Estimation of Origin-Destination Matrix Using Open-Source Applications Based on Traffic Counts for Mansoura City in Egypt

Ahmed N. Awaad
Sherif M. El-Badawy
Elsayed Abd-Elazem Shwaly
Usama Elrawy Shahdah

Follow this and additional works at: https://mej.researchcommons.org/home

Part of the Civil Engineering Commons, and the Transportation Engineering Commons
Estimation of Origin–destination Matrix Using Open-source Applications Based on Traffic Counts for Mansoura City in Egypt


Abstract

Origin–destination matrices (ODMs) are essential to transportation planning, but estimating them can be difficult and expensive. This study presents a framework for estimating ODMs using traffic flow counts using open-source tools. The framework was applied to a dataset from Mansoura in Egypt. The Flowrouter algorithm, which is part of the Simulation of Urban Mobility microscopic traffic simulation model, was used to route traffic between traffic flow counting locations. The ‘Implausible Routes’ algorithm was used to eliminate unrealistic routes from the generated solution. The results show that the ODM obtained is a reasonable representation of the demand between the different traffic analysis zones in the city, with an $R^2$ of 0.95 between the estimated and observed traffic in the morning peak. The proposed framework is simple to implement, making it a promising approach for estimating ODMs in developing countries and other resource-limited places. Nonetheless, it was applied only to the city of Mansoura. Thus in future work, the framework has to be validated for different cities to ensure its reliability and robustness.

Keywords: Flowrouter, Google distance matrix API, Origin–destination matrix, Simulation of urban mobility, Traffic demand, Traffic flow counts

1. Introduction

With the increasing demand for travel, traffic congestion has become a serious problem in cities. This greatly impacts the environment and people as it causes more fuel consumption, emissions, noise, and delays. According to a 2014 World Bank report, traffic congestion in Cairo costs the Egyptian economy ~$8 billion per year (Nakat et al., 2014). Therefore, travel demand estimation is very important for many transportation applications, such as short-term and long-term transportation planning, transportation policy evaluation, and traffic control and management (Krishnakumari et al., 2020).

The origin–destination matrix (ODM) is a typical method used by traffic engineers and researchers to represent travel demand. It describes the number of vehicles moving between travel zones in the transport network (Bauer et al., 2018). It is important to understand that ODM falls into two categories: static and dynamic ODMs. Static ODM refers to the accumulation of the number of trips in a certain time interval and the generation of a time-independent travel flow. It is mainly used for strategic transport planning. In contrast, dynamic ODM captures traffic fluctuations over a period of time, which can be intraday or daily variations (Yang and Rakha, 2019). Therefore, it is mainly used for short-term applications and diagnosing traffic problems, such as in dynamic traffic distribution models to evaluate the use of ride-sharing for congestion relief (Chiu and Pan, 2021).

Another important point is that ODM can be created in different ways. Traditional methods rely...
on large-scale surveys such as household interview surveys, public transport surveys, taxi surveys, and roadside interviews (Yang and Rakha, 2019), but this method is extremely expensive, time-consuming, and labor-intensive (Bera and Rao, 2011; Dey et al., 2020). Due to the time and financial limitations of the traditional methods, various methods using sensor data, traffic distribution methods, and statistical models have been developed since 1970 to estimate ODMs using different types of data such as traffic counts and other traffic measurements, as vehicle trajectory data calculated using Global Positioning System (GPS), Automatic Number Plate Recognition (ANPR), Wi-Fi, Floating Car Data, Bluetooth or Call Information Recording (CDR) data (Bera and Rao, 2011; Dey et al., 2020; Antoniou et al., 2011).

