Deep Learning for Integrated Origin–Destination Estimation and Traffic Sensor Location Problems

Mahmoud Owais

***Abstract*— Traffic control and management applications require the full realization of traffic flow data. Frequently, such data are acquired by traffic sensors with two issues: it is not practicable or even possible to place traffic sensors on every link in a network; sensors do not provide direct information about origin–destination (O–D) demand flows. Therefore, it is imperative to locate the best places to deploy traffic sensors and then augment the knowledge obtained from this link flow sample to predict the entire traffic flow of the network. This article provides a resilient deep learning (DL) architecture combined with a global sensitivity analysis tool to solve O–D estimation and sensor location problems simultaneously. The proposed DL architecture is based on the stacked sparse autoencoder (SAE) model for accurately estimating the entire O–D flows of the network using link flows, thus reversing the conventional traffic assignment problem. The SAE model extracts traffic flow characteristics and derives a meaningful relationship between traffic flow data and network topology. To train the proposed DL architecture, synthetic link flow data were created randomly from the historical demand data of the network. Finally, a global sensitivity analysis was implemented to prioritize the importance of each link in the O–D estimation step to solve the sensor location problem. Two networks of different sizes were used to validate the performance of the model. The efficiency of the proposed method for solving the combination of traffic flow estimation and sensor location problems was confirmed from a low root-mean-square error with a reduction in the number of link flows required.**

***Index Terms*— Deep learning, global sensitivity analysis, sensor location problem, traffic data prediction.**

NOMENCLATURE

Symbols

G	Graph.
N	Is the set of nodes/vertices.
A	Is the set of arcs/links.
a	Is a subscript for arcs.
$s,\ e$	Are subscripts of arc (a) start node and end node, respectively.
ε_a	Is arc/link (a) cost.
W	Is the demand node pair set.
w	Is a subscript of the demand nodes pair.
$o,\ d$	Are subscripts of the O-D of the pair (w), respectively.
$\hat{w}$	Is the number of o, d pairs in the set W.
$\mathcal{D}$	Is the total demand flow in the network.
T	Is the vector of nodal flows.
t_w	Is the flow of (w).
m	Is number of network links.
$\bar{l}$	Is the number of links installed with sensors.
$S(.,.)$	Is the search space domain function of TSLP solutions.
$t_w^*, \bar{t}_w$	Are the actual and estimated demand flow, respectively.
K_w	Is the path set of (w).
k	Is a subscript of an elementary path in K_w.
H	Is the path flows vector.
h_w^k	Is the flow of path (k) of the node pair (w).
X	Is the link flows vector.
x_a	Is the link (a) flow.
δ_{kw}^a	Is incident factor equals 1 if the link a is a part of the path k connects w, 0 otherwise.
Δ	Is the link-path incident matrix.
g_w^k	Is the path kw generalized cost.
n_k^w	Is the none-additive path kw cost.
π_w^k	Is the path choice proportion of pair w.
$G,\ C$	Are the vectors of all the paths and links' generalized costs, respectively.
N	Is the vector of all paths' nonadditive costs.
H^*, X^*	Are the optimal solutions for H and X.
$£_x$	Is the search space of all link flows.
$£_h$	Is the search space of all path flows.
θ	Is a parameter that denotes the SUE model stochasticity.
$\alpha,\ \chi$	Are calibration factors.
Q_a	Is the capacity of the link (a).
$\mathcal{B}$	Is the vector of all π values.
λ_w	Is the error term in the demand pair (w) estimation.
z_a	Is a decision variable equals 1 if the sensor is needed on the link (a), 0 otherwise.
c_a	Is the sensor installation cost.
μ_1, μ_2	Are weight factors.
F	Is the vector of positive Monte Carlo simulated values.
f_w	Is the relative value of w to the total demand $\mathcal{D}$.
γ	Is a vector of random variables γ_w with null mean and σ_w standard deviation.
Ω_{weight}	Is the weight decay function.
$\mathrm{C}_{\mathrm{sparse}}$	Is the sparse auto-encoder cost function.
$U,\ B$	Are decoding and encoding matrices.

Manuscript received 2 March 2023; revised 17 August 2023 and 8 November 2023; accepted 16 December 2023. Date of publication 2 January 2024; date of current version 2 July 2024. The Associate Editor for this article was S.-H. Kong.

The author is with the Civil Engineering Department, Assiut University, Assiut 71515, Egypt, on leave from the Civil Engineering Department, Faculty of Engineering, Sphinx University, New Assiut 71684, Egypt (e-mail: maowais@aun.edu.eg).

Digital Object Identifier 10.1109/TITS.2023.3344533

ρ	Is a small value that acts as a sparsity parameter.
$\hat{\rho}_j$	Is the average of hidden unit j.
$J_j(x_i)$	Is the activation function.
λ,β	Are weight factors.
GSI_a	Is the variance-based sensitivity analysis index for link (a).

Abbreviations

AAT	All variables at a time.
ANNs	Artificial neural networks.
AVI	Automatic vehicle identification sensors.
BIP	Binary integer linear programming.
BP	Backpropagation.
DL	Deep learning.
DUE	Deterministic user equilibrium.
EODM	Estimated O-D matrix.
FCL	Fully connected layer.
GA	Genetic algorithm.
GLW	Greedy layer-wise.
GSA	Global sensitivity analysis.
KL	Kullback-Leibler.
LFI	Link flow inference.
Lidar	Light detection and ranging sensors.
LSA	Local sensitivity analysis.
MAE	Mean absolute error.
MAPE	Mean absolute percentage error.
ML	Machine learning.
MPRE	Maximum possible relative error.
MSA	Method of successive averages.
OAT	One variable at a time.
OBS	Flow observability.
O-D	Origin-destination matrix.
ODEP	Origin-destination estimation problem.
PRC	Path reconstruction.
RMSE	Root mean square error.
RODM	Reference O-D matrix.
SA	Sensitivity analysis.
SAEs	Stacked auto-encoders.
SLP	Screen line/traffic surveillance.
SUE	Stochastic user equilibrium.
TME	Travel time estimation.
TODM	True O-D matrix.
TSLP	Traffic sensor location problem.
VBSA	Variance-based sensitivity analysis.

