***Dust Source Emission Forecasting with GraphSAGE***

Maryam Sanisales, Ali Vefghi, Zahed Rahmati

Department of Mathematics and Computer Science

Amirkabir University of Technology

Tehran, Iran

{m.sanisales, alivefghi, zrahmati} @aut.ac.ir

Ali Darvishi Boloorani

Department of Remote Sensing and GIS, Faculty of Geography

University of Tehran

Tehran, Iran

Ali.darvishi@ut.ac.ir

*Abstract*— This paper proposes a novel method for predicting dust source emission (DSE) using a graph neural network (GNN). Dust storms pose a significant environmental and health hazard, and accurately forecasting their origin is crucial for mitigation strategies. Traditional methods like direct field measurements or remote sensing are limited in coverage or resolution. This work introduces a framework that constructs a novel connected graph from historical hotspot data, incorporating spatial and temporal relationships. A state-of-the-art GNN model, inductive GraphSAGE, is applied to this graph to predict future active dust source locations. The proposed method outperforms a baseline random forest model, demonstrating its ability to capture temporal relationships throughout months. The AUC of GraphSAGE for dust source emission forecasting was approximately 73% which is 17% better than a simple random forest.

Keywords—dust source emission, graph neural network, GraphSAGE

# Introduction

Dust storms represent a significant environmental challenge driven by strong winds that transport fine particles across vast distances and regions [1,2], leading to a wide range of disastrous effects on the environment, including soil degradation, ecological disruption, soil fertility diminishment, and intensifying erosion, which all affect agriculture. Also, they present significant health hazards, particularly in the form of respiratory complications and allergic reactions [3,4]. Therefore, dust source emission (DSE) forecasting identifies where these dust storms occur and can be very important for governments that are prone to these storms to initiate strategies and policies that can mitigate the mentioned harmful environmental effects.

The most common methods for collecting data are direct field measurements which result in high-resolution data. However, this approach is limited in terms of coverage and is costly [5,6]. Another method is Remote sensing which employs sensors on satellites and aircraft. This method has large-scale coverage and less cost than direct field measurements but has limited resolutions [7]. The space for predicting the dust source emission is fairly large and the pattern must be learned over time and space, also it cannot be done with conventional approaches. Therefore, complex and effective computational and machine-learning methods are required. The objective of this paper is to develop a modeling approach that can effectively predict the emission of dust sources in response to changes in environmental variables and weather conditions. This paper’s codes are available on our Git Hub[[1]](#footnote-1).

# Related Works

The nature of the dust source prediction problem is too complex to be understood by simple models. For the model to have efficient applications in real life, the model needs to consider all the factors that are effective in dust source emission. Some of these factors may be too hard to measure or be considered. Previous research investigated models' performance and prediction accuracy in dust source modeling. The findings indicated that there may not be a direct relationship between the model's effectiveness in dust source modeling and its performance or prediction accuracy [8,9].

The researchers examined ensemble approaches for dust source mapping utilizing data mining models, yielding favorable outcomes. In this manner, the classifiers were combined based on the overall prediction accuracy for all classes [10,11]. Data mining, ML, and multi-criteria decision-making methods were investigated in these studies [12,13,14,10,15]. The purpose of using these methods is to map and evaluate the susceptibility of different places to become dust sources.

Given that numerous environmental factors cause the creation and emission of dust, these factors can be employed to model the spatiotemporal behavior of dust sources [16]. Among the environmental stimuli, include vegetation, wind speed [17], soil moisture, soil thickness, temperature [18], precipitation, and topography [19]. The spatio-temporal behavior of dynamic objects can be modeled using deep learning. Some examples of modeling these spatio-temporal objects are consumer location, weather forecasting, and urban traffic [20,21,22,23].



1. Overview of the proposed pipeline

# Proposed Method

In this section, we propose our method for DSE forecasting which is to construct the connected graph from data points and then feed it into the state-of-the-art graph neural network (GNN) model, GraphSAGE. Fig. 1 shows the overview of our proposed pipeline including data preprocessing, space sampler, graph constructor, temporal graph connector, and GraphSAGE. The task of DSE forecasting is a binary classification problem, and we want to find the hotspots that are active in the future based on past hotspots.