Due to the widespread use of mobile phones and Internet communities, it is possible to collect a large amount of information about user’s location. Calabrese et al. (2011) used mobile phone location and census data to estimate dynamic ODM on weekends and weekdays in Boston metropolitan area, USA, by identifying travel plans and creating ODM. However this method is effective, but there are some limitations to its use, as the feasibility of calculating a trip depends on the use of the mobile phone for calls and the Internet, so the use of other types of data is essential to obtain highly accurate ODM estimates (Calabrese et al., 2011). Another study, by Mamei et al. (2019) for Piemonte, Emilia Romagna, and Lombardia cites in Italy and by Toole et al. (2015) for Boston, SF Bay, Rio, Lisbon, and Porto cities in the USA, used a variety of big data sources, including CDRs, geospatial data, census data, and transportation surveys. The aggregation of different data sources provided representative ODMs. However, using multiple sources is considered expensive (Ceder, 2019). In addition, Barceló et al. (Barceló et al., 2010) considered data collected from Bluetooth monitoring devices tracking vehicles using a Kalman filter-based technique to estimate dynamic ODM for Barcelona, Spain, but care must be taken as the ODM initialization step is crucial in the estimation process. Moreover, Nigro et al. (2018) showed that using only Floating Car Data is not a suitable method to estimate dynamic ODM, so they used travel time between travel regions and route selection probability to improve the estimation accuracy.

Other studies have focused on ODM estimates based on methods that use traffic counts from different sources. A study was conducted in Tehran using intersection traffic counts collected by loop detectors, and ANPR, GPS, socioeconomic and demographic characteristics, and land use data (Zargari et al., 2021). Zargari and colleagues proposed a convolutional neural network model to estimate ODM with high accuracy in their study. In addition to grouping traffic counts for each vehicle type by combining phone bands with traffic count data, Friedrich et al. (2010) also estimated ODM using telephone routes’ start and end points. Furthermore, Parry and Hazelton (2012) provided a statistical model that uses route information and road traffic flow to construct an ODM to overcome the uncertainty problem between traffic flow and potential route flow.

Karim et al. (2020) estimated the ODM in uncongested transport networks via two stages. First, trips are allocated using a gravity model combined with traffic counts and available socioeconomic parameters to estimate the initial ODM. Then, the Bayesian inference (BI) was applied to estimate the optimal location of traffic flow counts, which improved the accuracy of the original ODM (Karim et al., 2020). Another study by Michau et al. (2019) calculated ODM using Bluetooth data collected from 600 sensors and traffic counts in an inverse optimization problem, but they found that more data are needed to address the effectiveness of their approach. Moreover, Rostami Nasab and Shafahi developed ODM using three different input functions in the model. The first model used only traffic counts, the second used partial lane data collected by license plate recognition, and the third used both traffic flow and lanes. Rostami Nasab and Shafahi’s (Rostami Nasab and Shafahi, 2020) model achieved better results than gradient algorithms.

All of the methods mentioned so far either do not use traffic counts at all or use traffic counts in combination with data from other sources to achieve the ODM. The availability of surveillance cameras and loop detectors on streets and the difficulty of obtaining other traffic measurements, such as GPS data, have increased traffic counts only for ODM estimation.

For example, without using the historic ODM, which is unavailable in most cities, and without any traffic measurements, except for traffic counts as the main stage of ODM. Yang et al. (2018) used a two-level optimization technique for counting traffic to calculate ODM. They used the canonical ridge regression method to create the initial ODM and then performed the calibration phase using traffic flow counting, traffic simulation, and a limited nonlinear programming algorithm called restart simultaneous perturbation stochastic approximation (restart-SPSA), which achieved high performance (Yang et al., 2018). In addition, Xiong et al. (2023) proposed a deep learning method using traffic counts and the topological structure of the city to
estimate the ODM of a large-scale network with an area equal to 400 km², 71,368 edges, and 31,584 intersections in Cologne, Germany.

In addition, Dey et al. (2020) used statistical principles for determining ODM on Internet Networks. They construct an ODM with acceptable accuracy based only on traffic counts using a stochastic optimization problem with transport geographic constraints (Dey et al., 2020). Another study by Englezou et al. (2021) focused on traffic flow sources, where the authors compared ODM results obtained using link counts collected from fixed sensors and unmanned aerial vehicle networks. It was concluded that unmanned aerial vehicles estimate ODM much better than fixed detectors (Englezou et al., 2021).

Overall, there are several methods for estimating ODMs using traffic counts and other traffic measurements. However, the best method to use will depend on the specific application and data availability. In addition, sensitive information such as trajectory data is not yet available in developing countries. Furthermore, the contribution of people with Bluetooth sensors is limited. Hence, the availability of traffic counts over other flow measurements encourages the estimation of ODM depending on traffic counts only.