I. Introduction

The origin–destination (O–D) demand matrix is a decisive data type in transportation planning and traffic operations, and is essential for intelligent transportation systems. Traffic analysts and operators strive for reliable O–D data to help make the right strategic, tactical, and operational decisions. The O–D matrix can be easily transformed into other flow data types (i.e., link and path flows) in static and dynamic ways. This transformation is achieved by solving a well-known traffic assignment problem [1], [2].

Traffic assignment models with different efficient solution algorithms have been reported previously [3]. However, the problem of how to estimate an accurate O–D matrix remains. The most accurate type of traffic data is the traffic count on transportation network streets/links. Unfortunately, installing sensors on all network links is not affordable because of the sensor installation cost with respect to the required number of links in a moderate-sized real network. Even if all the network links are covered with sensors, none of the traffic assignment models have an inverse formula for obtaining the estimated O–D matrix that causes the observed flows [2].

O–D estimation methods are sourced from a priori information. The prior O–D matrix that guides the search process is the most widely used type. Consequently, the accuracy of the estimation method depends on the reliability of the O–D data used, which usually originate from historical records. The accuracy of the estimated O–D matrix is calculated by its ability to produce the flows used in its estimation while achieving the smallest deviation from the historical O–D matrix [4].

However, the distribution of network traffic sensors is not trivial. This problem suffers from combinatorial complexity, which is challenging for the application of these sensors to real-sized transportation networks. From a multidisciplinary perspective, the problem is easy to formulate, but difficult to solve. Its search space with 2^m possibilities (where m is the number of network links) can be considered as one dimension of its complexity. If the number of available sensors is $\bar{l}$, the search space domain function is expressed as follows:

$$S(m,\bar{l}) = \frac{m!}{\bar{l}!(m-\bar{l})!}. \tag{1}$$

Adding other dimensions to the problem, such as installation objectives, budget constraints, sensor type, optimality criteria, and sensor error/failure probabilities, further complicates the problem [5].

The research question is two-dimensional. First, how can we obtain an accurate O–D matrix from traffic counts? Second, what is the optimal number/location of sensors to perform the first task efficiently? In the literature, the two problems are uniquely identified respectively as the O–D estimation problem (ODEP) and traffic sensor location problem (TSLP). Most studies have addressed either the ODEP or TSLP separately, whereas only some studies have addressed the ODE-TSLP [6]. The combined problem is typically handled as a two-stage or bilevel problem. The ODEP and TSLP were solved simultaneously or sequentially to minimize the error between the actual and predicted traffic flows while optimizing the location and number of deployed sensors. These two problems have several dimensions of complexity [7], [8], making their integration into a single solution method an interesting subject. In this study, the integrated problem is abbreviated as the ODE-TSLP, and the relationship among the different elements is depicted in Fig. 1.

Although solving the ODE-TSLP is not a new topic, in this study, we showed that the selected approach is unique and benefits from contemporary advances in machine learning (ML) techniques to achieve better O–D matrix estimation accuracy. Global sensitivity analysis (GSA) methods were introduced to solve the TSLP. This study used synthetic demand data to train the proposed ML model to solve the ODEP using only the free

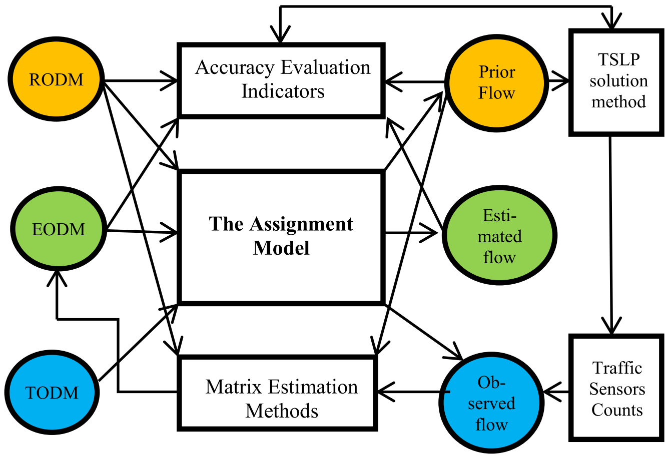


Fig. 1. The relation among ODEP and TSLP input and output data.

error counts on the network links. The model could extract latent flow features to map link flows to O–D flows to reverse the assignment problem. GSA was added to investigate the ML mechanism to locate the essential links to be observed as a solution to the TSLP. The framework presented in this article is new to the ODEP-TSLP literature, as it combines superior ML capabilities for the prediction problem with GSA logic for the selection problem.

The remainder of this article is organized as follows. Section II reviews the studies on the ODE-TSLP to highlight their motivations and potential contributions. Section III provides the formulation of this problem. Section IV presents a detailed description of the proposed method. Section V verifies the proposed approach using numerical case studies. Finally, Section VI concludes the article.

II. State-of-the-Art Methods

In this section, we review studies that have investigated the ODEP and TSLP. Furthermore, we present ML applications for traffic flow prediction and the necessary background on the GSA approach to demonstrate that adapting the selected ML and GSA methods in the proposed context is a new contribution.

A. Origin–Destination Estimation Problem

The ODEP is a pioneering application for recorded traffic flow data analysis [9]. The quantification of O–D data has been addressed over the last three decades. Studies have attempted to find the most accurate O–D matrix using statistical methods, such as entropy maximization [10], least squares [11], maximum likelihood [12], and Bayesian networks [13].

Reference [14] introduced the concept of essential links to enhance the quality of the estimated O–D matrix. The traffic volume is the best criterion for selecting these links. Reference [15] devised the maximum possible relative error (MPRE) indicator, which measures the worst possible deviation of the predicted O–D matrix from the actual O–D matrix. The superiority of this indicator is that its estimation is based on the fact that the true O–D matrix is unknown. Certain conditions for the location of the sensors are required in advance to restrict the MPRE value. Reference [16] proposed four heuristic sensor location rules for minimizing the expected MPRE. These rules are called coverage rules, and have become well known. Reference [17] developed a two-stage iterative algorithm for experiments with several link selection criteria to fulfill coverage rules. Reference [18] used different binary integer linear programming (BIP) formulations to achieve full O–D coverage and preserve the boundness of the MPRE. Reference [19] introduced the maximum total demand deviation, which is an error-bounding scale close to the MPRE.

