Computer Communications 189 (2022) 175–181

Contents lists available at ScienceDirect

Computer Communications

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

Intelligent deep fusion network for urban traffic flow anomaly identification

Youcef Djenouri [a], Asma Belhadi [b], Hsing-Chung Chen [c], 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 Computer Science & Information Engineering, Asia University, Taiwan
[d] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway

ARTICLE INFO

Keywords:
Convolution neural network
Decomposition
Urban traffic flow data

ABSTRACT

This paper presents a novel deep learning architecture for identifying outliers in the context of intelligent transportation systems. The use of a convolutional neural network with an efficient decomposition strategy is explored to find the anomalous behavior of urban traffic flow data. The urban traffic flow data set is decomposed into similar clusters, each containing homogeneous data. The convolutional neural network is used for each data cluster. In this way, different models are trained, each learned from highly correlated data. A merging strategy is finally used to fuse the results of the obtained models. To validate the performance of the proposed framework, intensive experiments were conducted on urban traffic flow data. The results show that our system outperforms the competition on several accuracy criteria.

1. Introduction

Urban traffic flow data have recently piqued the curiosity of researchers. [1–3], in particular, numerous deep learning and computer vision systems [4–6] have been implemented to analyze and understand urban traffic flow data in the context of intelligent transportation, and smart city based applications.

Urban traffic flow data consists of observations such as the number and speed of cars or other vehicles at specific locations, as determined by installed sensors. These numbers represent the flow of traffic, which is related to the capacity of roads and the demand on the transportation system. Urban planners are interested in the effects of various factors on traffic flow that result in unexpected patterns called outliers. In addition, we hope to learn from the behavior of independent participants (bicyclists, cars, trucks, and public transit) under different conditions (weather, events, road maintenance) to help urban planners and managers make decisions about roadway design, regulatory systems (e.g., traffic signals), and public transit routes, as well as temporary invasive building placement decisions.

1.1. Motivation

Existing techniques for detecting anomalies in urban traffic flow data [7,8] consider the entire urban traffic flow data for building the learning models. This degrades the overall performance of such approaches, especially for large and diversified urban traffic flow data where traffic flow variations are high throughout the year. Several works have been developed to deal with heterogeneous urban traffic flow data [9,10]. Even the advanced deep learning architectures with more layers show great improvement, but still suffer from both accuracy and inference runtime. Recently, the decomposition strategies have attracted great interest in the deep learning community [11–13]. The idea is to partition the heteregenous training space and work with more focused regions, and homogeneous sub-spaces. This research work follows this direction and explores decomposition for outlier detection in traffic flow.

1.2. Contribution

This paper addresses the shortcomings of existing methods in the literature for detecting outliers in urban traffic flow data processing and proposes a new framework for processing large and comprehensive data to find anomalies from urban traffic flow data. The main contributions of the paper are listed below:

1. We propose an efficient improvement of the convolutional neural network by exploring the decomposition strategy to partition the data into similar clusters.
2. We develop a merging strategy to fuse the different outputs of the trained models and accurately detect outliers.
3. We evaluate the proposed framework on two urban traffic flow data and with different metrics. The results show a clear superiority of the proposed system compared to the baseline methods.

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

https://doi.org/10.1016/j.comcom.2022.03.021
Received 31 October 2021; Received in revised form 11 February 2022; Accepted 27 March 2022
Available online 4 April 2022

1.3. Outline

The remainder of the article is organized as follows. Section 2 discusses known strategies for detecting anomalies in an intelligent transportation system. Section 3 summarizes the proposed strategy and its main components. Section 4 summarizes the experimental setup and results. Section 5 summarizes the main results of applying the proposed methodology to urban traffic flow data. Section 6 gives an outlook on the future development of the proposed framework. Finally, Section 7 concludes the paper.

2. Related work

Algorithms for detecting outliers in urban traffic can be divided into different classes. Classical methods, which use traditional machine learning methods, and advanced methods, which are based on deep learning. In this section we will discuss solutions of both classes.

2.1. Classical machine learning based solutions

Ngan et al. [14] study the Chinese restaurant process to create an endless number of clusters for the flow values. All the flow values belonging to the cluster with the most elements are considered as inliers, while the other flow values are considered as outliers. Gu et al. [15] proposed an intelligent model for passenger flow anomalies. First, a hybrid k-means and hierarchical clustering algorithm is performed to identify the passenger flow represented by time series data. Anomaly detection indices are generated to represent the different types of outliers in the passenger flow. The different detected anomalies are reported to the city planners as alarms.

Lin et al. [16] develop a method for predicting road traffic speeds based on Gaussian aggregation. To augment the training data, speed measurement data is first combined with tweet and trajectory data. The entire framework is then built using a mixture of a disaggregation model and a Gaussian process. Munoz-Organero et al. [17] present a method to filter out driving locations associated with unpredictable traffic situations, such as congestion, from infrastructure road features. Mahalanobis distance is used to determine the similarity of individual traffic flows recorded at each second within different time frames. Shi et al. [18] proposed a dynamic neighborhood-based technique to identify local anomalies in spatiotemporal traffic flow data. The dynamic flow is first represented by the real-time vehicle speed data. Then, the dynamic neighborhood structure is established by calculating the similarity of the spatio-temporal flows.