The methods employed in estimating ODM could be classified into two main categories: (a) the traditional methods and (b) machine learning methods (Bera and Rao, 2011; Malik et al., 2021). The traditional methods are based on mathematical and statistical techniques, such as linear programming, nonlinear programming, and BI, which could be divided into three main approaches: traffic modelling, statistical inference, and gradient-based technique, as described by Abrahamsson (Abrahamsson). The traffic modelling algorithms consider a certain trip distribution model representing the trip-making behavior, such as the gravity model and gravity-opportunity models (Tamin et al., 2003). However, the primary limitation of this technique was its inability to handle external—external trips accurately (Bera and Rao, 2011).

The statistical inference methods such as maximum likelihood, generalized least squares (GLS), and BI were utilized to estimate the ODM (Parry and Hazelton, 2012; Castillo et al., 2008; Cheng et al., 2014; Shao et al., 2015). Nonetheless, these methods often use a previous matrix (also known as a seed matrix) to obtain a unique answer (Yu et al., 2021). The most common traditional method for ODM estimation is the GLS method (Guo et al., 2019). The GLS method minimizes the difference between the observed traffic counts and the estimated OD flows, subject to constraints, such as the conservation of flow. Typically, these techniques deal with bilevel programming. The lower-level model solves a static user equilibrium problem. In contrast, the upper-level model seeks to minimize the weighted distances between the target ;and estimated OD demands and/or between the traffic counts and estimated traffic volumes. Different algorithms and techniques were developed to solve this bilevel problem such as gradient-based methods, evolutionary algorithms, and genetic algorithms (Cipriani et al., 2011; Huang et al., 2013; Kattan and Abdulhai, 2006).

Machine learning methods have become increasingly popular for OD matrix estimation in recent years. Machine learning methods can be used to learn the relationships between traffic counts and OD flows from historical data. Once the relationships have been learned, the machine-learning models can be used to estimate OD matrices from new traffic count data. Neural networks were employed to estimate the ODM from traffic counts by Lorenzo and Matteo (2013), Mussone et al. (2010), and Xiong et al. (2020). Meanwhile, Graph Convolutional and Recurrent Neural Networks were used by Hu et al. (2020) to develop stochastic ODM.

It is worth noting that traditional methods are generally more robust to noise in the traffic count data. However, they can be computationally expensive and may require some expertise. While machine learning methods are generally less robust to noise in the traffic count data. However, they can be more efficient and easier to implement than traditional methods.

Mansoura City in Egypt does not have any historical ODM. The necessity for effective transportation management and congestion mitigation within the city, has made estimating the ODM imperative. Nevertheless, as mentioned, the traditional method requires a massive amount of data currently unavailable in Mansoura. Furthermore, the data from sensor methods such as GPS, CDR, and ANPR are also inaccessible to the city. Given the existing traffic flow counts in Mansoura, this research aims to address the gap by developing a time-dependent ODM for the city. The primary focus is short-term planning, utilizing traffic counts and speeds (obtained through open-source tools like Google Distance API). To facilitate traffic assignment, an open-source traffic simulation tool will be employed. This approach enables a more feasible and data-driven strategy for ODM estimation, contributing to effective urban planning and congestion management in Mansoura.

The proposed research has the potential to make a significant contribution to the field of transportation
planning and management by developing a methodology for estimating time-dependent ODMs using traffic counts and speeds. It would make it straightforward and more affordable for cities to track and manage traffic congestion.

2. Model framework

This section describes the framework of using open-source software and traffic counts as a case study to estimate a dynamic ODM for the city of Mansoura. As shown in Fig. 1, the framework begins by harnessing the open-source services and software for data collection.

The data types employed in the framework were traffic flow counts, transportation networks, traffic analysis zones (TAZs), and traffic speeds. As the focus of the research was on developing countries, traffic loop detectors and other traffic flow measure devices were rarely used. Thus, traffic counts could be collected manually, as in the Mansoura City case. In the meanwhile, the Google service distance matrix API was utilized to estimate the traffic speeds at traffic count locations by applying a request to the service API with two points located in the count location, and it returns the distance between the two points and the travel duration between them which can be used to estimate the speed (Ali et al., 2021; Distance Matrix API overview).

Transportation networks are a vital part of the model, and since it is not available in many cities that need transportation planning and infrastructure assessment, OpenStreetMap (OSM), an open-source database established by a group of mappers who share data about transport infrastructure such as railways, and roads, is an adequate tool to launch the network (OpenStreetMap). Lastly, TAZs were generated depending on the different land uses of the city.