## Preprocessing

Assume that are all the hotspots that are active in a specific span of time which all have coordination related to their spatial place and a t for when that are occurred. All of these points are labeled “active”; therefore, we are sure that in a specific coordination for a specific datapoint, that point can be labeled “not active” for different t’s. For example, if is “active” in month 1 we can label for month 2 and other months “not active”. As a result, the search space for learning the pattern becomes too large and the dataset becomes imbalanced. The number of points in this stage is .

Data acquired from different software can have different formats, therefore making sure each datapoint is in its right month is important.

## Space Sampler

After processing the data in the right format and separating them into their corresponding months (approximately 11,000 points in each month), sampling space is needed to reduce the computation cost for machine learning algorithms. It also helps to reduce the redundant and excessive information that are present in each month. The points are reduced to approximately 100 each month by sampling. The sampling here is done randomly but other effective sampling techniques can be used to improve the performance of the model.



1. An example of a delaunay graph created for a specific month. Blue dots are non-active nodes and red nodes are active nodes. The convex hull is not deleted yet here.

## Graph Constructor

This module aims to build homogenous graphs from data points in each month in order to represent the data points in a structured way and incorporate the spatial relations between data points. Formally a homogenous graph is denoted by , where is the set of nodes, and is the set of edges between nodes. Each node in the graph is a dust source and the edge between them is created using their coordination relations.

The edges between nodes are created based on a method called Delaunay triangulation, which generally finds the nearest neighbors to a node. Also, simple KNN method can be used to create edges between nodes but the Delaunay graph is more informative and can capture the spatial relations between nodes in a better way. This claim is proven later in our experiments. Then, convex hull edges are calculated and omitted from each graph as they are redundant outer edges (they do not help to flow more information in graphs) that are not needed in the message passing of GNNs.

Graph Neural Networks (GNNs) can be trained in two ways: inductive learning, where a model is trained on labeled training data and then used to predict labels for unseen test data, and transductive learning, where both training and test data are available during training. The nature of DSE forecasting problem is inductive, therefore we need a model that is capable of inductive learning.

In computational geometry, Delaunay triangulation is a method of creating a triangulation of a set of points. This process entails the subdivision of the convex hull of the aforementioned points into triangular shapes, whose circumcircles do not contain any of the points [31]. Fig. 2 shows an example of these Delaunay graphs.

A convex hull is defined as the smallest convex polygon that encompasses all points within a given set. A convex polygon is a polygon where all its interior angles are less than 180 degrees [30].

## Temporal Graph Connector

This component connects different graphs in each month like a chain by choosing different nodes in each graph and forming a larger graph. The dataset is split here to avoid any possible leakage. 240 months are considered as a train graph, 12 months as a validation graph, and 12 months as a test graph. Assume The graphs are , n nodes of the first graph (for example for step 1) and m nodes of the second graph (for example for step 1) are selected in a random way, then k number edges are linked between these nodes. This process continues until it reaches the last month. The result would be three different graphs that are train graph, validation graph, and test graph.

This process eliminates the dysconnectivity between graphs in each month and connects the graphs by adding temporal edges that are trying to preserve the temporal order of graphs while using static graph neural networks like GraphSAGE.

## GraphSAGE

GraphSAGE is a flexible and powerful deep learning approach for inductive representation on large graphs [33]. It relies on a specific aggregation function to combine information from a node's neighbors to produce new representations for each node in the graph. This variation of graph neural networks is able to generalize better to unseen nodes and can learn the graph in an inductive manner. GraphSAGE has two main parts: Aggregation Function and Node Update Function.

### Aggregation Function

The formula for aggregation function is:

The aggregated representation of node v's neighbors at layer k.

The aggregation function, which can be one of the following:

**Mean Aggregator:** Calculates the average of the neighbor representations.

**LSTM Aggregator:** Uses an LSTM to process the sequence of neighbor representations.

**Pooling Aggregator:** Applies a pooling function (e.g., max, min, sum) to the neighbor representations.

### Node Update Function

The formula for node update function is:

**:** The new representation of node v at layer k.

**and :** Trainable weight matrix and bias vector, respectively.   y

Non-linear activation function (e.g., ReLU).

Concatenates the node's own representation and the aggregated neighbor representation.