2.2. Advanced deep learning based solutions

Nguyen et al. [19] proposed two stepwise solutions for vehicle anomaly detection. The process starts with vehicle detection using multiple adaptive vehicle algorithms. The detected vehicles are trained in a convolutional neural network to identify anomalous events. Bai et al. [20] developed a three-stage algorithm for traffic anomaly detection. First, traffic flow is analyzed to determine both road segmentation and stationary regions. The perspective map is then generated from the data obtained in the first step. Finally, the spatio-temporal information matrix is created to identify all anomalies.

Zhu et al. [21] developed a convolutional neural network to find anomalies in urban traffic. The image database is generated from traffic data observations, with each image representing a particular state of the traffic situation in the city. The convolutional neural network is used to identify anomalies, and each image is classified into two categories: normal or abnormal traffic flow. Huang et al. [22] studied the causality of traffic in a large urban network. They consider a visible outlier index as a probabilistic indicator of traffic anomalies by injecting the spatiotemporal anomalies into the deep autoencoder learning model. The proposed solution is able to detect contours around the zones that cause anomalies in the network, which helps urban planners to get an accurate understanding of such areas.

Based on this brief literature review, urban traffic flow anomaly detection strategies can be divided into two groups (classical and advanced). The classical methods can be divided into two categories. The first category is statistical methods, where inlier flows are assumed to follow a common statistical process, while outlier flows deviate from this statistical mechanism. The second group of approaches is based on similarity. These approaches use distance measurements and methods for computing neighborhoods, as well as traditional methods for identifying outliers cited in [23,24]. Normal flows are assumed to produce dense regions, while deviant flows produce sparsely populated regions. Statistical methods are extremely sensitive to outliers, i.e., outliers affect the fit of the model. In addition, they rely on a particular statistical model, and it is often unclear whether or not that model accurately represents the true distribution of traffic flow data. By using a non-parametric approach, similarity-based techniques solve the latter problem. However, they are very sensitive to the distance used to calculate the neighborhoods. The advanced methods are based on deep learning and use different deep architectures such as recurrent neural networks, convolutional neural networks, and autoencoder models to solve the problems of the classical solutions.

However, these systems still have a low detection rate because the entire database is considered/used in the learning process, even in distributed environments. Moreover, tuning the hyperparameters of deep learning models is not trivial. Given the success of cluster-based algorithms [25,26], in the next section, we present a hybrid fusion technique that combines decomposition and CNN to efficiently explore traffic flow data in a distributed environment system.

3. DCNN-TFO: Deep convolution neural network for traffic flow outliers

In this section, we present a novel deep learning algorithm for identifying anomalies in urban traffic data. It is a convolutional neural network that takes images as input and is applied to urban traffic data. As shown in Fig. 1, the process begins by collecting data from traffic sensors and generating traffic flow data. These are decomposed to create homogeneous clusters. The input of the convolutional neural network is the clusters generated by the decomposition algorithm. As a result, different models are generated, each of which is associated with a cluster of the urban traffic data. The main difference between the proposed convolutional neural network and the generic networks is that the proposed architecture is able to handle large and heterogeneous data. This is very difficult to achieve without decomposition. In the following, we describe the main steps of the DCNN-TFO, followed by its pseudocode shown in Algorithm 1.

3.1. Decomposition

Decomposition is first used to divide the set of urban traffic flows into similar clusters. This allows the deep learning models to be easily trained with homogeneous data. The goal is to determine a set of clusters that we can assign to each urban traffic flow while minimizing the distances between the urban traffic flow of the given cluster and its centroid. We need to optimize the following function:

$$argmin \sum_{i=1}^{k} \sum_{F_j \in C_i} D(F_j, g_i), \tag{1}$$

where F_j is the urban traffic flow data, C_i is the ith cluster, and g_i is the centroid of the C_i.

To solve this problem, we use the k-means heuristic. We start by randomly initializing the centroids of the clusters. The distance between the centroids and each urban traffic flow data is determined, and these urban traffic flow data are assigned to the cluster with the smallest distance value. After assigning all traffic flow data, the centroid of each

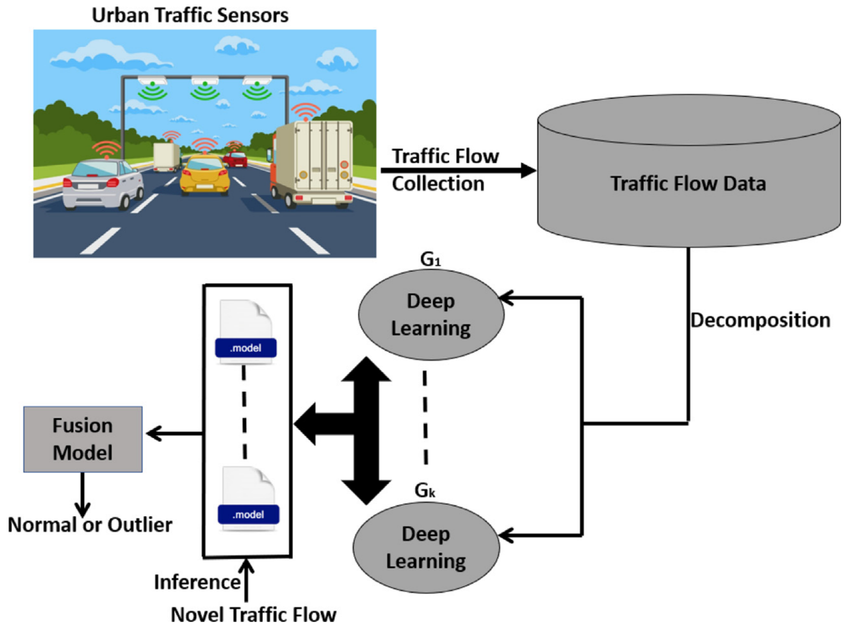

Fig. 1. DCNN-TFO framework.