After completing the data collection phase, the data manipulation process was commenced to cleanse the data deficiency using Python and Netedit for network modification and preparation, which is a tool packaged with open-source software for traffic modelling and simulation called SUMO ‘Simulation of Urban Mobility’ (Lopez et al., 2018). The organized data is fed to the Flowrouter module, which is also combined with SUMO.

![Fig. 1. Flowchart of ODM estimation framework. ODM, origin-destination matrix.](image-url)
3. Study area

Mansoura is the capital city of Dakahlia Governorate in Egypt, located in the Nile Delta region, ~120 km north of Cairo, as shown in Fig. 2. It is one of the largest cities in Egypt, with a population of over 1.1 million people (Egyptian central agency for public, 2017). Mansoura is a major educational center, home to Mansoura University, one of the largest and most prestigious universities in Egypt, with more than 176,800, and 41,000 undergraduate and postgraduate students, respectively. In addition, Mansoura University hospitals receive more than 1,814,000 medical visitors annually (University). The city is also home to several other colleges and institutes of higher education besides, it is a thriving commercial and cultural center. Thus, the traffic congestion in Mansoura becomes chronic and affects the people's daily lives and the country's economy, mainly during rush hours (Gabr et al., 2018).

4. Data collection and manipulation

Collecting data was the initial step in the research working frame. This step was divided into four parts as follows:

1. Building the transportation network.
2. Deciding on the size and number of TAZs.
3. Collecting traffic counts.
4. Determining traffic flow speeds.

4.1. Building the transportation network

Currently, there is no previous transport network for Mansoura. Thus, the network had to be built from scratch, which was the first task in this research. Therefore, OSM was the best choice as an initial step in building the network regardless of the OSM's imperfection.

Fortunately, SUMO has the ‘osmwebwizard’ Python tool that can be used to choose the area of interest, Mansoura in our case, and build the traffic network. After importing the Mansoura OSM net into SUMO, considering only the primary, secondary, and tertiary highways, a heavy cleansing step was carried out in such a way:

1. Clean and modify dead links.
2. Delete unused links.
3. Modify the number of lanes in each street.
4. Adjust streets speeds.
5. Amend the intersections.
6. Modify signal timing.

The last step was to create dummy zone links (i.e. dead links) in each zone, and this is because the Flowrouter tool depends on the dead links as sources and sinks (origins and destinations) while estimating the routes. Moreover, these dummy links was named with the zone name to extract the ODM.

For cleaning and modification purposes, Netedit was utilized with the assistance of Google Earth and field measurements. It is important to realize that the SUMO network depends on free nodes and free edges between these nodes, conversely to PTV Vissim, which performs using links and connectors. Henceforth, Fig. 3 illustrates the modified network consisting of 604 free nodes and 1000 free edges due to the existence of large numbers of U-turns.
4.2. Traffic analysis zones

Mansoura City, encompassing both Mansoura and Talkha, is divided into 14 Shiakhas (census blocks). Each Shiakha (i.e. neighborhood) is further divided into smaller subzones with a consistent population per zone. This yields 47 TAZs (i01, i02, ..., i47). Moreover, the surroundings were also divided into 10 TAZs (x01, x02, ..., x10), and all other external areas were divided into four TAZs (xx01, xx02, xx03, xx04). A Python script was developed to convert the coordinates of Google Earth TAZs from WGS84 to UTM and add the SUMO network shift, which is a shift used by SUMO to shift the down-left point to the origin, and then it stored the TAZs into a SUMO file to be displayed as shown in Fig. 4.

4.3. Traffic flow counts

A traffic count study was carried out in December 2018 to collect traffic data in 48 locations for major streets in the city. This study employed the collected traffic counts as a main input parameter for the framework. Fig. 5a depicts the spread of 50 data points used in the traffic collection study, and the traffic count was carried out from 7:00 am to 4:00 pm with 15-min intervals. Another traffic count collection was executed around the Happy-Land area in Mansoura in 2020. By assuming the stability of traffic flow in the short-term periods, this data was utilized as well.