Reference [20] formulated a linear system of equations to connect unobserved O–D pair flows with observed flows (e.g., link, node, and path flows). Reference [21] proposed a polynomial solution algorithm for a linear system of equations. Alternatively, the authors of [22] developed a comprehensive matrix tool that included all flow elements, representing O–D flows as binary variables (0 or 1). Reference [23] solved this problem by dispatching sensors to intercept all understudied O–D pairs using four BIP formulations. Subsequently, a genetic algorithm (GA) was presented to solve the formulations. A path estimator method was used to predict the reliability of the estimated O–D matrix without directly incorporating it into the fitness function of the GA.

Reference [24] adapted these BIP formulations, and extended them to time-expanding networks. The authors used the generalized least-squares method for the ODEP and Kalman filtering methods to highlight the critical links for the TSLP. Reference [25] discussed Bayesian networks for selecting links based on the Bayesian network convergence by considering the error of the estimated matrix. Reference [26] developed a hybrid greedy randomized adaptive search procedure to extend the work in [24] to a multiobjective problem. The model chooses links based on the O–D pair coverage and the information gained from installing the sensors on these links.

Reference [1] integrated the MPRE directly into a random priority metaheuristic to obviate the need for high-quality a priori information to obtain a more reliable O–D matrix. Simultaneously, the authors solved the TSLP as a min–max optimization model to reduce the MPRE. Interestingly, the proposed method was not sensitive to the O–D estimation method. Reference [8] extended [1] by addressing the time-dependent sensor failure effect by considering the sensor lifetime. Reference [2] derived a mathematical model using a factorization scheme. The authors used backward substitution to provide an exact solution to the ODEP. However, network scalability remains a challenge for such solutions. Reference [7] examined both heuristics and exact solution methods for the TSLP considering the full O–D pair coverage problem.

B. Traffic Sensor Location Problem

In the literature, the strategy of the TSLP solution is pivotal to the accuracy of the estimated O–D matrix. For example, the mean absolute error (MAE) and root-mean-square error (RMSE) are widely used statistical indices. Unfortunately, the absence of actual O–D data prevents true error estimation.

Therefore, this problem must be addressed [27]. Another way to predict accuracy is by measuring the predicted O–D to reproduce the link flows (i.e., the sensor input data) under the presumed traffic equilibrium. Bilevel optimization can address this dilemma by considering the ODE-TSLP at the upper level, whereas the traffic assignment model is considered at a lower level [28], [29]. In most reviewed studies, the TSLP solution strategies for the ODEP were restricted to simple rules, generally related to coverage factors, and subsequently, O–D estimation methods were adopted for tackling real-scale networks.

With progress in studying the ODE-TSLP, it is apparent that the information quality of the sensors (i.e., quantity, accuracy, and spatial coverage) is crucial in determining the estimated O–D matrix accuracy [27]. Generally, the TSLP, as a separable problem, targets the traffic flow information quality by optimizing the location and number of sensors. Reference [6] categorized TSLPs into six distinct classes: O–D estimation (ODE), flow observability, link flow inference, path reconstruction, screen line/traffic surveillance, and travel time estimation problems. The author observed that the ODEP had the highest share percentage among all the reviewed TSLP studies. However, there has been a recent decline in ODE research compared with other classes [6].

Some common assumptions for the TSLP should first be defined regarding the type of sensors studied, the sensor location in the network, sensor information uncertainty, and budget constraints. Numerous sensing technologies exist in practice. Examples of sensors include inductive loops, video cameras (i.e., image recognition), automatic vehicle identification sensors, quartz weigh-in-motion sensors, light detection and ranging sensors, and ultra-acoustic sensors. These types are categorized into two groups: passive sensors that only count vehicles, and active sensors for other types of information, such as vehicle type, vehicle observation time, and path information [30]. Studies usually state the sensing mode as passive, active, or a combination of these two (i.e., heterogeneous) [31]. The sensor type stipulates the sensor location in a network, where network intersections require sensors to track vehicle-turning ratios. Network links can accommodate all sensor types. The type and location of the sensors are essential assumptions for providing a solid basis for the ODE method and the mathematical formulation of the TSLP [17]. While most studies have considered sensors to be error-free tools, only a few have presented the uncertainty in sensor data as either an error in measurements [32] or a total sensor failure [8], [33], [34]. Budget constraints explicitly address the limited-resource issue in which there are few sensors to distribute in the network. Budget constraints are reflected in the number of sensors installed in the network. Furthermore, monetary restrictions can be included directly by incorporating the installation cost of each sensor when heterogeneous sensor types are used [31], [35].

C. Machine Learning for Traffic Flow Estimation Problem

ML is an intelligent tool for predicting or estimating unknown variables based on learning paradigms. Artificial neural networks (ANNs) are among the most well-known ML techniques. They have been widely utilized, and their structures have been developed over time to produce more intelligent learning techniques. Deep learning (DL) architectures are currently considered the most prevalent models in transportation science applications.

If the above ODE-TSLP is categorized as a traffic flow prediction problem, to the best of the author's knowledge, there is no ML application for the problem. ML models are confined to predicting the flow in network links as a time-series problem [36], [37], [38], [39]. The complexity of traffic prediction is based on dynamic temporal dependencies. Some studies have also addressed the spatial correlation of the network among intersection/street flows to correlate observed points with unobserved ones [40], [41]. In general, the accuracy of ML models is significantly higher than that of classical statistical techniques [42], [43], [44]. However, the presented ML models lack interpretability because, in most cases, they are treated as "black-box" tools [38]. Sensitivity analysis (SA) is an ad hoc tool used to overcome this major drawback [44].

To this end, ML has not been applied to solve the stated ODEP-TSLP so far because of two major problems. First, ML models require large input data to learn complex nonlinear relationships. The only available input datum for our problem was a single historical matrix. Second, the lack of interpretability in identifying inputs minimizes the loss function. Consequently, even if we trained a model for the ODEP, we could not identify the essential links required to produce an accurate O–D matrix.