cluster is updated. This process is repeated until convergence or the maximum number of iterations is reached. Convergence is achieved if and only if the value of the convergence criterion described by Eq. (2) is less than a given threshold.

$$Convergence = \sum_{i=1}^{k} \sum_{F_j \in C_i} D(F_j, g_i) \tag{2}$$

3.2. Convolution neural network

This phase identifies anomalies in the input image data. We were inspired by the Faster RCNN concept, which is widely considered to be the most advanced method for object detection [27]. In our context, the objects to be identified are anomalies from urban traffic data. Faster RCNN mainly consists of the following steps:

1. Region Proposal Determination: This phase identifies the regions of interest, or possible areas marked by bounding boxes, to which the element can be assigned. The traditional RCNN [28] uses the selective search technique described in [29]. This technique generates a large number of bounding boxes for each image, making the whole process tedious and memory intensive. Faster RCNN is a more efficient approach to determine the bounding boxes by using a convolutional neural network. The neural network is then used to suggest bounding boxes using the training images as a basis.
2. Fast RCNN: This stage focuses on classifying parts of images as objects and refining their boundaries. Classification and regression techniques are also used in this step.

In this study, we extend the Faster RCNN to detect anomalies in urban traffic data. First, we train the Faster RCNN model using transfer learning on each cluster of urban traffic data obtained in the previous phase. We train our Faster RCNN on the Imagenet dataset[1] and then apply the trained model to the urban traffic data clusters. This step generates hard negatives that are used to augment the models during the training process. The combination of multiscale training and feature chaining is used to improve the performance of the trained model. Our adaptation is described as follows:

1. Feature Concatenation: Faster RCNN performs region-of-interest pooling at the last feature map level to create region-specific features. This technique is insufficient and omits several critical elements, resulting in a loss of accuracy. To solve this problem, we combine the feature maps of many convolutional layers with features from different layers. We follow the same idea [30] by concatenating and rescaling the pooling results of multiple feature maps using L2 normalization to obtain the final pooling features for recognition tasks.
2. Hard Negative Mining: This method identifies hard negatives, i.e., places where the models make incorrect predictions. To improve the performance of our models, we incorporate hard negatives into them via reinforcement learning. During the second iteration of our training process, we extract hard negatives, where a region is called a hard negative if its intersection with the ground truth region is less than 40%.
3. Multi-Scale Training: The traditional Faster RCNN generates bounding boxes at a fixed scale. In real-world applications, such as urban traffic data, the objects to be detected have a range of sizes. In this study, we investigate five different scales for bounding boxes (tiny, small, medium, large, and very large) to detect objects of different sizes. Thus, five different groups are formed, each consisting of identical bounding boxes. In this context, for each group of bounding boxes, the process of determining the region is initiated. At the end of this step, the created bounding boxes are combined with the convolutional neural network for classification and regression.

3.3. Fusion model

The goal of this step is to merge the results obtained from the models. A voting strategy is used to find the final result of the proposed framework. We assume k different models $\{M_1, M_2, \dots, M_k\}$, where each model M_i gives an output O_i indicating for a given input whether it is an outlier or not. We assume that O_i is set to 1 if the model M_i considers the current input to be an outlier, and 0 otherwise. We first sum all the outputs of the k models shown in Eq. (3):

$$O = \sum_{i=1}^{k} O_i \tag{3}$$

[1] http://www.image-net.org/.

If O is less than $\frac{k}{2}$, then the current input is considered normal urban traffic flow data, otherwise it is considered an outlier. Eq. (3) is the aggregation of the results of all trained models. There is no guarantee that the correct output will be found, but our aim is to minimize the model errors and converge to the correct result. This is implemented using Eq. (3), where the decision is made by a voting mechanism. The decision will be the most the common decision of all models.

Algorithm 1 DCNN-TFO Algorithm

Input: $F = \{F_1, F_2, \dots, F_n\}$: the set of n traffic flow used for the training stage. $F' = \{F'_1, F'_2, \dots, F'_m\}$: the set of m traffic flow used in the inference stage.
Output: $best\_models$: the best models generated in the training phase. O: the set of outliers from the traffic flow in F'.
$G \leftarrow kmeans(F)$;
$HNM \leftarrow HardNegativeMining(F)$;
$FC \leftarrow FeatureConcatenation(F)$;
$MST \leftarrow MultiScaleTraining(F)$;
$best\_models \leftarrow \emptyset$;
for G_i in G **do**
    $best\_model \leftarrow FasterRCNN(G_i, HNM, FC, MST)$;
    $best\_models \leftarrow best\_models \cup best\_model$;
end for
$O \leftarrow \emptyset$;
for $F'_i \in F'$ **do**
    $O_i \leftarrow 0$;
    for $best\_model \in best\_models$ **do**
        $result \leftarrow best\_model(F'_i)$;
        $O_i \leftarrow O_i + result$;
    end for
    if $O_i > \frac{|G|}{2}$ **then**
        $O \leftarrow O \cup O_i$;
    end if
end for
return $(best\_models, O)$.