Missing traffic counts were the main issue in preparing traffic flows. Thus two methods were
performed to impute the missing values. The first method was forward and backward imputation, and the other method was creating a light machine learning model to learn from existing data and predict the missing flows. It is important to mention that the second way was better hence, it was generalized on all data. Fig. 5b illustrates the density of traffic in Mansoura using traffic counts. Clearly, the traffic inside Mansoura streets was heavier than the traffic on the outer roads. The last and important step in traffic flow preparation was to map the traffic count locations to detectors in the SUMO network to be utilized as calibration points with Flowrouter.

4.4. Traffic flow speeds

A Python tool was developed to harness the traffic speeds from the Google Maps service distance matrix API, as the Flowrouter algorithm can utilize the speeds to enhance the predictions. The validation of this step was executed based on the work of Ali et al. (2021). The distance matrix provides two types of data. The first one is the average historical travel time between two points in the Google database, which can be used to calculate the average long-term speed in this location using the distance between these two points. Additionally, it provides the current travel time used to calculate the spot speed for on-time traffic. Traffic speeds were collected on a working day at 15-min intervals as the traffic count was carried out.

Fig. 6a depicts the long-term average speed in all count locations, which was skewed to the right. In addition, Fig. 6b shows the distribution of live speeds, which was also skewed with a mean of less than 20 km/h. The lower speed values indicate that Mansoura City requires an effective traffic management scheme.

Fig. 7 illustrates the profile of the traffic speed at some count locations. The green line is the average historical speed at each location and the blue line
represents the average spot speed at the time of execution. From these plots, the traffic speed was higher during the night than in the afternoons.

5. Methodology

The research methodology depends on harnessing the GLS capabilities using the Flowrouter algorithm for the ODM estimation. Flowrouter is a free tool for estimating the travel routes from traffic flow counts by defining the counts as capacities at their locations on the edges and trying by optimization process to find the best routes that maximize these capacities (Behrisch and Erdmann, 2018; Behrisch and Hartwig, 2022).

The algorithm has two objectives to find the best routes ‘d’ as a list. The first one is to minimize the total error between the traffic counts and the current step edge count meanwhile, the predicted flow in each edge is less than the observed traffic count as represented by equation (1).

\[
\text{Min } \sum_{e \in E} c(e) - w_d(e) \quad \text{Eqn. (1)}
\]

Subject to:

\[ w_d(e) < c(e) \ \forall \ e \in E \]

where,

- \( e \) represents the edge.
- \( E \) is the list of all edges.
- \( c \) is for observed count.

\[ w_d = \text{represents the edge load on route } d. \]

The second objective is to maximize the traffic flow in every estimated route as in equation (2).

\[
\text{Max } \sum_{e \in E} w_d(e) \quad \text{Eqn. (2)}
\]

The Flowrouter algorithm starts with the network graph \( G = (V, E) \) where \( E \) is the list of network edges between the vertices \( V \), and the traffic counts \( c \) as well as the list of sources \( S \subseteq V \) and list of targets \( T \subseteq V \). Whereas \( T \) and \( S \) are generated from dead links in the network. In other words, the points that are connected from one direction only.

Then, it takes two steps to find the routes \( d \). First, the algorithm uses Kirchhoff’s law to define weights for every vertex so that the total number of entering flow equals the summation of leaving flows in all following edges except source and target points where the total flow generated from source points equals the total flows absorbs by sink vertexes. Second, the routes are estimated using repeated depth-first searches using the assigned weights (Behrisch and Erdmann, 2018).

After obtaining the routes using Flowrouter, a comparison between observed traffic counts and predicted traffic volumes was conducted to estimate the accuracy of the results. Reaching a reasonable precision, the dynamic ODM was extracted from the routes as the route started from an origin and finished at the destination within a specific time.
Thus, the solution could be described as a bilevel problem.

The model performance was measured using two indicators. The coefficient of determination ($R^2$), as shown in equation (3); and the root mean square error (RMSE), as shown in equation (4).

\[
R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}
\]  
Eqn. (3)

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}
\]  
Eqn. (4)

where,

$R^2$ = the coefficient of determination.

RSS = sum of squares of residuals.

TSS = total sum of squares.

Yi = the predicted value of traffic count from model.

$\hat{y}_i = the\ observed$ value of traffic count.

$\bar{y} = the\ mean$ of the observed data.