# Experiments

## Dataset

The Dataset is available on GitHub[[2]](#footnote-2) which contains the array of 2,616,768 points that have data coordination alongside 11 features (mentioned in the features section) for each of the points and is acquired for 22 years (264 months).

### Study Area

The Middle East is an arid and semi-arid region, and the rainfall in this region is very low. Dust storms represent a significant climate challenge in this region. The Tigris and Euphrates Basin (TEB) is one of the most important sources of dust in the Middle East. The TEB covers more than 700,000 square kilometers. A large number of dust sources are geographically distributed in the TEB, which are affected by natural and anthropogenic changes [24,25].

### Dust Hotspot Sources (DHS)

The DHS locations are from 2000 to 2021 time periods, with a total of 11, 618 DHS identified in TEB using a set of remote sensing image visual interpretation keys and through the analysis of MODIS-Terra/Aqua satellite[[3]](#footnote-3) [26].

### Dust Emission Features

In this study, dust driving factors identified in studies [10, 27, 32] include vegetation, precipitation, soil moisture, wind speed, soil temperature, soil texture (silt, sand, clay), soil thickness, altitude, and slope.

## Experiments and results

We run different experiments with different hyperparameters to get the best result on the mentioned dataset. Hyperparameter optimization is used to search the best hyperparameters for random forest models. Each experiment is executed 50 times with different random seeds to ensure the stability of the result. The random forest model is used as the baseline model. RF1 is a random forest model run on the whole dataset without reducing the dataset and sampling. However, RF2 is a random forest model run on the dataset after sampling. GSAGE is a GraphSAGE model on the reduced and sampled dataset.

There are different hyperparameters through the whole process to finetune with their specific search space, including:

For GraphSAGE:

* Min\_sample\_each\_month: number of “non-active” nodes to sample in each month (5, 10, …, 1000)
* Random\_connector\_num: number of edges to connect between graphs (5, 10, …, 1000)
* Repeat: number of repeats for the whole process (10, 50)
* Sage\_hidden\_channel: number of hidden channels and embedding between layers (8, 16, 32, 64, 128, 256, 512)
* Sage\_learning\_rate: learning rate of the GraphSAGE model (0.1, 0.3, 0.5, 0.01, 0.05, 0.001, 0.003)
* Epochs: number of max epochs (500, 1000, 2000)
* Patience: number of epochs to wait in loss in validation is not decreased (100, 200)
* Criterion: binary cross entropy loss (BCEWithLogitsLoss)
* Optimizer: Adaptative Moment Estimation (Adam)

For RandomForest:

* Max\_features: The number of features to consider when looking for the best split at each node (5, 10, 11)
* max\_depth: The maximum depth of each tree (5, …, 100)
* Min\_leaf\_size: The minimum number of samples required to be at a leaf node (1, …, 25)
* N\_estimators: The number of trees in the forest (100, 200, 300)

As it can be seen from the results in Table. 1 RF1 have the highest accuracy and lowest AUC because the dataset is imbalanced and search space is too large to be understood by simple models. After sampling and balancing the dataset, RF2 has the better results than applying Random Forest on the whole dataset. Both of these models cannot capture the temporal relations between months. Although we executed GraphSAGE on the dataset in a static way, it can be seen from the results that the model’s AUC and accuracy is improved from 56.28% to 72.97% and 56.12% to 76.09%.

1. AUC and Accuracy scores for 3 models

| Model |
| --- |
| AUC | Accuracy |
| **RF1** | 51.04 | 87.06 |
| **RF2** | 56.28 | 56.12 |
| **GSAGE** | **72.97** | 76.09 |

## Evaluation Metrics

Accuracy and F1 score are two metrics that are commonly used to evaluate text classification methods and are calculated according to the following formulas:

* Accuracy =
* Precision =
* Recall =
* F1-score = ,

Where:

TN: Represents the number of records that the model correctly identified as negative and labeled as negative.

TP: Represents the number of records that the model correctly identified as positive and labeled as positive.

FP: Represents the number of records that the model incorrectly identified as negative but were labeled as positive.

