Future Generation Computer Systems 139 (2023) 100–108

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Hybrid graph convolution neural network and branch-and-bound optimization for traffic flow forecasting

Youcef Djenouri [a], Asma Belhadi [b], Gautam Srivastava [c,e,f], Jerry Chun-Wei Lin [d,*]

[a] Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[b] Department of Technology, Kristiania University College, Oslo, Norway
[c] Department of Math and Computer Science, Brandon University, Brandon, Canada
[d] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway
[e] Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan
[f] Department of Computer Science and Math, Lebanese American University, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 11 June 2022
Received in revised form 15 August 2022
Accepted 19 September 2022
Available online 22 September 2022

Keywords:
Deep learning
Graph convolution neural network
Forecasting
Smart roads

ABSTRACT

In this study, we combine graph optimization and prediction in a single pipeline to investigate an innovative convolutional graph-based neural network for urban traffic flow prediction in an edge IoT environment. Pre-processing of the linked graph is first performed to remove noise from the set of original road networks of urban traffic data. Outlier detection strategy is used to efficiently explore the road network and remove irrelevant patterns and noise. The resulting graph is then implemented to train an extended graph convolutional neural network to estimate the traffic flow in the city. To accurately tune the hyperparameter values of the proposed framework, a new optimization technique is developed based on branch and bound. For comparison, an intensive evaluation is conducted with multiple datasets and baseline methods. The results show that the proposed framework outperforms the baseline solutions, especially when the number of nodes in the graph is large.

1. Introduction

The vast volume of data created by Internet of Things (IoT) sensors is a difficult computing problem to solve [1–3]. Edge Computing is a new computing paradigm that is being used to solve this issue [4,5]. It can bring computing power closer to end devices, reducing computational and energy consumption. Edge computing has been used for various smart city applications [6] such as smart home [7], smart healthcare [8], and smart transportation [9]. In the latter, one of the most important case studies is traffic forecasting for use in any edge computing environment [10]. The forecasting of traffic is of the utmost importance in ever evolving Intelligent Transportation Systems (ITS). At this stage of ITS, agreement can be found that it has a significant impact on global life [11–13]. It is vital to accurately predict accurate traffic flow in order to safeguard public transportation from both efficiency and congestion. ITS is also considered for decision support in traffic management. Some examples include traffic signal changes and controls mechanisms [14–16]. Nonetheless, it has been able to gradually attract a lot of attention from many research groups [17,18], with a slew of applications in optimal resource allocation, transportation anomaly detection, and city management [19–21].

1.1. Motivation

Standard Machine Learning (ML) and statistical methods can be commonly used to first build approaches in the prediction of traffic flow. However, these methods [22,23] only take into account temporal data and disregard the importance of spatial data in effective prediction. According to our observations, road conditions around roadways have a high causal relationship with the prediction of traffic flow. Deep learning-based traffic forecasting algorithms have recently been intensively investigated [17,24,25]. Some researchers [26,27] described the traffic network as grids and capture the spatial correlations using a convolutional neural network (CNN). Due to the irregularity of the roadways, modeling with grids will ultimately lose topology information within the traffic network. Some studies suggested representing roads as nodes and creating edges based on the connections between road networks to solve this problem. Then, using a Graph Convolutional Neural Network (GCNN), which can efficiently capture various correlations, aggregate the information of neighboring nodes to further incorporate prior knowledge about the transportation network and capture the complicated

* Corresponding author.
E-mail address: jerrylin@ieee.org (J.C.-W. Lin).

https://doi.org/10.1016/j.future.2022.09.018

spatiotemporal connections. Especially for large cities, the resulting graphs usually contain a considerable number of nodes and edges. Pre-processing of the graphs is required to remove noise, outliers, and redundant nodes from the graphs before they can be accurately processed. The abundance of hyperparameters offered by graph convolutional neural network-based models poses a significant additional barrier to traffic flow prediction. The random selection of these values leads to a significant decrease in the overall effectiveness of the learning process. In addition, it may be uncertain whether adequate convergence is achieved, since it takes a long time to adjust the parameters for such structures. Branch and bound methods based on optimization have been successfully used to solve similar problems in the recent research literature [28,29]. Motivated by the success of graph convolution neural network [12,18,30], outlier detection [30–32] in solving ITS applications, and the branch and bound in solving optimizations problems [28,29] this article presents a new framework for traffic flow forecasting. The novelty of this work is the development of an end-to-end framework that first cleans the data using outlier detection techniques, then processes the cleaned data using graph convolution neural network, and finally optimizes the learning process using hyperparameter optimization techniques.

1.2. Contributions

PROMOTION (graPh neuRal netwOrk fraMework fOr TraffIc flOw forecastiNg) is an intelligent hybrid framework for traffic flow forecasting that we developed in this work. The framework uses the neural network with graph convolution and explores intelligent graph-based pre-processing techniques. First, the road conditions are collected and the noise is removed with an intelligent filter based on anomaly detection. The new urban traffic flows are then predicted using the neural network with graph convolution. Below is a list of our contributions:

1. A unique filtering method based on outlier detection is used to remove noise from road-based graphs. In addition, advanced deep learning based on a neural graph convolutional network is used to predict traffic flow from the road network.
2. Accurately explore the configuration space of different PROMOTION hyper-parameter values, we propose a new branch-and-bound optimization technique. This hyper-parameter optimization approach improves the convergence of PROMOTION for traffic flow prediction.
3. PROMOTION is evaluated using extensive experiments with known traffic flow prediction data. Results show that our methodology outperforms the baseline algorithms in both accuracy and runtime.

Paper Organization. Section 2 gives an in-depth review of the main works for traffic flow forecasting, followed by details of the PROMOTION framework in Section 3. Next, Section 4 evaluates our framework. Section 5 provides the lessons learned in this research study and the future directions of PROMOTION. Lastly, Section 6 gives some brief concluding remarks.

2. Related work

Deep learning has recently been used in a variety of problems relating to intelligent transportation applications, including group anomaly detection, object detection, and traffic flow forecasting prediction [30,33–35]. Traffic flow forecasting is a basic component that is required in almost all applications. It is defined as the task of predicting flows from a certain urban road scenario [11, 35].

Li et al. [35] used a data-driven strategy to create a new graph that preserves hidden spatial–temporal relationships. This data-driven adjacency matrix can extract correlations that may not be apparent in a particular geographical graph. Then, to capture spatial–temporal dependencies synchronously, the authors offer a unique spatial–temporal fusion graph module and present a methodology for simultaneously capturing local and global correlations by combining a Gated dilated Convolutional Neural Network (CNN) module with a spatial–temporal fusion graph module. Long-term spatial–temporal disparities.