D. Global Sensitivity Analysis

In this study, DL was selected as the O–D computational tool. There is no analytical method for inferring the links influencing estimation accuracy in the proposed DL structure. Answering this question will help solve the accompanying TSLP. SA was designed to determine how the variations in independent variables would affect the studied outcome value. Within this broad definition, the analyst's choices of inspection level, purpose, and modeling domain influence the complexity and method of SA. SA can be differentiated into several types based on whether and how these questions are approached. The analyst first chooses the modeling technique, whose accuracy in modeling the real world is the main determinant of the accuracy of the SA results. The analyst should then specify how to vary the explanatory factors in their domain space. Finally, the analyst chooses the appropriate SA method to address the imposed questions.

There are two approaches for linking flow-domain manipulation: local sensitivity analysis (LSA) and GSA. The LSA searches for the output variation amount in the form of a designed index when the explanatory variable is moved around a reference value while the other variables are fixed to their optimal/nominal values. Therefore, the LSA is stipulated by studying one variable at a time (OAT). In contrast, the GSA can address (like LSA) the same outcome variation index while examining the variation in the entire domain of the explanatory factor under study. The GSA can be performed when all other variables are permitted to change globally in their domain space (AAT) or are simply reduced to OAT. GSA-AAT has advantages over GSA/LSA-OAT because it considers the interactions among explanatory variables when calculating indices. The analyst finally chooses the SA method, which mainly defines the measured SA index.

The SA literature includes several methods ranging from analytical models, such as partial derivatives, to sample-based models, such as variance-based indices [45]. Analytical models are confined to the first modeling approach (i.e., parametric models), whereas sample-based models can handle both parametric and nonparametric models at the cost of performing several runs [46]. Several researchers have widely accepted GSA techniques for various topics and applications [47], [48], [49], [50]. In this study, GSA-AAT was selected for the variance-based sensitivity analysis (VBSA) [51].

The salient contribution of this study is the novelty of the proposed method, which combines GSA and DL for the first time (to the best of the author's knowledge) to solve the ODE-TSLP. The robustness of the method is ensured by the capability of the GSA tools to examine network link contributions simultaneously in addition to their inherent interactions in estimating the O–D matrix. Simultaneously, the sophistication of the DL architecture enables the accurate prediction of the O–D matrix by learning the latent relationships among the traffic flow elements.

III. Problem Formulation

Traffic flow information can be categorized into three main types: O–D, link, and path flows. An O–D flow is formed by a two-dimensional matrix, where the value of each entry represents the trips between the origin zone (defined by the row number) and the destination zone (column number). The matrix can be abbreviated as an ordered vector for all the entries. Both the link and path flows were obtained from the O–D flow information using an appropriate assignment technique. Reversing this task using link flows is not as easy as in the original assignment step [11], [52]. An unlimited number of O–D matrices can be obtained from a complete link flow vector [53], [54]. In this study, we aimed to estimate the O–D matrix from a limited number of passive traffic counters, where the optimization of the required number and location of sensors will be investigated as well.

A. Network Representation

Consider a directed transport network, $G = (N, A)$, where N is the set of nodes/vertices that are connected by the set of arcs/links $A = \{a \equiv s, e \in N \mid (s, e) \neq (e, s), |A| = m\}$. Each arc ($a$) has a start node ($s$) and an end node ($e$) associated with the travel time/cost ε_a. By definition, an O–D (nodal) flow is generated between prespecified nodes in the network. The demand node pair set $W = \{(o, d) \equiv w \mid o, d \in N,\ W \subseteq N^2, |W| = \hat{w}\}$, whereas the total demand flow in the network ($\mathcal{D}$) is decomposed into a vector of nodal flows $T = \{t_w \mid \sum_w t_w = \mathcal{D}\}$. The flow streams in the network pass through a set of paths that connect each demand node pair (w). Although the path size in a network increases with the network size, the path set K_w is finite for each w with an elementary path k. The path flow vector $H = \{h_w^k \mid k \in K_w,\quad \sum_k h_w^k = t_w\}$. Each path is connected with consecutive links, where the link flow vector is denoted by $X = \{x_a, a \in A \mid x_a = \sum_w \sum_k \delta_{kw}^a\ h_w^k\}$, where δ_{kw}^a is an incident factor that equals 1 if the link a is a part of the path k connecting w node pairs, and equals 0 otherwise. The link-path incident matrix (Δ) is an arrangement of all δ in the network size $|A| \times |H|$. We can now define the generalized path cost as follows:

$$g_w^k = \sum\nolimits_{a \in A} \delta_{kw}^a\ \varepsilon_a(x_a) + n_k^w\ \forall\ k \in K,\ w \in W, \quad (2)$$

where g_w^k is the generalized cost of the path kw, which is the sum of its constituent link costs, in addition to the nonadditive path cost (n_k^w), which can only be calculated after the final configuration of the path is determined, and is given by the waiting time at different intersections.

B. Network Loading

Network loading models are a major topic in transportation science. The required task is to map the flows among the transportation network elements (i.e., nodal, path, and link flows). These models are considered the predominant tool for strategic and tactical planning. However, there is no closed function to solve this problem because of the complexity of modeling user behavior. Transport network flow propagation is expressed as follows:

$$h_w^k = \pi_w^k(g_w^k) t_w,\ \ \forall\ k \in K,\ w \in W, \quad (3)$$

$$x_a = \sum\nolimits_w \sum\nolimits_k \delta_{wk}^a\ \ \pi_w^k \big(\sum\nolimits_{a \in A} \delta_{kw}^a \varepsilon_a(x_a) + n_k^w\big) t_w, \quad (4)$$

where π_w^k is the path-choice proportion for w. Equation (3) expresses the path flow as a function of the path-generalized cost g_w^k. Equation (4) expresses the fixed-point nature of link flows. Flow propagation has two definitions (path- or link-based). Consider two sets, $\pounds_h$ and $\pounds_x$, of feasible solutions to (3) and (4). In the transportation axioms, the Wardrop principle of equilibrium is a well-accepted procedure for obtaining an optimal solution. In general, the equilibrium can be stated as a variational inequality in terms of path and link flows, as follows:

$$G^t\ (H - H^*) \geq 0,\ \forall\ H\ \in \pounds_h, \quad (5)$$

$$C^t\ (X - X*) + N^t(H - H*) \geq 0,\ \forall X\ \in \pounds_x\ \&\ H\ \in \pounds_h,$$
$$s.t.\ X^* \in \pounds_x \& H^*\ \in \pounds_h \quad (6)$$

where G *and* C are the vectors of all the paths and the generalized costs of the links, respectively. N is the vector of the nonadditive costs of all the paths. t is the superscript for the vector transpose. H^* and X^* are the optimal solutions based on the selected model assumptions. Numerous models for simulating user movements/choices in a network have been discussed in the literature. Table I presents the basic dimensions of assignment problems that can be priorities for analysts' decisions.