FN: Represents the number of records that the model incorrectly identified as positive but were labeled as negative. The area Under the Curve (AUC) metric is another metric that takes into account various aspects of the prediction acts. Receiver Operating Characteristics (ROC) analysis is a widely utilized tool for comparing classifier performance, particularly

in situations where the class distribution and cost parameters are uncertain. In binary classification, ROC analysis assesses classifiers based on the true positive rate (TPR) and false positive rate (FPR)[28]. The AUC represents the area under the ROC curve and is used to assess the overall performance of a binary classification model [29].

# Conclusion and Future works

In this work we constructed a modified homogenous graph from a DHS dataset and applied a static graph neural network to predict which hotspots will be active and which are not in the future. It can be seen from the results that this method is performing better than a simple random forest which is not considering the order of points. Our method can capture the relationships even through time and space using just a simple static graph neural network only by creating the graph in a novel way.

One improvement on model can be facilitating smart sampling method rather than simple random sampling. Sampling can be done in the area around the point to ensure that the features of “non-active” points which are selected are close to the features of the original point, which can help elevate the distinguishability of points that have different labels.

##### References

1. Middleton, N.J. Desert dust hazards: A global review. Aeolian research 24, 2017,53–63.
2. Moghaddam, M.H.R., Sedighi, A., Fasihi, S., Firozjaei, M.K. Effect of environmental 21 policies in combating aeolian desertification over sejzy plain of iran. Aeolian research 35, 2018,19–28.
3. Goudie, A.S. Desert dust and human health disorders. Environment international 63, 2014, 101–113.
4. Okin, G.S., Baker, A.R., Tegen, I., Mahowald, N.M., Dentener, F.J., Duce, R.A., Galloway, J.N., Hunter, K., Kanakidou, M., Kubilay, N., et al. Impacts of atmospheric nutrient deposition on marine productivity: Roles of nitrogen, phosphorus, and iron. Global Biogeochemical Cycles 25, 2011.
5. Baddock, M.C., Bryant, R.G., Acosta, M.D., Gill, T.E. Understanding dust sources through remote sensing: Making a case for cubesats. Journal of Arid Environments 184, 2021,104335.
6. Boloorani, A.D., Papi, R., Soleimani, M., Al-Hemoud, A., Amiri, F., Karami, L., Samany, N.N., Bakhtiari, M., Mirzaei, S. Visual interpretation of satellite imagery for hotspot dust sources identification. Remote Sensing Applications: Society and Environment 29, 2023,100888.
7. Von Holdt, J., Eckardt, F., Wiggs, G. Landsat identifies aeolian dust emission dynamics at the landform scale. Remote Sensing of Environment 198, 2017,229–243.
8. Boroughani, M., Pourhashemi, S., Hashemi, H., Salehi, M., Amirahmadi, A., Asadi, M.A.Z., Berndtsson, R. Application of remote sensing techniques and machine learning algorithms in dust source detection and dust source susceptibility mapping. 2020 ,Ecological Informatics 56, 101059.
9. Rahmati, O., Mohammadi, F., Ghiasi, S.S., Tiefenbacher, J., Moghaddam, D.D., Coulon, F., Nalivan, O.A., Bui, D.T. Identifying sources of dust aerosol using a new framework based on remote sensing and modelling. Science of The Total Environment 737, 2020,139508.
10. Gholami, H., Mohamadifar, A., Collins, A.L. Spatial mapping of the provenance of storm dust: Application of data mining and ensemble modelling. Atmospheric Research 233, 2020a,104716.
11. Gholami, H., Mohamadifar, A., Sorooshian, A., Jansen, J.D. Machine-learning algorithms for predicting land susceptibility to dust emissions: The case of the jazmurian basin, iran. Atmospheric pollution research 11, 2020b, 1303–1315.
12. Boloorani, A.D., Najafi, M.S., Soleimani, M., Papi, R., Torabi, O. Influence of hamoun lakes’ dry conditions on dust emission and radiative forcing over sistan plain, iran. Atmospheric Research 272, 2022, 106152.