Lu et al. [18] predict traffic conditions for several time steps ahead and present a unique Spatio-temporal adaptive gated graph convolutional network. It is made up of two primary components: (1) multidimensional self-awareness, and (2) mix-hop AG-GCN extraction. To obtain the final prediction results, the output of several components is weighted together.

Zhang et al. [12] offered a graph-based temporal attention framework that takes both spatial and temporal correlation into account when forecasting traffic flow using data from numerous sensors. Because it keeps more features in the algorithms, this approach can be better capture spatial dependencies when using graph embedding techniques on sensor networks. It also introduced an attention technique for identifying temporal relationships.

Fang et al. [36] developed the long short-term memory with an attention mechanism for traffic flow forecasting. The attention mechanism is used to assist the network model in assigning different weights to various inputs, focusing on vital and important data. It also examines the regular pattern of traffic flow where a large training data with the associated ground truth is created.

Chen et al. [37] predicted the traffic flow and status of the selected road segment. The solution is based on deep reinforcement learning and the long short-term memory models. A fuzzy-based model is also used to characterize the traffic situation.

Zeng et al. [25] introduced a deep learning model for predicting the influx and outflow of each region of the entire metropolis. In particular, the temporal closeness, and the periodicity of crowd flow are analyzed using the structure of interactive attention and convolution. The interactive attention layer learns the importance of closeness, and the periodicity of crowd flow to model the long-term dependence of time, and then uses the feature fusion to capture complex correlations at different levels.

Ouyang et al. [38] proposed a new system for crude oil traffic flow forecasting that uses long short-term memory and a graph convolutional network. It first built a maritime transportation network, then calculated and predicted node traffic flow based on trajectory data as well as crude oil dock location. A maritime crude oil transportation network based on the supply–demand connection, ship trajectory, and route data is created. Then, for each port, the number of crude oil tankers that completed the up-load/offtake tasks is determined. The deep learning neural network called the long short-term memory and graph convolutional network is finally designed to extract the temporal and spatial properties of crude oil.

Raskar et al. [39] explored a hidden Markov model to create an improved traffic flow forecast model. The average true range, the exponential moving average, the relative strength indicator, and the rate of change is the input features that have been exposed to the hidden Markov model. The mean fitness-oriented dragonfly algorithm is also investigated to optimally tune the state numbers of the hidden Markov model.

Wang et al. [40] explored the time-variant graph convolutional network to model the dynamic spatial correlations at different times and capture the traffic graph's stable spatial correlations. In addition, a gated multi-scale temporal convolutional module is constructed to extract the long-range temporal dependencies inside traffic nodes, which are then fed into the first model to jointly investigate traffic node spatial correlations.

Yi et al. [41] developed an attention-based spatiotemporal graph attention network (ASTGAT), which investigates the attention mechanism with the graph convolution neural network to process the spatiotemporal features of traffic flow data. It is able to deepen the network to improve the accuracy of its predictions by using multiple residual convolutions and high-low feature concatenation. These techniques allow dynamic spatiotemporal correlations to be captured in the data. The temporal correlations of the latest, daily, and weekly periods are each modeled independently by one of the three elements that make up ASTGAT. It is possible to construct numerous spatiotemporal blocks for each component using the attention mechanism, dilated gated convolution, and the graph attention network.

To simulate the complicated spatiotemporal interactions in road networks, Lan et al. [42] proposed a unique Dynamic Spatial–Temporal Aware Graph Neural Network (DSTAGNN). It is possible to express the dynamic spatial meaning between nodes through an improved multi-head-attention mechanism. At the same time, it is able to capture a wide range of dynamic temporal dependencies of features of multireceptive fields thanks to multi-scale gated convolution.

For the purpose of short-term traffic flow prediction, Jiang et al. [43] introduced a unique method known as Deep Graph Gaussian Processes (DGGPs). To accurately represent the dynamic spatial features, an attention kernel and a combination of gaussian processes are used for aggregation. In addition, they were able to solve the current short-term traffic flow prediction models by combining the aggregation gaussian process, temporal convolution gaussian process, and gaussian process with a linear kernel.

Current traffic flow prediction technologies, especially those based on graph convolutional neural networks, have several weaknesses. When it comes to real-world scenarios, the first consideration is scalability in terms of accuracy and processing cost. The second problem is that these solutions are inadequate. Only the detection model has been trained, and there is no end-to-end framework method for a traffic flow prediction scenario. This study investigates an intelligent and end-to-end traffic flow prediction framework that could be used in intelligent traffic scenarios. This can be achieved by incorporating outlier detection for data cleaning, graph convolution neural network for training, and hyperparameter optimization to determine the best parameters for the learning process. In the next section, the proposed framework is explained in detail.

3. PROMOTION: graPh neuRal netwOrk fraMework fOr TraffIc flOw forecastiNg

3.1. Framework overview

In this section, the PROMOTION (graPh neuRal netwOrk fraMework fOr TraffIc flOw forecastiNg) framework is explained in detail. As shown in Fig. 1, PROMOTION consists of three stages: (1) Pre-processing: in this stage, an attempt is made to remove irrelevant nodes from the road network graph. In this context, the outlier detection algorithm is used to identify regions that are different from the whole input region. (2) Graph convolution neural network: the graph convolution neural network architecture is extended at this level to accurately predict traffic flows. (4) Hyper-parameter optimization: In this step, we present an intelligent approach to automatically identify the best hyper-parameters for all previous phases of the PROMOTION framework. It should be noted that the proposed system requires a large number of fine-tuned parameters. Therefore, branch-and-bound optimization is used to find the optimal hyper-parameters of the proposed system. In this way, the space of hyper-parameters can be explored efficiently and the best accuracy can be achieved. In the following, we give a detailed explanation of each step of the PROMOTION framework.

3.2. Pre-processing

Input data represented by the road network can be viewed as a graph, where each node is a road and each edge is a connection between two adjacent roads. We also consider the information of the traffic in each road. The generated graphs from the road network are generally big, especially for big cities in the world. Pre-processing is needed to reduce the graph size by removing the irrelevant nodes. In PROMOTION, we consider the use of outlier detection to remove noises, and unnecessary regions. Our target is not to exclude these regions but they need further analysis and investigation by the city planners which is out of the scope of this research work. To identify outliers, we use the Local Outlier Factor (LOF) [44], which determines the uniqueness of each node based on its distance from the k-nearest neighbors. Because it makes no assumptions about data distributions, the LOF algorithm can find outliers regardless of data distribution. The key assumption is that the density around an outlier node differs dramatically from that of its neighbors. LOF is an unsupervised method for detecting outliers. It is useful when the data being analyzed is not labeled or cannot be labeled. This is typical in a road network with a large number of roads created. The reachability density value is determined for each node in the graph. This reachability density value is based on the number of neighbors of each node. Local reachability density will then be used to calculate the LOF value. Nodes with a LOF value larger than one are considered as outliers and removed from the graph. Edges related to these nodes are removed as well.