To alleviate the complexity of the assignment problem, the travel time of the links was assumed to be independent of the flow for links that only loaded the flows to the shortest paths (i.e., all-or-nothing assignment). If the dependence is

TABLE I
CHOICE DIMENSIONS FOR TRANSIT ASSIGNMENT MODELS

Assumption	Model choice dimensions		This study
Congestion effect	All or nothing	Multipath	Multipath
Equilibrium type	Deterministic	Stochastic	Stochastic
Capacity constraint	Mild	Strict	Mild
Solution algorithm	Link-based	Path-based	Link-based

addressed, the multipath assignment is observed to achieve Wardrop's equilibrium principle. In deterministic user equilibrium (DUE) models, passengers are assumed to have precise knowledge of network travel times and conditions. Alternatively, stochastic user equilibrium (SUE) models remove this perfect knowledge assumption, and consider perceived user information to be random variables. Users typically encounter congested streets that change their path selection owing to either an increase in link impedance (mild capacity) or an incapability to pass through this link owing to capacity insufficiency (strict capacity). With strict capacity constraints, user-choice modeling becomes a more complicated task because congestion levels strictly determine individual preferences. The literature contains numerous iterative solution algorithms for different traffic-assignment formulations. Each algorithm has its own method to reach the equilibrium solution in the least number of iterations with the best convergence values. The solution is interpreted as either a link or path flow. In most cases, link-based algorithms are preferred because of the absence of path enumeration problems and the uniqueness of the solution in terms of link flows.

The configuration of the traffic assignment model is based on what an analyst believes to resemble truly the route choices of a user. SUE was selected for analysis in this study because of its generality and capability to simulate different user behaviors. Interestingly, the DUE results can be incorporated by adjusting the SUE parameters. However, the traffic assignment model in the proposed framework is flexible, and can be replaced by any type of model.

The nonadditive cost term was removed from the analysis to reduce the variational inequality in (6) into an optimization model. The SUE based on the multinomial logit assignment solves the following optimization problem:

$$argmin_{\ x_a} = \frac{1}{\theta}\sum\nolimits_{w\in W} t_w \log\left(\sum\nolimits_{k\in K_w} \exp\left(-\theta h_w^k\right)\right) + \sum\nolimits_{a\in A} x_a\varepsilon_a(x_a) - \sum\nolimits_{a\in A}\int_0^{x_a}\varepsilon_a(x)dx, \tag{7}$$

$$s.t.\ \ \varepsilon_a(x_a) = \varepsilon_a^0\left(1+\alpha_a\left(\frac{x_a}{Q_a}\right)\chi_a\right), \tag{8}$$

$$h_w^k = \sum\nolimits_{a\in A}\delta_{kw}^a x_a,\ \forall k\ \&w, \tag{9}$$

where θ denotes the degree of model stochasticity. Equation (8) expresses the well-known Bureau of Public Roads formula [55], where $\varepsilon_a(x_a)$ is the link-volume cost function, and ε_a^0 is the cost associated with the initial condition of the link (a). α *and* χ are calibration factors, and Q_a is the capacity of the link.

C. ODE-TSLP

We consider an actual traffic assignment procedure to understand the problem better. Therefore, link/O–D mapping can be expressed as follows:

$$X^* = \Delta\ \ B^t\ T^*, \tag{10}$$

where B is the vector of all π values that resulted from the proposed assignment model. In practice, distributed sensors can monitor a subset of the link flows in X^*. Two questions arise sequentially or simultaneously: What are the optimal numbers/locations of the links? How can (10) be reversed? This problem can be reduced to the following optimization problem:

$$\underset{t_w,\ z_a}{argmin}\quad \mu_1\frac{\sum_{w\in W}\lambda_w^2}{\hat{w}} + \mu_2\left(\sum\nolimits_{a\in A} c_a z_a\right), \tag{11}$$

$$\text{s.t.}\quad \lambda_w = \frac{t_w^* - \bar{t}_w}{t_w^*},\ \forall w\ \in W, \tag{12}$$

$$\sum\nolimits_{w\in W}\sum\nolimits_{k\in K}\delta_{wk}^a\pi_w^k\bar{t}_w\lambda_w z_a = 0,\ \forall a\ \in A\ , \tag{13}$$

$$z_a \in\ (0,\ 1), \tag{14}$$

where λ_w is the error term in the demand pair (w) estimation procedure. t_w^* and $\bar{t}_w$ denote the actual and estimated demand flows, respectively. z_a is a decision variable equal to one if the sensor is required on link (a) and zero otherwise, and c_a is the sensor installation cost. μ_1 and μ_2 are the weight factors that reflect the multiobjective nature of the problem. They denote the relative importance of the two terms in objective function (11). Higher accuracy requires more information and, consequently, a higher sensor installation cost. Equation (13) ensures that the estimated demand vector produces the same link flows on the observed links, regardless of the error resulting from the estimation method.

The formulated problem was too complex to be solved using a direct solution algorithm. Each term in (11) constitutes a high-dimensional problem. Solving the first term (i.e., the ODE) is not a straightforward task, where t_w^* is not known in advance. The second term (TSLP) suffers from combinatorial complexity, as shown in (1). Its multiobjective nature adds more complexity to the problem. Interestingly, the method presented in the following section circumvents the difficulties stated in attempting to present Pareto-optimal solutions in the section describing the numerical study.