3.4. Pseudo-code

Algorithm 1 shows the pseudocode of the developed DCNN-TFO. The input data is the set of n traffic flow used for training. This data is accompanied by the ground truth. Each ground truth indicates whether a particular traffic flow in the training data is an outlier or a normal traffic flow. To test the trained models, we used a set of m traffic flows that also have ground truth. In this way, the accuracy of the developed models can be calculated. The results are the best trained models and the outliers identified from the testing set. The process begins by creating clusters of training data using the k-means algorithm. The models are then trained, with each model using a previously created cluster of traffic flow data. As a result of the training phase, the weights of the best trained models, noted $best\_models$, are stored. In the inference phase, the weights of the $best\_models$ for each test data are propagated to identify the anomalies. The fusion operation is performed to calculate the outlier score and then decide whether the current traffic flow is an outlier or not. The algorithm returns the best models as well as the set of outliers of the test data. We note that the training phase, which is performed only once regardless of the amount of data in the inference, is a very time-consuming task that involves multiple optimizations. However, the inference step contains only one loop and requires simple propagation of the models learned in the training phase.

4. Performance evaluation

The proposed framework was validated through extensive experimentation. Two types of data were used:

1. Odense data: It is a real urban traffic data retrieved from Odense Kommune (Denmark).[2] The data are organized as a series of rows, each of which provides information about the automobiles identified at specific locations. The traffic data are from observations of traffic flow in Odense between 1st January 2017 and 30th April 2018, and include more than 12 million worth of cars and motorcycles.
2. Beijing data: It is a recent urban traffic data obtained from the Beijing traffic flow.[3] They include about 900 million of traffic flow values during a two-month period at a single site.

All implementations were performed on a computer equipped with a 64-bit Core i7 CPU running Windows 10 and 16 GB RAM and an Nvidia Tesla C2075 GPU with 448 CUDA cores (14 multiprocessors with 32 cores each) and a clock speed of 1.15 GHz. It has a total memory capacity of 2.8 GB, a shared memory capacity of 49.15 KB and a warp size of 32. Simple precision is used on both CPU and the GPU. It is evaluated using numerous metrics, including True Positive Rate (TPR), True Negative Rate (TNR), and Area Under Curve (AUC), all of which are commonly used to evaluate outlier detection systems. The TPR and TNR metrics are described below:

$$TPR = \frac{TP}{TP + FP} \tag{4}$$

$$TNR = \frac{TN}{TN + FP} \tag{5}$$

Note that TP is the number of true positive, FP is the number of false positive, and TN is the number of true negatives. To compute AUC, we used TNR, and TPR values, and we calculate the cumulative sums of positives (or negatives) along the sorted outliers and the normal flows.

The first experiment aimed to adjust the number of clusters of the DCNN-TFO. Intensive experiments were performed by varying the number of clusters from 1 to 100. The results show that the accuracy of DCNN-TFO increases to stabilization at 35 clusters for the Odense data and 76 for the Beijing data. Therefore, we set up 35 clusters for the data from Odense and 76 clusters for the data from Beijing in the remaining experiments. For comparison, we also ran several tests with data from Odense and Beijing. We varied the amount of urban traffic data used from 20% to 100% and applied the various measures mentioned above (TPR, TNR, and AUC). For comparison, we used two recent algorithms as baseline methods. The first is an improved version of CNN (Convolution Neural Network) [31], a deep learning network for identifying outliers, and the second combines both SVM (Support Vector Machine) and transfer learning [32] to infer the outliers. These algorithms show high accuracy in identifying outliers in urban traffic. Fig. 2 presents the true positive rate by varying the percentage of the data used as input from 20% to 100% on both Odense and Beijing data. The results reveal clear superiority of the proposed DCNN-TFO compared to CNN, and SVM. Indeed, the TPR of the DCNN-TFO does not go below 0.75, and exceed 0.85, where CNN does not exceed 0.82, and SVM does not exceed 0.70. Fig. 3 presents the true negative rate by varying the percentage of the data used as input from 20% to 100% on both Odense and Beijing data. The results reveal again clear superiority of the proposed DCNN-TFO compared to CNN, and SVM. Indeed, the TNR of the DCNN-TFO does not go below 0.80, and exceed 0.90, where CNN does not exceed 0.80, and SVM does not exceed 0.75. In terms of AUC, the results are shown in Fig. 4, the results validate the obtained ones in the previous experiments, where superiority of the DCNN-TFO is validated on both data (Odense, and Beijing). Responsible for these encouraging results is the efficient decomposition approach, which allows the whole data to be divided into homogeneous clusters. This allows to better train the different models of the generated clusters. SVM and CNN-based solutions, which process all the data at once, do not ensure this process.

[2] https://www.odense.dk/.
[3] https://www.beijingcitylab.com/.