3.3. Graph Convolutional Neural Network

Convolutional Neural Networks (CNNs) were developed for grid-like data, which is essentially image data with pixels arranged in a grid-like structure. An image can be viewed as a graph in which each pixel is a node connected to all its neighbors. However, this graph has some unique characteristics. It is nearly regular in that all nodes have the same number of neighbors, except for those at the boundary, and the arrangement of these nodes is ordered. CNNs use both of these characteristics to learn good representations. However, general graphs introduce some difficulties. They are not always regular, but even irregular, and the neighboring nodes are not ordered, and even if they are ordered, they have no semantic meaning. Therefore, conventional CNNs cannot be used directly in this case. For this reason, Graph CNN (GCNN) is used in this work to make such generalizations. Graph Convolutional Neural Networks (GCNN) are a deep learning framework for processing graph structured data. Instead of a pixel whose value is updated by the information contained in its neighboring pixels (as in convolutional neural networks), we have a node whose properties are updated by the information contained in its neighboring nodes. In the proposed GCNN, each node can be classified separately, the graph can be classified as a whole, the edges can be classified, or the existence of a link between two nodes can be investigated. To create a Graph Convolutional Neural Network (GCNN), we begin by building the graph's adjacency matrix A: For instance, in a non-oriented graph, $A_{ij} = 1$ if and only if there is a connection between nodes i and j, and $A_{ij} = 0$ if and only if i and j are disjoint. In addition, we

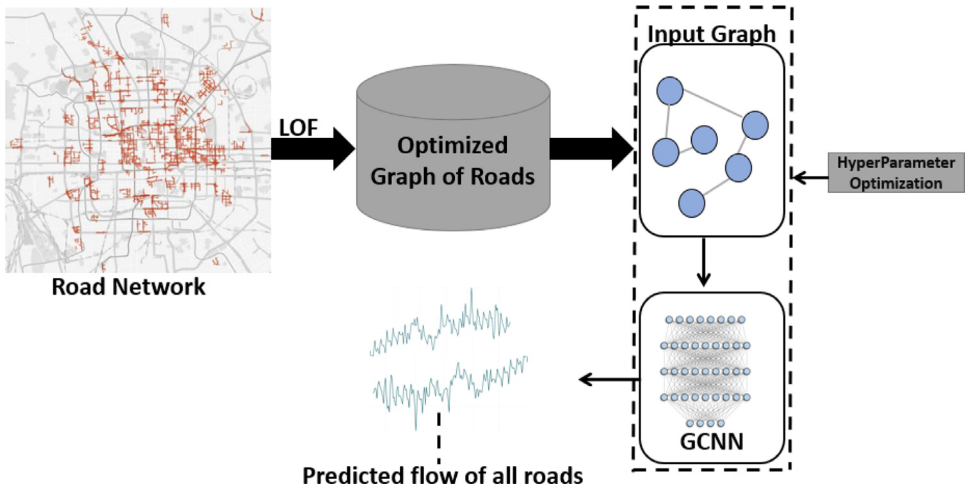

Fig. 1. PROMOTION Framework: The road network is first transformed to the optimized graph of roads using the local outlier factor algorithm. This is achieved by removing outliers and irrelevant roads. The optimized graph of roads is then injected in the graph of convolution neural network and learns the different features of the graph to predict the flow of all roads. HyperParameter optimization is also performed to tune the different parameters of the graph convolution neural network.

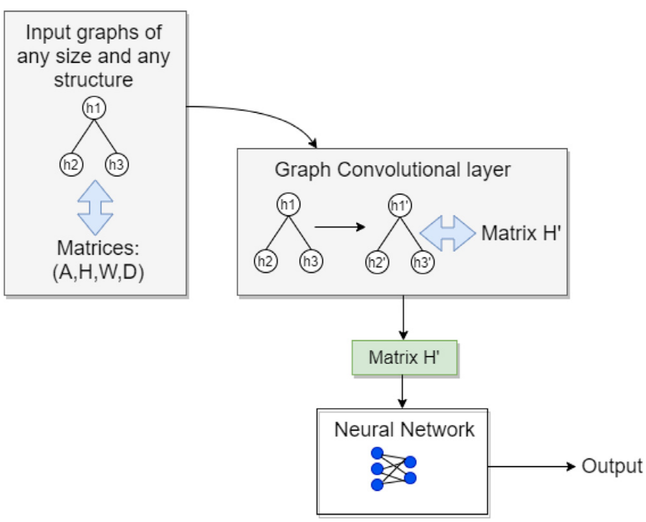

Fig. 2. GCNN diagram.

create the node matrix H and then construct the matrix shown below:

$$H' = \sigma\left(\widehat{D}^{-1}\hat{A}HW\right), \tag{1}$$

where W is a node-by-node shared linear transformation that may be learned (which is a linear layer in a deep learning framework), σ is a nonlinear function such as $ReLU$, $\hat{A} = A+I$ (it ensures that a node is always connected to itself to avoid discarding the central node.), $\widehat{D}$ is the degree matrix ($\widehat{D}^{-1}$ is present to normalize the adjacency matrix and prevent the features from exploding when they are summed). It is known as the mean-pooling update rule. The results of symmetric normalization are as follows:

$$H' = \sigma\left(\widehat{D}^{-1/2}\hat{A}\widehat{D}^{-1/2}HW\right), \tag{2}$$

The next thing to do is to set the update rule for the Graph Convolutional Network (GCN). At the moment, this is the Graph Convolutional layer that is most commonly used. Alternatively, in an extended model, nodes can transmit any message along edges $\overrightarrow{e_{ij}}$; the node then aggregates all received messages using a permutation invariant function (like a sum). Let $\overline{m_{ij}}$ be the message transmitted from node i to node j, computed as follows using the message function f_e:

$$\overrightarrow{m_{ij}} = f_e\left(\overrightarrow{h_i}, \overrightarrow{h_j}, \overrightarrow{e_{ij}}\right) \tag{3}$$

Following that, using the readout function below, all messages entering a node are aggregated:

$$f_v = \overrightarrow{h_i'} = f_v\left(\overrightarrow{h_v}, \sum_{j\in N_i}\overrightarrow{m_{ji}}\right), \tag{4}$$

where Ni is the collection of neighbors of node i. This gives rise to the message-passing neural network MPNN, which is useful in practice for small graphs. Both f_e and f_v are typically tiny multilayer perceptrons. A more general version is as follows:

$$\overrightarrow{h_i'} = \sigma\left(\sum_{j\in N_i}\alpha_{ij}W\overrightarrow{h_j}\right), \tag{5}$$

where α_{ij} is a coefficient that is either defined explicitly which causes some shortcomings, or

$$\alpha_{ij} = \frac{\exp\left(a_{ij}\right)}{\sum_{k\in N_i}\exp\left(a_{ik}\right)}, \tag{6}$$

where

$$a_{ij} = a\left(\overrightarrow{h_v}, \overrightarrow{h_j}, \overrightarrow{e_{lj}}\right) \tag{7}$$

a can be learned through a shared self-attention mechanism, as we have seen. The graph attention network update rule is what it is called. Fig. 2 explains the Graph Convolutional Neural Network (GCNN) used in our model.