IV. METHODOLOGY

The main idea of the proposed method is that each transportation network flow results from common user behavior in response to the network structure and demand level. This supports the existence of a strong correlation between the network topology and the traffic flow elements. If we can generate good sampling data, a specified ML technique will learn how the flow propagates and consequently estimate the O–D demand from the observed link flows. Fig. 2 illustrates the general framework of the method proposed herein, and the

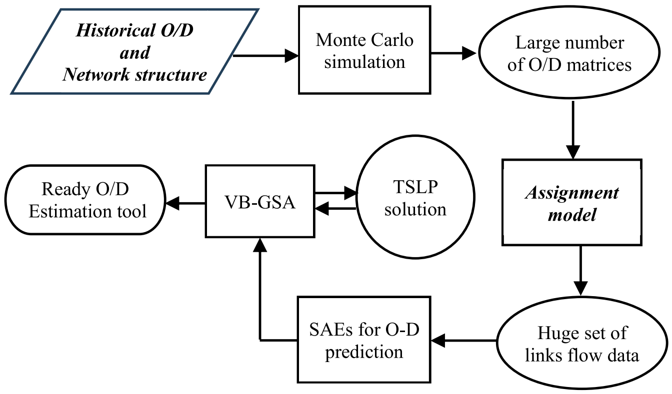


Fig. 2. The ODE-TSLP solution general framework.

subsequent sections illustrate the key steps of the proposed method.

A. Synthetic Input Data Generation

Three demand categories can be dealt with in each ODEP: true unknown demand, reference/historical demand, and estimated demand (T^*, T^0, and $\overline{T}$, respectively). These vectors have the same size as ŵ. To target T^*, the total demand ($\mathcal{D}$) is assumed to be a random variable; therefore, we can create a space of demand vectors T, which can be represented as follows:

$$T = \mathcal{D}F + \gamma, \tag{15}$$

$$F = \frac{1}{\mu_{\mathcal{D}}}\, T^0, \tag{16}$$

where $\mathcal{D}$ is normally distributed with the expected mean $\mu_{\mathcal{D}}$ and the standard deviation $\sigma_{\mathcal{D}}$. $F = [f_w]_{\hat{w}\times 1}$ is the vector of positive real values, where f_w denotes the relative value of w to the total demand $\mathcal{D}$, as in (16). γ is a vector of random variables γ_w with null mean and the standard deviation σ_w. Equations (15) and (16) represent both the independent and correlated stochasticities in T, see [13] and [25].

Synthetic input data can now be generated using a Monte Carlo simulation to create a sample set for training, S_T={T_1, T_2,…..,T_i,……T_n}. The assignment model generates the corresponding link flow vectors S_X={X_1, X_2,…..,X_i,……X_n} to prepare data for the supervised learning step. The assignment formula in (7)–(9) can be solved with the following iterative algorithm:

Algorithm 1 provides the link flow vector X for the corresponding seeded demand vector T. This algorithm was based on the method of successive averages (MSA) [57]. In this algorithm, a search direction is derived at each iteration by assigning the flow to stochastic paths generated using the travel costs estimated from the predecessor link flows. In similar assignment algorithms, the step length q is optimized in various ways to help the algorithm reach convergence [58]. In the MSA, a step-length sequence is used: $q^{ctr} = 1/(ctr+1)$.

B. Learning and Estimation Step

In this study, the proposed DL structure has three consecutive essential components: stacked autoencoders (SAEs), fully connected layers (FCL), and finetuning.

Algorithm 1 Deterministic User Equilibrium

Pre-condition: *connected network (G), nonempty T*

Post-condition: *link flows X*

1. *Initialize: Assign each flow t_w to the set of stochastic paths using Dial's method [56], yielding x_a^{ctr} flows on the links.*
2. *Set the counter (ctr=1).*
3. *Update*$\varepsilon_a(x_a^{ctr}) = \varepsilon_a^0\left(1 + \alpha_a\left(\frac{x_a^{ctr}}{Q_a}\right)^{\chi_a}\right)$
4. *Reassign t_w to the newly generated stochastic paths with ε_a^{ctr}, yielding x_a^{ctr} a set of auxiliary flows (ω_a^{ctr}).*
5. *Update* x_a^{ctr+1}=$x_a^{ctr} + q^{ctr}(\omega_a^{ctr} - x_a^{ctr})$,$\forall a \in A$
6. *If* $\frac{\sqrt{\sum_a (x_a^{ctr+1} - x_a^{ctr})^2}}{\sum_a x_a^{ctr}} \leq \kappa$, *go to the next step; otherwise, set ctr:=ctr+1 and go to step 3.*
7. *Return* $X = \{x_a^{ctr+1}\}$
8. *End algorithm*

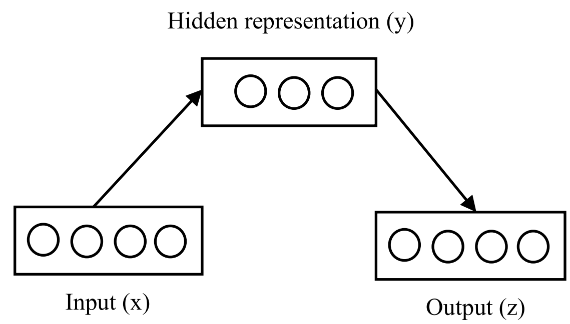


Fig. 3. An auto-encoder structure.

In the first stage, the SAE model was pretrained using the greedy layer-wise (GLW) pretraining in an unsupervised pretraining procedure. GLW pretraining aims to initialize the weights of SAEs with sensible values that may mirror the input data characteristics. In the pretraining for the FCL stage, a supervised technique was used to perform the O–D estimation at the last level. The backpropagation (BP) algorithm is typically used to update network weights, where the SAE model output of the last layer is fed immediately to the FCL. Finally, in the finetuning stage, the stack with the estimator layer (i.e., the entire architecture) is finetuned using the BP algorithm in the top-down direction.

SAEs are a recent and effective improvement of ANNs [37], composed of a stack of autoencoders/neurons. Each autoencoder acts as a layer-building block [59], [60], [61], [62]. In other words, SAEs can be viewed as a simple form of ANNs to reproduce input data with reduced or the same dimensionality [37], [63], [64], as shown in Fig. 3.