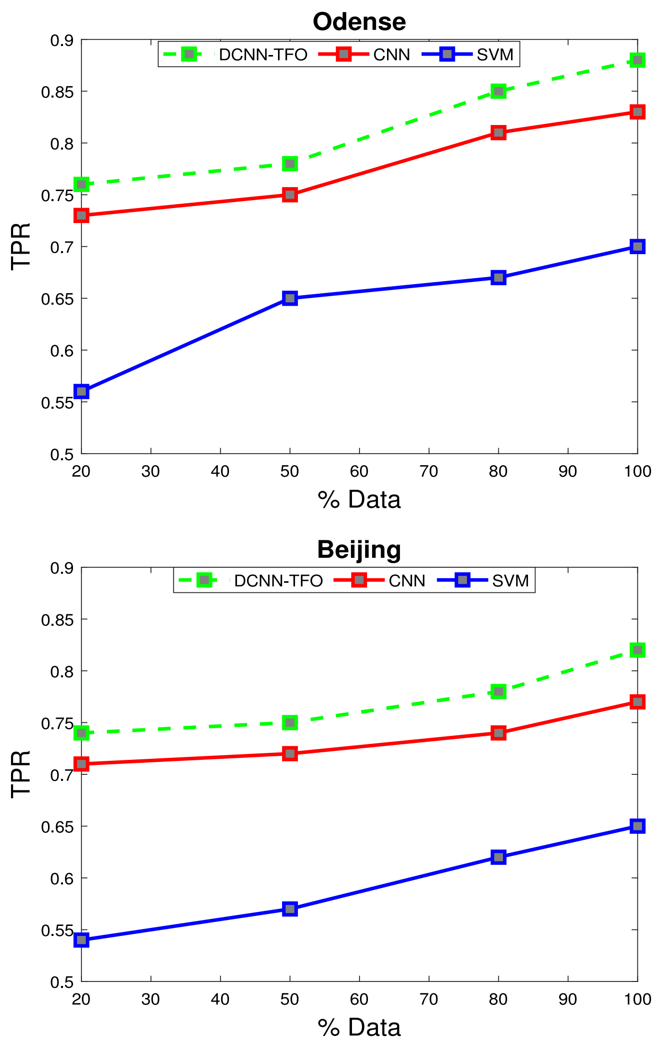

Fig. 2. TPR performance.

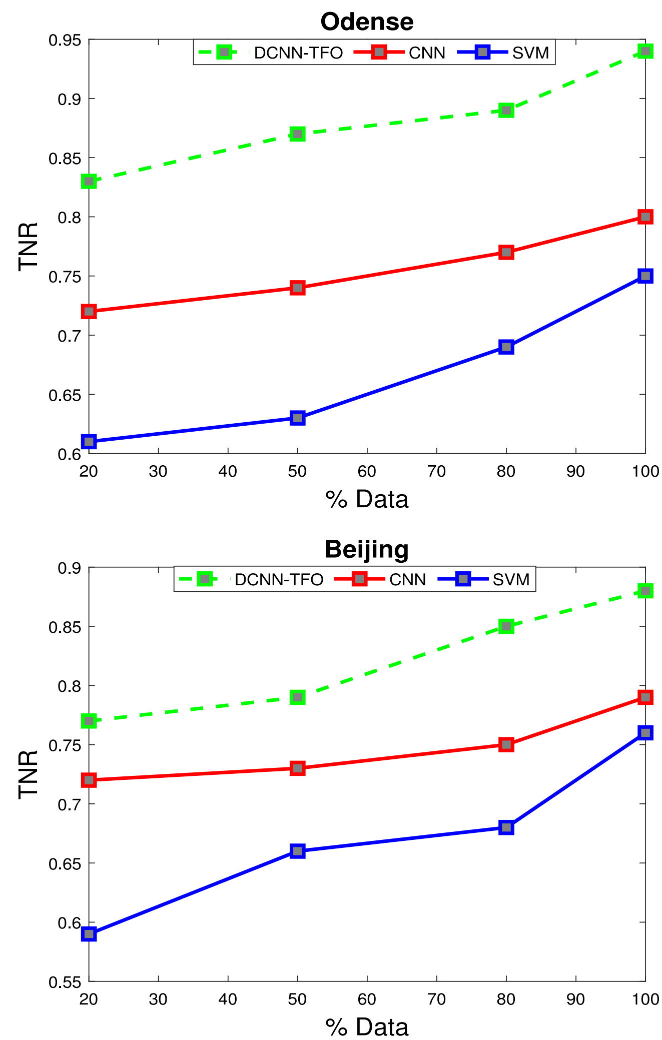

Fig. 3. TNR performance.

5. Discussions

This section summarizes the main findings from the application of decomposition methods and deep learning to the problem of identifying anomalies in urban traffic.

- The first discovery of the study is that the proposed framework is able to handle large urban traffic datasets, such as those from Beijing. This is in contrast to previous anomaly detection techniques that require long execution times and evaluate the entire traffic flow database throughout the outlier detection process. The proposed framework is not only able to derive outliers from urban traffic data, but also to explore the different correlations between urban traffic data and identify different groups within these data [33]. We argue that the inclusion of decomposition approaches during the pre-processing phase enables rapid derivation and identification of outliers.
- DCNN-TFO is an example of integrating data mining and machine learning research. In our scenario, artificial intelligence is combined with outlier identification to manage large amounts of urban traffic data and accelerate the mining process. This adaptation is done in stages, including decomposition and learning processes.
- In addition, this study found that Deep Learning models benefit from pre-treatment of data through decomposition. Since each model uses comparable input, the recognition process is accelerated.
- The last remark is that the framework is general and can be used for any type of network data, unlike previous methods that are limited to specific types of urban traffic data. The type of data shown in this paper is just an illustration of how our framework could be used. Our approach can also be used to solve different types of urban traffic data, such as trajectories [34,35], time series [36], and other [37,38].