3.4. Hyper-parameters optimization

Let $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots \mathcal{P}_{|P|}\}$ represent the set of all parameters that were used by the proposed framework. Let us call the domain space of the parameter $\mathcal{P}_i$, which contains all of its potential values, $D(\mathcal{P}_i)$. The different configurations are represented in the configuration space $\mathcal{C}$, and each configuration is a collection of values for all parameters in $\mathcal{P}$. The investigation of all configurations in $\mathcal{C}$ is required to determine the optimal values of all parameters in $\mathcal{P}$, which necessitates a lot of computing and memory resources, especially for quantities with

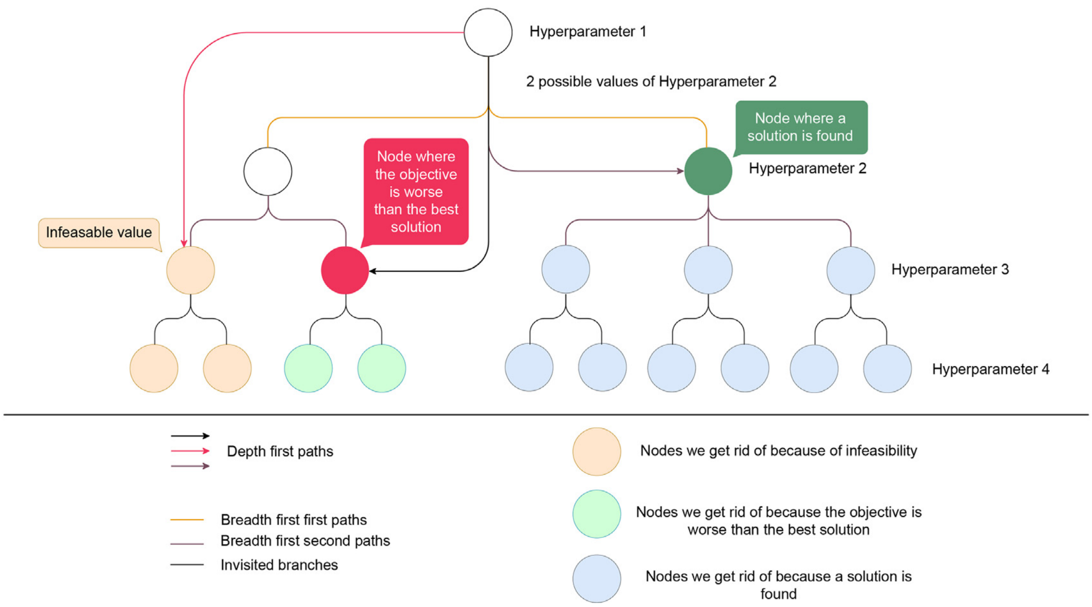

Fig. 3. Branch-and-Bound diagram.

continuous values like the loss rate. Furthermore, because the number of all configurations is proportional to the number of all parameters and the domain values of each parameter, the total number of configurations is enormous. The number of all possible configurations in $\mathcal{C}$ can be computed by the following formula:

$$\mathcal{C} = \prod_{i=1}^{|P|} |D(\mathcal{P}_i)| \tag{8}$$

Example 1. Consider the following configurations, we varied the epochs from 10 to 20,000 with 10 steps, this generates 2000 possibles values of epochs. We varied the loss rate from 0.01 to 0.10 with steps of 0.0005 this will also generate 2000 possible values for the loss rate. The total number of configurations in $\mathcal{C}$ will be 4 million.

As a result, when processing the preceding use case, standard enumeration-based methods such as depth-first search [45], and A* [46] would be bluntly blocked. We offer an effective heuristic based on branch and bound [47] for exploring configurations in $\mathcal{C}$ to tackle this problem. The following are the fundamental operations of the proposed approach (see Fig. 3):

The Branch-and-Bound technique as described is well-known as far as algorithms go and can be used for any discrete optimization problems, combinatorial optimization problems [48] as well as also for any mathematical optimization [49]. The main scope of its use is where keeping track of what can be the lowest bound to date found is and needs to be remembered. This kind of lower bound is used to compare with all possible solutions and keep only one. The new solution or the previously retained solution is considered the new lower bound if and only if it is smaller than the previously retained lower bound. Although one could argue that we are primarily interested in solving minimization problems, any maximization problem can be formulated as a minimization problem by increasing the objective function by -1. Only convex problems have a guaranteed global optimal solution. The branching process involves the construction of a root tree containing all possible solutions to the problem. Then we have the option of performing an exhaustive search (examining every branch of the tree) or eliminating the branches that we know are not solutions and not examining them (this is called pruning). If the problem is convex, we can prune only if one of the following conditions is satisfied (if the problem is not convex, these requirements may force us to prune prematurely and/or miss a local or global optimum):

1. The infeasibility of the value of a variable (a constraint), which is in our case the actual value of any hyperparameter. Therefore, we eliminate any and all of the branches that can be linked to that actual value. Due to the fact that by going down, we are simply adding more constraints, therefore if and only if one it can be shown to be already infeasible, there is really no need to go any further.
2. We finally reach a known objective function that can be shown to be worse than the actual best solution.
3. A solution can be found where going down any further can not help us in finding any better solutions. This is due to the fact that in the end this can be scene as just adding more constraints. Hence, in the end we can just compare the current solution with the best one found to date.

The question here is which variable do we need to branch upon? This is an illustration of a binary problem (a binary problem in this context is an optimization problem where the vector X of the variables x_i for $i = \{1, 2, 3, \ldots, n\}$ is represented in this given form:

$$X \in \{0, 1\}^n \tag{9}$$

where n is the number of variables in the problem) relaxed to:

$$0 \leq x_i \leq 1, for : i = \{1, 2, 3, \ldots, n\} \tag{10}$$

For instance, consider this solution: $X = [0.1, 0.6, 1, 0, 1]$, we make a branch on the variable whose value is closest to 0.5. (in this example it is x_2). We propose two exploration strategies:

1. Depth-first strategy: First climbs down a branch until it is unable to continue go up, and then begins go back up the branch.
2. Breadth-first strategy: At each depth, we examine several branches and then descend further, repeating the procedure until we reach the bottom.