N = number of data points.

5.1. Flowrouter algorithm testing and validation

To make sure that the Flowrouter technique is valid for ODM estimation, a synthetic network was built with assumed ODM, as shown in Fig. 8. The network consisted of four TAZs and a simple road network with induction loop detectors at the assumed traffic count locations. Table 1 shows the generated demand between the four TAZs.

The testing phase was carried out in two steps. First, the feedforward process was executed to assign the demand in the network using the default simulation parameters. During this period, the allocated detectors calculated the traffic flow and speed representing the field traffic counts. Following this stage, the estimated counts were passed to the Flowrouter module as input coupled with the network. Finally, the predicted routes with their projected flows were calculated, and the predicted ODM was estimated from the routes.

Fig. 9a illustrates the goodness-of-fit measurement between the assumed and the estimated ODMs; the $R^2$ was over 0.95, and the RMSE was 23.7. It is worth noting that although the estimated ODMs closely matched the assumed ODMs, this was entirely coincident, as the solution to such problems is indeterminate. The focus should be on matching the estimated link counts with the observed ones. Fig. 9b shows the matched between the input (observed) and the output (estimated) traffic counts. The $R^2$ and RMSE are 0.94 and 46, respectively.

5.2. Model building for Mansoura City

The Flowrouter algorithm absorbs the data and produces the optimized routes and the flow in these routes, as shown in Fig. 10. However, sometimes a few unrealistic routes should be eliminated. Therefore, the Implausible Route algorithm, which is another SUMO tool, was utilized to compute the score of implausibility of each route by a weighted sum of several implausibility measurements that include multiple factors such as the route length, air distance, route travel duration, shortest route duration, etc. (Routes).

After executing the implausible routes, the Flowrouter was executed again using the file describing the impossible routes to avoid using it in the assignment stage. A visual inspection phase was conducted following this step to ensure no unrealistic routes could be added manually to the prohibited routes. Finally, a Python script was
developed to obtain ODM from the routes. It is important to mention this framework was implemented in a Python notebook for automation intention and reproducibility.

5.2.1. Model sensitivity

Flowrouter and implausible routes techniques have several hyperparameters that need to be tuned to estimate the best values for these parameters. Implausible routes important parameter was the threshold value between acceptable and unrealistic routes. While the Flowrouter’s first important parameter was the limit, which is the amount of traffic flow the algorithm can assign in the iteration. Additionally, the aggregation time was the second

Table 2. Model hyperparameters.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameter</th>
<th>Tested values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implausible route</td>
<td>Threshold</td>
<td>Not implemented (without), 50, 100, 150, 200, 250</td>
</tr>
<tr>
<td>Flowrouter</td>
<td>Limit (vehicle)</td>
<td>10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000, 1050, 1100, 1150, 1200, 1250, 1300, 1350, 1400, 1450, 1500, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10 000</td>
</tr>
<tr>
<td></td>
<td>Aggregation time (min)</td>
<td>5, 15, 30, 60, 150, 300</td>
</tr>
<tr>
<td></td>
<td>Traffic flow characteristics</td>
<td>Traffic counts only, traffic flow counts + traffic flow speeds</td>
</tr>
</tbody>
</table>
Fig. 11. Limit sensitivity analysis coupled with the implausible routes' threshold values (including check time in Google Colab).
Fig. 12. Aggregation time sensitivity analysis coupled with implausible routes threshold values.

(a) R-Squared sensitivity of the aggregation time with different implausible routes thresholds.

(b) RMSE sensitivity of the aggregation time with different implausible routes thresholds.
parameter in the technique. The last and most important factor was to implement the algorithm using traffic counts only and traffic counts and traffic speeds together to address the effect of traffic speed. Table 2 describes the hyperparameter values managed together in different combinations.

The first stage was executed by engaging traffic flow counts only, flow counts fed the algorithm and the goodness-of-fit measurements were calculated for the outputs. Fig. 11a and b represent the relationship between Flowrouter limit values with R² and RMSE with different implausible route thresholds when the value of aggregation interval was fixed within 15 min.

Increasing the limit value led to an increase in accuracy. However, the accuracy is still steady without change after a value of 700 vehicles per iteration when using the implausible algorithm. Important note: using a small limit value led to an increase in the running time as depicted in Fig. 11c, yet it helps the algorithm distribute the flows more in the network (Detector). Therefore, the limit of 700 was applied as it was the lowest possible value to yield high accuracy.