6. Future perspectives

Different paths may be studied in light of the promising results presented in this paper:

1. **Improving the decomposition step:** k-means algorithm examined in this research paper is only example of decomposition technique in action. Numerous attempts should be made to reduce and minimize the number of clusters sharing urban traffic data. Therefore, incorporating additional decomposition techniques, such as entity resolution and/or record linkage [39,40] and/or the genetic algorithm [41], into the DCNN-TFO framework is a possibility for the near future. A system for automatic cluster size adjustment should also be developed. Multiple runs to determine the optimal number of clusters is not an efficient

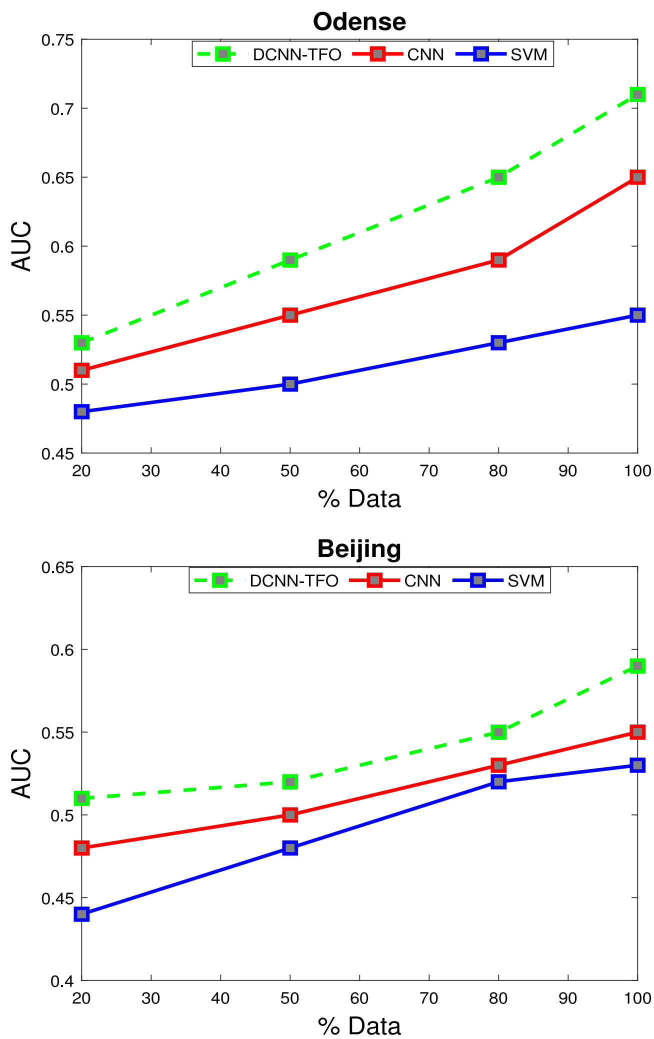

Fig. 4. AUC performance.

method in practice. One way to solve this problem is to create a knowledge base that contains each training set of urban traffic data along with the optimal number of clusters, and then examine the correlation between the meta-features of the urban traffic data (number of flow values, number of trajectories, etc.) and the optimal number of clusters. This allows automatic prediction of the optimal number of clusters in fresh urban traffic data.

2. **Improving the deep learning step:** By using high-performance computing technologies such as GPUs [42], supercomputers [43], and cluster computing [44], we aim to increase the performance of DCNN-TFO and apply it to large-scale applications. The goal in this context is to set up autonomous jobs for each cluster of urban traffic data, while also addressing the difficulties of high-performance computing such as synchronization, communication, memory management, and load balancing. In this context, strategic considerations of load balancing solutions are also on our future agenda. One way to overcome this problem is to develop decomposition algorithms that allow the identification of equivalent clusters based on the amount of urban traffic data in each cluster. Another way is to find new ways to repair clusters, such as identifying clusters with approximately the same amount of urban traffic data.
3. **Case studies:** We have already presented a case study of a DCNN-TFO application in intelligent transportation in this publication. Based on the promising results shown in this case study, we intend to adapt DCNN-TFO to solve domain-specific complicated problems that require the management of large amounts of data. For example, this can be seen in the context of business intelligence applications [45] or financial data mining [46]. In automated trading systems, runtime speed is critical, as profits are sometimes made by exploiting the volatility of stock or currency prices in extremely short time intervals. Outlier detection algorithms capable of detecting divergent patterns are critical in these circumstances, as they enable the discovery of new opportunities for smarter trading. Another potential application is the processing of sensor data, especially for real-time applications related to Internet of Things systems, such as smart cities and related services [9,47,48], energy management in smart buildings [49], smart environment [50], and intrusion detection [51,52] where the process of outlier detection must be performed with a very low latency.