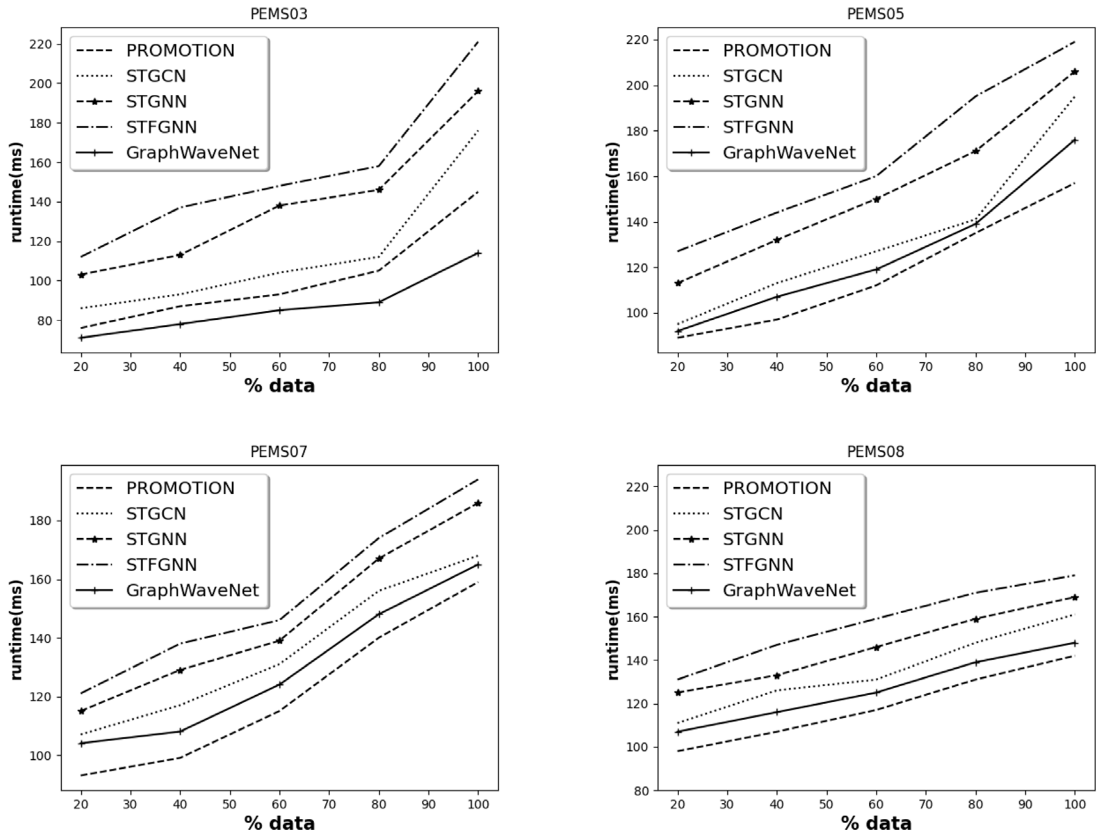

Fig. 4. Runtime of PROMOTION and the baseline solutions for flow forecasting.

4. Performance evaluation

4.1. Environmental settings

Extensive tests for traffic flow forecasting have been carried out to validate the proposed PROMOTION framework using four public traffic network datasets. PEMS03, PEMS04, PEMS07, PEMS08 [50]. They are made up of four districts in California, accordingly. All of this information is gathered via the Caltrans Performance Measurement System (PeMS) and aggregated into 5-minute periods, resulting in 288 traffic flow points in a single day. The spatial adjacency networks for each dataset are built using a distance-based road network. To normalize the data input, Z-score normalization is used. PROMOTION is compared to the following approaches:

1. STGCN [51]: spatio-temporal Graph Convolutional Networks that combines graph convolution with a one-dimensional convolution unit
2. STGNN [52]: Spatial–Temporal Graph Neural Network, which is a complex transformer model with a learnable positional attention mechanism as well as a sequential component.
3. STFGNN [35]: Spatial–Temporal Fusion Graph Neural Networks, which is based on fusion operation of diverse spatial as well as temporal graphs, addressed for different time periods in parallel, and could successfully learn hidden spatial–temporal relationships.
4. GraphWaveNet [53]: It is a framework that blends adaptive adjacency matrices with 1D dilated convolution in graph convolution.

The strategies are implemented on a computer with an Intel Core i7 CPU installed and also coupled with an NVIDIA GeForce GTX 1070 graphics unit (GPU). The benchmark known as mAP (mean Average Precision) is used in evaluating the proposed framework for PROMOTION. Most applications of mAP involve testing various aspects of traffic flow predictions. It is possible that it is defined by:

$$mAP = \frac{\sum_{i=0}^{n} AvgP(i)}{n}, \tag{11}$$

where n is considered as the corrected predicted flows among all flows, and $AvgP(i)$ is calculated as the precision results at i-rank.

4.2. Runtime

The runtime of PROMOTION is compared to the baseline traffic flow forecasting techniques utilizing different ITS sets, the results are shown in Fig. 4. The X-axis depicts the proportion of the data used for each test, while the Y-axis represents the runtime for each test in milliseconds. The results demonstrate that PROMOTION outperforms STGCN, STGNN, and STFGNN and that it is competitive with GraphWaveNet. The proposed solution's runtime is less than the STGCN, STGNN, and STFGNN approaches, whatever the percentage of data used in the experiments that varied from 20% to 100%. Furthermore, PROMOTION outperforms GraphWaveNet in three data collections, while GraphWaveNet exceeds PROMOTION in only one case. The effective graph convolution neural network with the pruning strategy utilized to discover and remove outliers are responsible for these promising results.

4.3. Accuracy

The accuracy of PROMOTION is compared to the baseline traffic flow forecasting techniques utilizing different ITS sets,

Table 1
Mean average precision of PROMOTION and the baseline solutions for flow forecasting.