13. Papi, R., Argany, M., Moradipour, S., Soleimani, M. Modeling the potential of sand and dust storm sources formation using time series of remote sensing data, fuzzy logic and artificial neural network (a case study of euphrates basin). Engineering Journal of Geospatial Information Technology 8, 2021, 61–82.
14. Naghibi, A., Hashemi, H., Zhao, P., Brogaard, S., Eklund, L., Hassan, H.H., Mansourian, A. Spatiotemporal variability of dust storm source susceptibility during wet and dry periods: The tigris-euphrates river basin. Atmospheric pollution research 15, 2024, 101953.
15. Boloorani, A.D., Kazemi, Y., Sadeghi, A., Shorabeh, S.N., Argany, M. Identification of dust sources using long term satellite and climatic data: A case study of tigris and euphrates basin. Atmospheric Environment 224, 2020, 117299.
16. Papi, R., Kakroodi, A.A., Soleimani, M., Karami, L., Amiri, F., Alavipanah, S.K. Identifying sand and dust storm sources using spatial-temporal analysis of remote sensing data in central iran. Ecological Informatics 70, 2022,101724.
17. Shi, P., Yan, P., Yuan, Y., Nearing, M.A. Wind erosion research in china: past, present and future. Progress in Physical Geography 28, 2004,366–386.
18. Shepherd, G., Terradellas, E., Baklanov, A., Kang, U., Sprigg, W., Nickovic, S., Darvishi Boloorani, A., Al-Dousari, A., Basart, S., Benedetti, A., et al., 2016. Global assessment of sand and dust storms .
19. Goudie, A.S., Middleton, N.J. Desert dust in the global system. Springer Science & Business Media, 2006.
20. Rasp, S., Pritchard, M.S., Gentine, P. Deep learning to represent subgrid processes in climate models. Proceedings of the national academy of sciences 115, 2018, 9684–9689.
21. Yuan, J., Zheng, Y., Xie, X. Discovering regions of different functions in a city using human mobility and pois, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 2012,186–194.
22. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y. Traffic flow prediction with big data: A deep learning approach. Ieee transactions on intelligent transportation systems 16, 2014, 865–873.
23. Zhang, J., Zheng, Y., Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Proceedings of the AAAI conference on artificial intelligence, 2017.
24. Al-Taei, A.I., Alesheikh, A.A., Boloorani, A.D. Evaluating the effects of land use/land cover change on the emergence of hazardous dust sources in the tigris-euphrates basin. Spatial Information Research , 2024,1–14.
25. Shao, Y., Dong, C. A review on east asian dust storm climate, modelling and monitoring. Global and Planetary Change 52, 2006, 1–22.
26. Boloorani, A.D., Papi, R., Soleimani, M., Al-Hemoud, A., Amiri, F., Karami, L., Samany, N.N., Bakhtiari, M., Mirzaei, S. Visual interpretation of satellite imagery for hotspot dust sources identification. Remote Sensing Applications: Society and Environment 29, 2023,100888.
27. Shao, Y., Dong, C. A review on east asian dust storm climate, modelling and monitoring. Global and Planetary Change 52, 1–22, 2006.
28. Bradley, A.P. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern recognition 30, 1997,1145–1159.
29. Vuk, M., Curk, T. Roc curve, lift chart and calibration plot. Metodoloski zvezki 3, 89. ang, Z., Wang, R., Gao, J., Gao, Z., Liang, Y., 2020. Fault recognition using an ensemble classifier based on dempster–shafer theory. Pattern Recognition 99, 107079, 2006.
30. Barber, C. Bradford, David P. Dobkin, and Hannu Huhdanpaa. "The quickhull algorithm for convex hulls." ACM Transactions on Mathematical Software (TOMS) 22.4 (1996): 469-483.
31. Lee, Der-Tsai, and Bruce J. Schachter. "Two algorithms for constructing a Delaunay triangulation." International Journal of Computer & Information Sciences 9.3 (1980): 219-242.
32. Mahowald, N.M., Engelstaedter, S., Luo, C., Sealy, A., Artaxo, P., Benitez-Nelson, C., Bonnet, S., Chen, Y., Chuang, P.Y., Cohen, D.D., et al. Atmospheric iron deposition: global distribution, variability, and human perturbations. Annual review of marine science 1, 2009, 245– 278.
33. Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." *Advances in neural information processing systems* 30 (2017).

1. www.github [↑](#footnote-ref-1)
2. Www. github [↑](#footnote-ref-2)
3. <https://data.mendeley.com/datasets/7937gn7g8c/1> [↑](#footnote-ref-3)