7. Conclusion

In this paper, we explored decomposition and deep learning to accurately retrieve the abnormal behavior of urban traffic flow data. The data is first divided into clusters, each of which contains similar data. This makes the training process of the convolutional neural network simpler and more oriented to homogeneous behaviors. As a result of this combination, several models are trained, each representing the data of the corresponding cluster. A fusion model is developed to combine the results of the trained models. Extensive testing was performed to improve the validation procedure of the proposed framework. The results obtained using urban traffic data show that the proposed approach outperforms state-of-the-art outlier identification algorithms using many metrics.

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.

References

[1] M. Al-Khafajiy, S. Otoum, T. Baker, M. Asim, Z. Maamar, M. Aloqaily, M. Taylor, M. Randles, Intelligent control and security of fog resources in healthcare systems via a cognitive fog model, ACM Trans. Internet Technol. (2020).
[2] T. Wang, Z. Zheng, A.K. Bashir, A. Jolfaei, Y. Xu, FinPrivacy: A privacy-preserving mechanismfor fingerprint identification, ACM Trans. Internet Technol. (2020).
[3] F.-M. Plaza-Del-Arco, M.D. Molina-González, L.A. Ureña-López, M.T. Martín-Valdivia, Detecting misogyny and xenophobia in spanish tweets using language technologies, ACM Trans. Internet Technol. 20 (2) (2020) 1–19.
[4] D. Colucci, L. Morra, X. Zhang, D. Fissore, F. Lamberti, An automatic computer vision pipeline for the in-line monitoring of freeze-drying processes, Comput. Ind. 115 (2020) 103184.
[5] M. Nagel, M.v. Baalen, T. Blankevoort, M. Welling, Data-free quantization through weight equalization and bias correction, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1325–1334.
[6] H. Rebecq, R. Ranftl, V. Koltun, D. Scaramuzza, Events-to-video: Bringing modern computer vision to event cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3857–3866.
[7] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J.C.-W. Lin, G. Fortino, Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection, Inf. Fusion 65 (2021) 13–20.
[8] A. Belhadi, Y. Djenouri, D. Djenouri, T. Michalak, J.C.-W. Lin, Deep learning versus traditional solutions for group trajectory outliers, IEEE Trans. Cybern. (2020).
[9] A. Belhadi, Y. Djenouri, J.C.-W. Lin, A. Cano, Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges, ACM Trans. Manage. Inf. Syst. 11 (3) (2020) 1–29.
[10] M. Zhang, T. Li, Y. Yu, Y. Li, P. Hui, Y. Zheng, Urban anomaly analytics: Description, detection and prediction, IEEE Trans. Big Data (2020).
[11] D. Chen, M.E. Davies, Deep decomposition learning for inverse imaging problems, in: European Conference on Computer Vision, 2020, pp. 510–526.