Data	PROMOTION	STGCN	STGNN	STFGNN	GraphWaveNet
PEMS03(20%)	0.53	0.35	0.38	0.38	0.39
PEMS03(40%)	0.59	0.37	0.39	0.41	0.42
PEMS03(60%)	0.66	0.41	0.42	0.44	0.46
PEMS03(80%)	0.68	0.42	0.45	0.49	0.52
PEMS03(100%)	0.75	0.47	0.51	0.55	0.58
PEMS04(20%)	0.59	0.33	0.36	0.41	0.43
PEMS04(40%)	0.68	0.37	0.42	0.45	0.49
PEMS04(60%)	0.74	0.39	0.47	0.50	0.58
PEMS04(80%)	0.77	0.41	0.51	0.53	0.59
PEMS04(100%)	0.79	0.45	0.55	0.56	0.63
PEMS07(20%)	0.55	0.52	0.50	0.48	0.47
PEMS07(40%)	0.63	0.57	0.55	0.54	0.52
PEMS07(60%)	0.69	0.59	0.59	0.59	0.55
PEMS07(80%)	0.75	0.64	0.62	0.66	0.59
PEMS07(100%)	0.81	0.69	0.65	0.70	0.63
PEMS08(20%)	0.63	0.58	0.57	0.55	0.61
PEMS08(40%)	0.69	0.63	0.64	0.61	0.62
PEMS08(60%)	0.72	0.68	0.69	0.66	0.70
PEMS08(80%)	0.75	0.72	0.70	0.68	0.71
PEMS08(100%)	0.83	0.75	0.74	0.69	0.73

Table 2
Mean average precision of PROMOTION and the advanced baseline solutions for flow forecasting.

Data	PROMOTION	ASTGAT	TVGCN	DSTAGNN	DGGP
PEMS03x1K	0.61	0.58	0.59	0.60	0.57
PEMS03x10K	0.62	0.57	0.58	0.55	0.59
PEMS03x100K	0.73	0.68	0.66	0.69	0.64
PEMS04x1K	0.64	0.62	0.60	0.58	0.57
PEMS04x10K	0.61	0.57	0.59	0.52	0.55
PEMS04x100K	0.68	0.68	0.67	0.66	0.65
PEMS07x1K	0.59	0.57	0.57	0.55	0.56
PEMS07x10K	0.61	0.60	0.59	0.60	0.61
PEMS07x100K	0.62	0.62	0.61	0.60	0.60
PEMS08x1K	0.71	0.55	0.56	0.53	0.62
PEMS08x10K	0.74	0.71	0.70	0.67	0.69
PEMS08x100K	0.72	0.72	0.68	0.65	0.69

the results are shown in Table 1. By varying the percentage of data from 20% to 100%, the results reveal a clear superiority of the PROMOTION against the baseline solutions (STGCN, STGNN, STFGNN, and GraphWaveNet). For instance, the mean average of PROMOTION is 0.75 in dealing with 100% of PEMS03 data, where the mean average precision of the other solutions does not exceed 0.60 for handling the same scenario. This result is reached thanks to the efficient combination between the data pruning, the graph convolution neural network, and the hyper-parameter optimization executed by the branch and bound strategy. Thus, the data pruning allows efficient training of the graph convolution neural network by removing outlier and irrelevant data. In addition, the branch and bound-based optimization allow to retrieval of the best parameters for the training process.

4.4. PROMOTION vs. Advanced traffic flow forecasting solutions on big data

The final experiments aim to compare the accuracy of PROMOTION with more advanced baseline traffic flow prediction methods on big data. The results show that PROMOTION significantly outperforms the baseline solutions (ASTGAT [41], TVGCN [40], DSTAGNN [42], and DGGP [43]) in replicating the original data from 1000 to 100,000, which can be seen in Table 2. For example, for a 100,000 times of the PEMS03 data, the average accuracy of PROMOTION is 0.73, while the average accuracy of the other solutions in the same situation is no more than 0.69. The effective integration of data pruning, graph convolution neural network, and hyperparameter optimization through the branch-and-bound approach has led to this result. By eliminating outliers and unnecessary data, data pruning enables effective training of the graph convolution neural network. Moreover, branch-and-bound based optimization enables the recovery of the ideal training process parameters.

5. Discussions and future directions

This section outlines the main advantages of using the proposed PROMOTION framework for traffic flow forecasting. We also offer some suggestions for how to make the PROMOTION framework better. It is clear that the framework as presented has a high level of novelty and applicability to real-world Internet of Vehicles (IoV) scenarios. However, there are still ways to make the framework operate better.

1. Deep learning, graph pruning, and optimization are an effective mix of smart technologies that create a high level of precision. Runtime efficiency is still an issue when it comes to managing large graphs and forecasting traffic flow in real-time. Creating hybrid systems that combine evolutionary and precise approaches [54] to improve the PROMOTION performance could be an intriguing route to take.
2. Traffic has been effectively estimated using the recommended methods. It outperforms earlier methods for traffic flow forecasting in terms of accuracy. The PROMOTION's findings on other ITS applications, such as trajectory analysis [55,56], vehicle routing problem [57,58], and traffic congestion [59,60] would be fascinating to investigate.
3. When it comes to PROMOTION, it can be difficult to understand the results. This is because they are based on black-box models that do not clearly define how conclusions are drawn from the results. For urban planners to have confidence in a particular outcome, they need to understand how that outcome is arrived at. The field of eXplainable Artificial Intelligence, or XAI, is working to solve this problem. It offers a number of different approaches to provide some level of explanation for deep learning AI solutions. The PROMOTION system will be upgraded to incorporate XAI approaches. This will allow a more accurate evaluation of the results from PROMOTION.

6. Conclusion

We investigated a graph convolution neural network-based framework for urban traffic flow forecasting in an edge environment. It combines graph pre-processing, forecasting, and branch and bound in a single pipeline called PROMOTION. To reduce noise from the set of original road networks of the urban traffic data, preprocessing of the associated graph is initially done. We used the outlier detection strategy to efficiently explore the road network and remove irrelevant patterns and noises. The extended graph convolution neural network is then used to forecast the urban traffic flow from the generated graph. We also designed a new branch-and-bound optimization technique to precisely adjust the hyper-parameter values of PROMOTION. Intensive evaluations are used with different datasets and various baseline methods for comparison. The results reveal the superiority of PROMOTION compared to the baseline solutions in particular when the number of nodes in the graph is considerably large.