For the supervised learning scheme for the input and output data samples ({X_1, X_2,…..,X_i,……X_n}&{T_1, T_2,…..,T_i,……T_n}), where $X_i \in R^m$, $T_i \in R^{\hat{w}}$, and n is the sample size, S_X is reproduced by SAEs. Each autoencoder encodes an input x into a hidden function *y(x),* as shown in (17). Then, *y(x)* is decoded to reconstruct the representation *z(x),* see (18). Both the decoding and encoding data are stored in matrices U_1, U_2, B_1, and B_2.

$$y(x) = f\left(u_1 x + b_1\right), \tag{17}$$

$$z(x) = g\left(u_2 y(x) + b_2\right), \tag{18}$$

where u_1 and b_1 are the encoding weight and bias, respectively; and u_2 and b_2 are the decoding weight and bias, respectively. The sigmoid function (*1 / (1 + exp (-x)*) was used as the encoding activation function (*f(x)*), whereas a linear function was used as the decoding activation function (*g(x)*).

To control the overfitting tendency of DL models and help the SAEs learn the inherent relationships among the data points, the size of the hidden layer should be smaller than that of the input layer. With larger hidden layer sizes, the model is more likely to learn the identity function so that the estimated values perfectly imitate the learned values without true learning [59]. Moreover, we can impose restrictions on the hidden layer structure, such as denoising restrictions and sparsity [37], [62]. Sparsity constraints on the hidden layer representation show a better prediction/classification performance than denoising constraints, particularly in high-dimensional spaces [65], [66].

A new element can be defined in the proposed representation, called a sparse autoencoder, which is simply an autoencoder with sparsity constraints. These constraints are considered an urgent element of the problem because transportation networks usually consist of thousands of links. A cost function (C_{sparse}) is optimized to train the sparse autoencoder. C_{sparse} is the sum of the three components used to denote the adjusted mean squared error between the output (z) and the input (x).

The first component is the mean error function $L(x,\ z)$, which denotes the mean-sum-of-squares error for all the training data, as follows:

$$L(x,\ z) = \frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{1}{2}\|x_i - z_i\|^2. \tag{19}$$

The second component is a regularization factor called the weight decay function (Ω_{weight}). It acts as a penalty function to reduce the values of the weights and, consequently, reduces the likelihood of overfitting as follows:

$$\Omega_{weight} = \frac{1}{2}\sum\nolimits_{l=1}^{n_l-1}\sum\nolimits_{i=1}^{S_l}\sum\nolimits_{j=1}^{S_{l+1}}\left(u_{ij}^{l}\right)^2, \tag{20}$$

where S_l is the number of units in layer l, and n_l is the total layer number. u_{ij}^{l} is the weight associated with the connection between unit j in layer $l+1$ and unit i in layer l.

The third component is the Kullback–Leibler (KL) divergence function [67]. This acts as a sparsity penalty term that is added to C_{sparse} to penalize active hidden units, regardless of the number of hidden units used. The KL function is defined as follows:

$$\mathrm{KL}\left(\rho||\hat{\rho}_j\right) = \left(\rho\, log\,\frac{\rho}{\hat{\rho}_j} + (1-\rho)\, log\,\frac{1-\rho}{1-\hat{\rho}_j}\right), \tag{21}$$

$$\hat{\rho}_j = \frac{1}{n}\sum\nolimits_{i=1}^{n} J_j(x_i), \tag{22}$$

where ρ is a small value (close to zero) that acts as a sparsity parameter, and $\hat{\rho}_j$ is the average of the activation function value $J_j(x_i)$ of the hidden unit j over the training dataset. $KL\left(\rho||\hat{\rho}_j\right)$ is zero if the sparsity parameter equals $\hat{\rho}_j$; otherwise, the KL function increases monotonically as ρ becomes sparser from $\hat{\rho}_j$. Therefore, the cost function C_{sparse} can be expressed as follows:

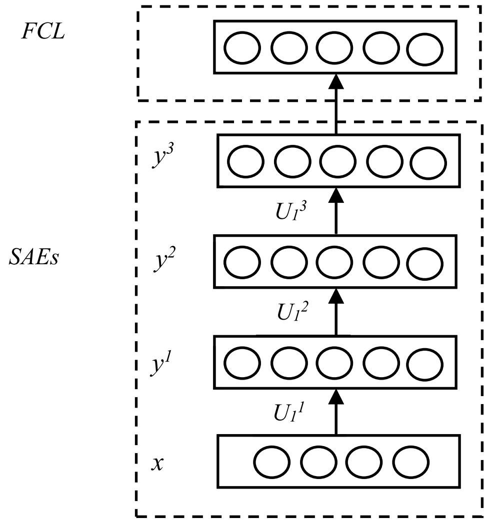


Fig. 4. A SAEs/GLW in connection with the FCL.

$$\begin{aligned} C_{Sparse} = &\frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{1}{2}\|x_i - z_i\|^2 \\ &+ \lambda\frac{1}{2}\sum\nolimits_{l=1}^{n_l-1}\sum\nolimits_{i=1}^{S_l}\sum\nolimits_{j=1}^{S_{l+1}}\left(u_{ij}^{l}\right)^2 \\ &+ \beta\sum\nolimits_{j=1}^{s_h}\left(\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}\right), \end{aligned} \tag{23}$$

where λ and β are weight factors reflecting the relative importance of both the weight decay term and the sparsity penalty term in the cost function. S_h is the number of units in the hidden layer.

The GLW pretraining was used to pretrain the SAEs, as in the procedure introduced in [68], which has been demonstrated with DL models. Fig. 4 illustrates the GLW unsupervised training for the SAEs. The SAEs were trained from bottom to top using an autoencoder. At this stage, the first sparse autoencoder was trained in an unsupervised manner. Subsequently, its hidden representation (y) fed the second sparse autoencoder to be trained as in the first one. This process was performed sequentially for all the sparse autoencoders in the SAEs. In another format, multiple sparse autoencoders can be pretrained by GLW pretraining while being stacked hierarchically [59], [63], [68]. Finally, FCL was applied to the traffic flow estimation step to produce the T vector.