[12] H. Rezaei, H. Faaljou, G. Mansourfar, Stock price prediction using deep learning and frequency decomposition, Expert Syst. Appl. 169 (2021) 114332.
[13] Y. Djenouri, G. Srivastava, J.C.-W. Lin, Fast and accurate convolution neural network for detecting manufacturing data, IEEE Trans. Ind. Inf. 17 (4) (2020) 2947–2955.
[14] H.Y. Ngan, N.H. Yung, A.G. Yeh, Outlier detection in traffic data based on the Dirichlet process mixture model, IET Intell. Transp. Syst. 9 (7) (2015) 773–781.
[15] J. Gu, Z. Jiang, W.D. Fan, J. Wu, J. Chen, Real-time passenger flow anomaly detection considering typical time series clustered characteristics at metro stations, J. Transp. Eng. A 146 (4) (2020) 04020015.
[16] L. Lin, J. Li, F. Chen, J. Ye, J.-P. Huai, Road traffic speed prediction: A probabilistic model fusing multi-source data, IEEE Trans. Knowl. Data Eng. (2017).
[17] M. Munoz-Organero, R. Ruiz-Blaquez, L. Sánchez-Fernández, Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving, Comput. Environ. Urban Syst. (2017).
[18] Y. Shi, M. Deng, X. Yang, J. Gong, Detecting anomalies in spatio-temporal flow data by constructing dynamic neighbourhoods, Comput. Environ. Urban Syst. 67 (2018) 80–96.
[19] K.-T. Nguyen, T.-H. Hoang, M.-T. Tran, T.-N. Le, N.-M. Bui, T.-L. Do, V.-K. Vo-Ho, Q.-A. Luong, M.-K. Tran, T.-A. Nguyen, et al., Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis, in: Proc. CVPR Workshops, 2019.
[20] S. Bai, Z. He, Y. Lei, W. Wu, C. Zhu, M. Sun, J. Yan, Traffic anomaly detection via perspective map based on spatial-temporal information matrix, in: Proc. CVPR Workshops, 2019.
[21] L. Zhu, R. Krishnan, A. Sivakumar, F. Guo, J.W. Polak, Traffic monitoring and anomaly detection based on simulation of Luxembourg road network, in: IEEE Intelligent Transportation Systems Conference, 2019, pp. 382–387.
[22] G.-L. Huang, K. Deng, Y. Ren, J. Li, Root cause analysis of traffic anomalies using uneven diffusion model, IEEE Access 7 (2019) 16206–16216.
[23] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: ACM Sigmod Record, 29, (2) ACM, 2000, pp. 427–438.
[24] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: ACM Sigmod Record, 29, (2) 2000, pp. 93–104.
[25] Y. Djenouri, A. Belhadi, P. Fournier-Viger, J.C.-W. Lin, Fast and effective cluster-based information retrieval using frequent closed itemsets, Inform. Sci. 453 (2018) 154–167.
[26] A. Belhadi, Y. Djenouri, J.C.-W. Lin, C. Zhang, A. Cano, Exploring pattern mining algorithms for hashtag retrieval problem, IEEE Access 8 (2020) 10569–10583.
[27] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[28] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2015) 142–158.
[29] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[30] X. Sun, P. Wu, S.C. Hoi, Face detection using deep learning: An improved faster RCNN approach, Neurocomputing 299 (2018) 42–50.
[31] A.R. Javed, M. Usman, S.U. Rehman, M.U. Khan, M.S. Haghighi, Anomaly detection in automated vehicles using multistage attention-based convolutional neural network, IEEE Trans. Intell. Transp. Syst. (2020).
[32] A. Al-Dhamari, R. Sudirman, N.H. Mahmood, Transfer deep learning along with binary support vector machine for abnormal behavior detection, IEEE Access 8 (2020) 61085–61095.
[33] A. Belhadi, Y. Djenouri, D. Djenouri, T. Michalak, J.C.-W. Lin, Machine learning for identifying group trajectory outliers, ACM Trans. Manage. Inf. Syst. 12 (2) (2021) 1–25.
[34] J.-G. Lee, J. Han, X. Li, Trajectory outlier detection: A partition-and-detect framework, in: IEEE International Conference on Data Engineering, 2008, pp. 140–149.
[35] J. Wang, Y. Yuan, T. Ni, Y. Ma, M. Liu, G. Xu, W. Shen, Anomalous trajectory detection and classification based on difference and intersection set distance, IEEE Trans. Veh. Technol. 69 (3) (2020) 2487–2500.
[36] J.C.-W. Lin, Y. Shao, Y. Djenouri, U. Yun, ASRNN: a recurrent neural network with an attention model for sequence labeling, Knowl.-Based Syst. 212 (2021) 106548.
[37] A. Gupta, S. Seshasai, 24-Hour knowledge factory: Using internet technology to leverage spatial and temporal separations, ACM Trans. Internet Technol. 7 (3) (2007) 14–es.
[38] Z. Lv, A. Kumar Singh, Big data analysis of internet of things system, ACM Trans. Internet Technol. (2020).
[39] A. Allam, S. Skiadopoulos, P. Kalnis, Improved suffix blocking for record linkage and entity resolution, Data Knowl. Eng. 117 (2018) 98–113.
[40] M. Hadjieleftheriou, N. Koudas, D. Srivastava, Incremental maintenance of length normalized indexes for approximate string matching, in: ACM SIGMOD International Conference on Management of Data, 2009, pp. 429–440.
[41] Y. Djenouri, M. Comuzzi, Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem, Inform. Sci. 420 (2017) 1–15.
[42] M. Silberstein, S. Kim, S. Huh, X. Zhang, Y. Hu, A. Wated, E. Witchel, GPUnet: Networking abstractions for GPU programs, ACM Trans. Comput. Syst. 34 (3) (2016) 1–31.
[43] I. Raicu, I.T. Foster, Y. Zhao, Many-task computing for grids and supercomputers, in: The Workshop on Many-Task Computing on Grids and Supercomputers, 2008, pp. 1–11.
[44] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 97–107.
[45] S. Fan, R.Y. Lau, J.L. Zhao, Demystifying big data analytics for business intelligence through the lens of marketing mix, Big Data Res. 2 (1) (2015) 28–32.
[46] J. Sun, H. Li, Data mining method for listed companies' financial distress prediction, Knowl.-Based Syst. 21 (1) (2008) 1–5.
[47] Y. Djenouri, A. Belhadi, J.C.-W. Lin, D. Djenouri, A. Cano, A survey on urban traffic anomalies detection algorithms, IEEE Access (2019).
[48] Y. Djenouri, D. Djenouri, J.C.-W. Lin, Trajectory outlier detection: New problems and solutions for smart cities, ACM Trans. Knowl. Discov. Data (TKDD) 15 (2) (2021) 1–28.
[49] D. Djenouri, R. Laidi, Y. Djenouri, I. Balasingham, Machine learning for smart building applications: Review and taxonomy, ACM Comput. Surv. 52 (2) (2019) 24.
[50] D. Hemanand, D. Jayalakshmi, U. Ghosh, A. Balasundaram, P. Vijayakumar, P.K. Sharma, Enabling sustainable energy for smart environment using 5G wireless communication and internet of things, IEEE Wirel. Commun. 28 (6) (2021) 56–61.
[51] S. Rawat, A. Srinivasan, V. Ravi, U. Ghosh, Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network, Internet Technol. Lett. 5 (1) (2022) e232.
[52] D.K.K. Reddy, H. Behera, J. Nayak, B. Naik, U. Ghosh, P.K. Sharma, Exact greedy algorithm based split finding approach for intrusion detection in fog-enabled IoT environment, J. Inf. Secur. Appl. 60 (2021) 102866.