CRediT authorship contribution statement

Youcef Djenouri: Conceptualization, Writing – original draft. **Asma Belhadi:** Methodology, Visualization. **Gautam Srivastava:** Formal analysis. **Jerry Chun-Wei Lin:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] T. Li, W. Liu, Z. Zeng, N. Xiong, DRLR: A deep reinforcement learning based recruitment scheme for massive data collections in 6G-based IoT networks, IEEE Internet Things J. (2021).
[2] J. Zhang, Z. Wang, D. Wang, X. Zhang, B.B. Gupta, X. Liu, J. Ma, A secure decentralized spatial crowdsourcing scheme for 6G-enabled network in box, IEEE Trans. Ind. Inf. 18 (9) (2021) 6160–6170.
[3] V. Adat, B.B. Gupta, Security in internet of things: issues, challenges, taxonomy, and architecture, Telecommun. Syst. 67 (3) (2018) 423–441.
[4] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, Future Gener. Comput. Syst. 97 (2019) 219–235.
[5] A. Al-Qerem, M. Alauthman, A. Almomani, B.B. Gupta, IoT transaction processing through cooperative concurrency control on fog–cloud computing environment, Soft Comput. 24 (8) (2020) 5695–5711.
[6] Z. Lv, D. Chen, R. Lou, Q. Wang, Intelligent edge computing based on machine learning for smart city, Future Gener. Comput. Syst. 115 (2021) 90–99.
[7] I. Cvitić, D. Peraković, M. Periša, B. Gupta, Ensemble machine learning approach for classification of IoT devices in smart home, Int. J. Mach. Learn. Cybern. 12 (11) (2021) 3179–3202.
[8] A.M. Rahmani, T.N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach, Future Gener. Comput. Syst. 78 (2018) 641–658.
[9] A. Ferdowsi, U. Challita, W. Saad, Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview, IEEE Veh. Technol. Mag. 14 (1) (2019) 62–70.
[10] I. Ahammad, A.R. Khan, Z.U. Salehin, A review on cloud, fog, roof, and dew computing: IoT perspective, Int. J. Cloud Appl. Comput. 11 (4) (2021) 14–41.
[11] H. Lu, Z. Ge, Y. Song, D. Jiang, T. Zhou, J. Qin, A temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting, Neurocomputing 427 (2021) 169–178.
[12] S. Zhang, Y. Guo, P. Zhao, C. Zheng, X. Chen, A graph-based temporal attention framework for multi-sensor traffic flow forecasting, IEEE Trans. Intell. Transp. Syst. (2021).
[13] R.K.C. Chan, J.M.-Y. Lim, R. Parthiban, A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system, Expert Syst. Appl. 171 (2021) 114573.
[14] H. Wei, G. Zheng, V. Gayah, Z. Li, Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation, ACM Knowl. Manag. Explor. Newsl. 22 (2) (2021) 12–18.
[15] S. Neelakandan, M. Berlin, S. Tripathi, V.B. Devi, I. Bhardwaj, N. Arulkumar, IoT-based traffic prediction and traffic signal control system for smart city, Soft Comput. 25 (18) (2021) 12241–12248.
[16] Z. Lv, Y. Li, H. Feng, H. Lv, Deep learning for security in digital twins of cooperative intelligent transportation systems, IEEE Trans. Intell. Transp. Syst. (2021).
[17] P. Jiang, Z. Liu, L. Zhang, J. Wang, Advanced traffic congestion early warning system based on traffic flow forecasting and extenics evaluation, Appl. Soft Comput. (2022) 108544.
[18] B. Lu, X. Gan, H. Jin, L. Fu, X. Wang, H. Zhang, Make more connections: Urban traffic flow forecasting with spatiotemporal adaptive gated graph convolution network, ACM Trans. Intell. Syst. Technol. (TIST) 13 (2) (2022) 1–25.
[19] L. Deng, D. Lian, Z. Huang, E. Chen, Graph convolutional adversarial networks for spatiotemporal anomaly detection, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[20] A. Almeida, S. Brás, I. Oliveira, S. Sargento, Vehicular traffic flow prediction using deployed traffic counters in a city, Future Gener. Comput. Syst. 128 (2022) 429–442.
[21] Y. Modi, R. Teli, A. Mehta, K. Shah, M. Shah, A comprehensive review on intelligent traffic management using machine learning algorithms, Innov. Infrastruct. Solut. 7 (1) (2022) 1–14.
[22] S. Sun, C. Zhang, G. Yu, A Bayesian network approach to traffic flow forecasting, IEEE Trans. Intell. Transp. Syst. 7 (1) (2006) 124–132.
[23] B.L. Smith, B.M. Williams, R.K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, Transp. Res. C 10 (4) (2002) 303–321.
[24] A. Belhadi, Y. Djenouri, D. Djenouri, J.C.-W. Lin, A recurrent neural network for urban long-term traffic flow forecasting, Appl. Intell. 50 (10) (2020) 3252–3265.
[25] H. Zeng, Z. Peng, X. Huang, Y. Yang, R. Hu, Deep spatio-temporal neural network based on interactive attention for traffic flow prediction, Appl. Intell. (2022) 1–12.
[26] C. Ma, Y. Zhao, G. Dai, X. Xu, S.-C. Wong, A novel STFSA-CNN-gru hybrid model for short-term traffic speed prediction, IEEE Trans. Intell. Transp. Syst. (2022).
[27] F. Harrou, A. Dairi, F. Kadri, Y. Sun, Effective forecasting of key features in hospital emergency department: Hybrid deep learning-driven methods, Mach. Learn. Appl. 7 (2022) 100200.
[28] J. Zhai, F. Boukouvala, Data-driven spatial branch-and-bound algorithms for box-constrained simulation-based optimization, J. Global Optim. 82 (1) (2022) 21–50.
[29] J. Hou, L. Yu, S. Fei, A high-quality voxel 3D reconstruction system for large scenes based on the branch and bound method, Expert Syst. Appl. (2022) 116549.
[30] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano, J.C.-W. Lin, A two-phase anomaly detection model for secure intelligent transportation ride-hailing trajectories, IEEE Trans. Intell. Transp. Syst. 22 (7) (2020) 4496–4506.
[31] Y. Djenouri, A. Belhadi, J.C.-W. Lin, D. Djenouri, A. Cano, A survey on urban traffic anomalies detection algorithms, IEEE Access 7 (2019) 12192–12205.
[32] A. Belhadi, Y. Djenouri, G. Srivastava, A. Cano, J.C.-W. Lin, Hybrid group anomaly detection for sequence data: application to trajectory data analytics, IEEE Trans. Intell. Transp. Syst. (2021).
[33] C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 22 (3) (2021) 1840–1852.
[34] G. Srivastava, J.C.-W. Lin, A. Jolfaei, Y. Li, Y. Djenouri, Uncertain-driven analytics of sequence data in IoCV environments, IEEE Trans. Intell. Transp. Syst. (2020).
[35] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4189–4196.
[36] W. Fang, W. Zhuo, J. Yan, Y. Song, D. Jiang, T. Zhou, Attention meets long short-term memory: A deep learning network for traffic flow forecasting, Physica A 587 (2022) 126485.
[37] Z. Chen, X. Luo, T. Wang, W. Wang, W. Zhao, Deep reinforcement learning-based LSTM model for traffic flow forecasting in internet of vehicles, in: Proceedings of 2021 Chinese Intelligent Automation Conference, Springer, 2022, pp. 515–523.
[38] Q. Ouyang, T. Sun, Y. Xue, Z. Liu, Long short-term memory and graph convolution network for forecasting the crude oil traffic flow, IEEE Access (2022).
[39] C. Raskar, S. Nema, Metaheuristic enabled modified hidden Markov model for traffic flow prediction, Comput. Netw. (2022) 108780.
[40] Y. Wang, S. Fang, C. Zhang, S. Xiang, C. Pan, TVGCN: Time-variant graph convolutional network for traffic forecasting, Neurocomputing 471 (2022) 118–129.
[41] Y. Wang, C. Jing, S. Xu, T. Guo, Attention based spatiotemporal graph attention networks for traffic flow forecasting, Inform. Sci. 607 (2022) 869–883.
[42] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, P. Li, DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting, in: International Conference on Machine Learning, PMLR, 2022 pp. 11906–11917.
[43] Y. Jiang, J. Fan, Y. Liu, X. Zhang, Deep graph Gaussian processes for short-term traffic flow forecasting from spatiotemporal data, IEEE Trans. Intell. Transp. Syst. (2022).
[44] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM International Conference on Management of Data, 2000, pp. 93–104.
[45] J. Wang, Y. Xie, S. Xie, X. Chen, Cooperative particle swarm optimizer with depth first search strategy for global optimization of multimodal functions, Appl. Intell. (2022) 1–20.
[46] Z. Bu, R.E. Korf, A*+ BFHS: A hybrid heuristic search algorithm, 2021, arXiv preprint arXiv:2103.12701.

