecasting. Without loss of generality, we implement EAST with three spatio-temporal forecasting methods that cover most of the spatio-temporal forecasting scenarios. EAST reduces the human endeavor significantly. Experimental evaluations of real-world applications have demonstrated its effectiveness. In the future, we will explore automatic machine learning methods for the new category of spatio-temporal tasks (mainly trajectory mining) and multi-source heterogeneous data fusion.

[3] RMSE: Root Mean Square Error

Table 1: Performance comparisons among different models.

Task	Dataset	Model	RMSE[3]
STPoint Forecasting	AirBJ	DeepAir[13]	22.21
		EAST	22.15
STGrid Forecasting	TaxiBJ	STResNet[14]	17.51
		ST-3DNet[2]	17.82
		DeepSTN[6]	15.98
		AutoST[3]	15.88
		AutoST(EAST)	15.99
	TaxiGY	STResNet[14]	2.77
		ST-3DNet[2]	2.24
		DeepSTN[6]	2.15
		AutoST[3]	2.15
		AutoST(EAST)	2.17
STGraph Forecasting	METR-LA	GWNet[12]	6.27
		DCRNN[4]	6.16
		ST-MetaNet[9]	6.16
		AutoSTG[8]	6.10
		AutoSTG(EAST)	6.08
	PEMS-BAY	GWNet[12]	3.70
		DCRNN[4]	3.66
		ST-MetaNet[9]	3.72
		AutoSTG[8]	3.57
		AutoSTG(EAST)	3.55

ACKNOWLEDGEMENT

This paper was supported by National Natural Science Foundation of China(62172034) and Beijing Nova Program (Z201100006820053).

REFERENCES

[1] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural Architecture Search: A Survey. *Journal of Machine Learning Research* 20, 55 (2019), 1–21.
[2] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. 2019. Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3913–3926. https://doi.org/10.1109/TITS.2019.2906365
[3] Ting Li, Junbo Zhang, Kainan Bao, Yuxuan Liang, Yexin Li, and Yu Zheng. 2020. Autost: Efficient neural architecture search for spatio-temporal prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 794–802.
[4] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.
[5] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3428–3434. https://doi.org/10.24963/ijcai.2018/476
[6] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1020–1027.
[7] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).
[8] Zheyi Pan, Songyu Ke, Xiaodu Yang, Yong Yu, Junbo Zhang, and Yu Zheng. 2021. AutoSTG: Neural Architecture Search for Predictions of Spatio-Temporal Graphs. In *In Proceedings of the Web Conference*.
[9] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1720–1730.
[10] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting Citywide Crowd Flows in Irregular Regions Using Multi-View

Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering* (7 2020), 1–1. https://doi.org/10.1109/tkde.2020.3008774

[11] Wei Wen, Feng Yan, and Hai Helen Li. 2019. AutoGrow: Automatic Layer Growing in Deep Convolutional Networks. *arXiv preprint arXiv:1906.02909* (2019).

[12] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 1907–1913. https://doi.org/10.24963/ijcai.2019/264

[13] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep Distributed Fusion Network for Air Quality Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 965–973. https://doi.org/10.1145/3219819.3219822

[14] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. *31st AAAI Conference on Artificial Intelligence, AAAI 2017* (2 2017), 1655–1661.