Focusing on the implausible thresholds, using it enhanced the accuracy of the model with all limits. Nonetheless, their results were identical to the counts-only stage. For this reason, the traffic counts were used only with Flowrouter to estimate the ODM.

### 6. Results and discussions

The Mansoura model was implemented for a weekday from 7:00 am to 12:00 pm (for 5 h) to capture the dynamic in the morning period, and the output of the Flowrouter was compared with the input traffic counts for all 5 h together and every 15-min interval.

The comparison between field traffic flow counts and Flowrouter predicted traffic was barely the same except for some a few points with R² 94.8% and 797 vehicles RMSE out of more than 230 000 vehicles in the all 5-h duration as shown in Fig. 13.

In addition to the whole period check Fig. 14 illustrates the condition of traffic prediction at every single hour. At the morning period at 7:00 am and 8:00 am the accuracy of the algorithm was more than 0.94 R². However, the precision started to decrease in the following hours to be around 0.92 R² due to the traffic congestion. Moreover, every 15 min was compared as in Fig. 15.

Following the checking phase, ODMs were extracted from traffic routes using the first link in the route to estimate the origin and the last link as the destination. Meanwhile, the ODMs for every separate hour were estimated from 7:00 am to 12:00
Fig. 14. Observed counts versus estimated counts every hour.
Fig. 15. Observed counts versus estimated counts every 15 min.
pm. The ODMs were summarized using heatmap visualization, as portrayed in Fig. 16.

Furthermore, Fig. 17 shows the total number of produced trips from each zone. In Fig. 17, the highest number of trips departed from zone i02, which is the nearest zone to the university, followed by zone i26 as it is a high-density residential area.

Similarly, Fig. 18 shows the total number of attracted trips to each zone. All the traffic zones adjacent to the university or governmental services attracted more trips (above average) than other zones.

7. Conclusions and recommendations

This study presents a framework to determine the ODM of trips between different TAZs in Mansoura.
City, Egypt, based on traffic count data collected for 5 h between 2018 and 2020, for 5 h. The SUMO’s ‘osmwebwizard’ Python tool was used to extract the urban transportation network from the open-source OSM database.

The Google Service Distance Matrix API was also used to estimate traffic speeds at the traffic counting locations. Then, the Python SUMO tool ‘Flowrouter’ was used to estimate the short-term ODM for Mansoura.

The proposed framework for estimating ODMs using traffic flow counts and open-source tools was effective. The model achieved an $R^2$ of 0.948 and an RMSE of 797 vehicles out of more than 230 000 vehicles over 5-h period. The model could also accurately predict traffic conditions at every hour during the morning peak hours, with an $R^2$ of $\sim 0.94$.

The results of this study suggest that the proposed framework constitutes a promising approach for estimating ODMs in developing countries and other
resource-limited locations. The framework is relatively inexpensive and simple to implement.

It should be noted that the ‘Flowrouter’ algorithm was trained only on the number of traffic flows, so future work will focus on developing a new algorithm to incorporate traffic speeds into the framework. Furthermore, the framework used in this analysis was only tested for the city of Mansoura, so more data from different cities would highlight the relevance of the model for other cities with different traffic patterns and infrastructure. In addition, this study demonstrated the potential of the proposed ODM estimation framework, which has the potential to improve transportation planning and decision-making. Furthermore, the employed traffic counts in this study were obtained from two traffic surveys conducted in 2018 and 2020. Therefore, it is essential to incorporate more recent data to accurately estimate an updated ODM.

Author contributions

The authors confirm contribution to the paper as follows: study conception and design: Awaad, and

(a) The number of trips attracted to each zone.

(b) A choropleth map to represent the most and least attractive zones.
Shahdah; data collection: Awaad, El-Badawy, and Shahdah; analysis and interpretation of results: Awaad, El-Badawy, and Shahdah; draft manuscript preparation: Awaad, El-Badawy, Shwaly, and Shahdah. All authors reviewed the results and approved the final version of the manuscript.

Conflicts of interest

The authors declare no conflict of interest.

References


A.N. Awaad et al. / Mansoura Engineering Journal 49 (2024) 1–20