[47] Y. Zhou, W. Lin, J.-K. Hao, M. Xiao, Y. Jin, An effective branch-and-bound algorithm for the maximum s-bundle problem, European J. Oper. Res. 297 (1) (2022) 27–39.
[48] K. Fleszar, A branch-and-bound algorithm for the quadratic multiple knapsack problem, European J. Oper. Res. 298 (1) (2022) 89–98.
[49] H. Hazimeh, R. Mazumder, A. Saab, Sparse regression at scale: Branch-and-bound rooted in first-order optimization, Math. Program. (2021) 1–42.
[50] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, in: Proceedings of the Conference on Artificial Intelligence, Vol. 34, 2020, pp. 914–921.
[51] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2017, arXiv preprint arXiv:1709.04875.
[52] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, J. Yu, Traffic flow prediction via spatial temporal graph neural network, in: Proceedings of the Web Conference, 2020, pp. 1082–1092.
[53] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, 2019, arXiv preprint arXiv:1906.00121.
[54] Y. Djenouri, M. Comuzzi, Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem, Inform. Sci. 420 (2017) 1–15.
[55] Y. Djenouri, D. Djenouri, J.C.-W. Lin, Trajectory outlier detection: New problems and solutions for smart cities, ACM Trans. Knowl. Discov. Data 15 (2) (2021) 1–28.
[56] C. Cheng, L. Guo, T. Wu, J. Sun, G. Gui, B. Adebisi, H. Gacanin, H. Sari, Machine learning-aided trajectory prediction and conflict detection for internet of aerial vehicles, IEEE Internet Things J. (2021).
[57] R. Kuo, S.-H. Lu, P.-Y. Lai, S.T.W. Mara, Vehicle routing problem with drones considering time windows, Expert Syst. Appl. 191 (2022) 116264.
[58] P.C. Nolz, N. Absi, D. Feillet, C. Seragiotto, The consistent electric-vehicle routing problem with backhauls and charging management, European J. Oper. Res. (2022).
[59] Z. Kan, M.-P. Kwan, D. Liu, L. Tang, Y. Chen, M. Fang, Assessing individual activity-related exposures to traffic congestion using GPS trajectory data, J. Transp. Geogr. 98 (2022) 103240.
[60] S. Luan, R. Ke, Z. Huang, X. Ma, Traffic congestion propagation inference using dynamic Bayesian graph convolution network, Transp. Res. C 135 (2022) 103526.

Youcef Djenouri obtained the Ph.D. in Computer Engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2014. From 2014 to 2015, Dr Djenouri was a permanent teacher-researcher at the university of Blida, in Algeria, where he was member of LRDSI Lab. He was granted a post-doctoral fellowship from the Unist university on South Korea, and he worked on BPM project supported by Unist university in 2016. In 2017, he was postdoctoral research at Southern Denmark University, where he ha working on urban traffic data analysis. Currently, he is granted a post-doctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM), and he worked at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. He is working on topics related to artificial intelligence and data mining, with focus on pattern mining, parallel computing, swarm and evolutionary algorithms. He has published more than 45 international conference and journal papers, 2 book chapter, and 1 tutorial paper in the areas of data mining, parallel computing and artificial intelligence.

Asma Belhadi obtained the Ph.D. in Computer Engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2016. She is a postdoctoral researcher at Kristiania University College in Oslo, Norway. She is working on topics related to artificial intelligence and data mining, with focus on logic programming. Dr. Belhadi has published over 25 refereed research articles in the areas of artificial intelligence, and smart city applications.

Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in the U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2012, respectively. He then taught for 3 years at the University of Victoria in the Department of Computer Science, where he was regarded as one of the top undergraduate professors in the Computer Science Course Instruction at the University. From there in the year 2014, he joined a tenure-track position at Brandon University in Brandon, Manitoba, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank of Associate Professor in January 2018. Dr. G, as he is popularly known, is active in research in the field of Cryptography, Data Mining, Security and Privacy, and Blockchain Technology. In his 5 years as a research academic, he has published a total of 140 papers in high-impact conferences in many countries and in high status journals (SCI, SCIE) and has also delivered invited guest lectures on Big Data, Cloud Computing, Internet of Things, and Cryptography at many universities worldwide. He is an Editor of several SCI/SCIE journals. He currently has active research projects with other academics in Taiwan, Singapore, Canada, Czech Republic, Poland, and the U.S.A. He is an IEEE Senior Member and also an Associate editor of the world renowned IEEE Access journal.

Jerry Chun-Wei Lin received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is currently a full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 300 research articles in refereed journals (IEEE TKDE, IEEE TCYB, ACM TKDD, ACM TDS, ACM TMIS) and international conferences (IEEE ICDE, IEEE ICDM, PKDD, PAKDD). His research interests include data mining, soft computing, artificial intelligence and machine learning, and privacy preserving and security technologies. He is also the project co-leader of well-known SPMF: An Open-Source Data Mining Library, which is a toolkit offering multiple types of data mining algorithms. He also serves as the Editor-in-Chief of the International Journal of Data Science and Pattern Recognition. He is the IET Fellow, senior member for both IEEE and ACM.