The RMSE or simply RE is a quadratic scoring rule that assesses the typical error magnitude. The mean absolute percentage error (MAPE) is a linear score indicating that all the individual differences contribute equally to the mean. These two indices were selected as performance metrics for the expected traffic flow generated by the proposed design. The RE and MAPE were used to assess the deviation from the projected values as follows:

$$RE = \sqrt{\frac{\sum_{w=1}^{\hat{w}}\left(\frac{t_w^{true} - t_w^{est.}}{t_w^{true}}\right)^2}{\hat{w}}}, \tag{24}$$

$$MAPE = \frac{\sum_{w=1}^{\hat{w}}\left\lceil\frac{t_w^{true} - t_w^{est.}}{t_w^{true}}\right\rceil}{\hat{w}}, \tag{25}$$

where t_w^{true} represents the seeded T_i values in the assignment step, and $t_w^{est.}$ denotes the estimated flow.

C. GSA Step

In general, GSA methods have the drawback of systematically examining a vast search space for a problem. For m explanatory factors with t levels, a grid with t^m points is searched. Several techniques, such as crude Monte Carlo (MC), Latin hypercube, and quasi random sequences (QRS), have been proposed. The DL approach offers the computational power to provide a large number of samples/realizations to help the GSA provide robust results. Here, the generated demand realizations provide indices that are stable against increasing sample set size. This aimed robustness originates from two factors: generating a sufficiently large sample size and using a good perturbation technique in the generating mechanism (such as MC simulation) so that it would equally cover the input factor space with the least possible gaps.

Consider the structured DL model as a function that maps the link flows to the total demand as follows:

$$\mathcal{D} = f(x_1, x_2, \ldots, x_a = x_i, \ldots., x_m) : R^m_{\rightarrow} R, \qquad (26)$$

where $P_{\mathcal{D}}$ is the marginal distribution of $\mathcal{D}$. Suppose ζ is an index that measures the difference between any two distributions with the property $\zeta(P, P) = 0$. The VBSA index for a single input X_i can then be derived as

$$SI(x_a) = \zeta(\mathrm{P}_{\mathcal{D}}, P_{\mathcal{D}|xa=xi}). \qquad (27)$$

$SI(x_a)$ denotes the LSA index of the dependent variable ($\mathcal{D}$) when the distribution is conditioned to a determined value for the explanatory factor ($x_a = x_i$). Conditionality represents the total local effect of x_a, which is assumed to be fixed, whereas all other variables can vary. To obtain the GSA version, we performed an outer expectation over $SI(x_a)$ to include the effect of all x_a flow domains on the unconditional $\mathcal{D}$ distribution (P_D).

$$GSI_a = E[SI(x_a = x_i)] \qquad (28)$$

The total global sensitivity index (GSI) was calculated using (28) by allowing the link under study to take several values. The expectation in (28) was taken over all the other variable changes ($x_{\sim a}$). For the VB-GSA, the sensitivity index is formally defined as

$$GSI_a = \frac{E_{X_a}(V_{x_a}[\mathcal{D}/x_{\sim a}])}{V[\mathcal{D}]} = 1 - \frac{V_{x_{\sim a}}[E_{x_a}(\mathcal{D}/x_{\sim a})]}{V[\mathcal{D}]}. \qquad (29)$$

The VBSA index in (29) was measured directly from the variance reduction by fixing one variable. The total effect of GSI was measured by the residual contribution of the first-order and higher-order indices for that variable in determining the response variance. The index should vary between zero and one. $GSI_a \cong 0$ is a necessary and sufficient condition to conclude that link (a) could be neglected in studying the ODEP; consequently, a sensor does not need to be installed on that link.

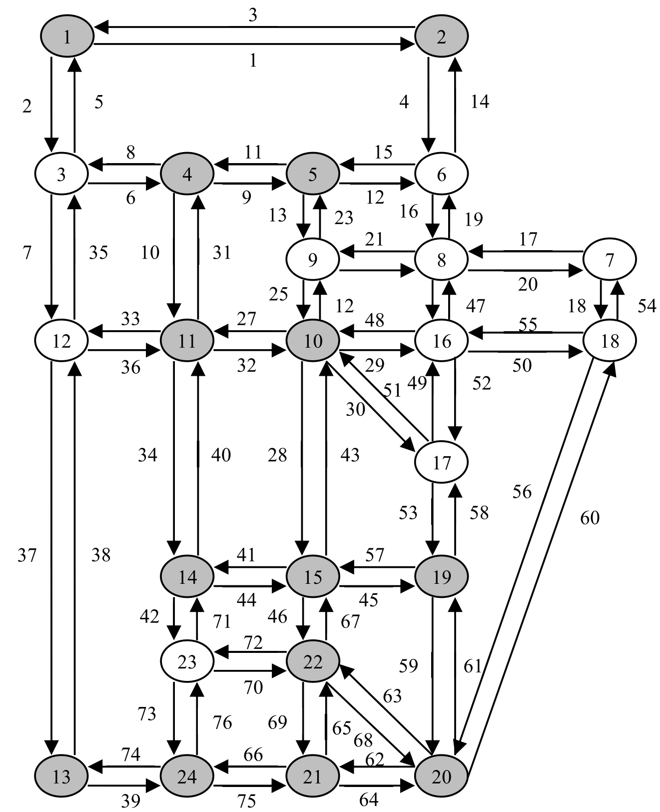


Fig. 5. Sioux falls network structure.

V. Numerical Study

Two road network datasets were employed in the numerical tests to determine the effectiveness of the proposed framework in the solution process. The medium network in Section A is a well-known transportation network in the literature. An artificial network is developed in Section B to highlight the advantages of DL in terms of large-scale network management, and to allow flexibility in developing oriented/directed practical solutions.

As the traffic flow data for every network are highly linked to the network topology, the proposed architecture must be trained independently for each network under consideration. Based on the assumption of error-free sensors, the link flow information from the sensors was assumed to correspond to the findings derived from the assigned demand using the SUE assignment model. Therefore, the flow in the links with the sensors is believed to be known. The entire DL architecture was developed using the MATLAB toolbox [69], which was processed on a PC with an Intel Core I7, 2.8 GHz CPU, and 16 GB of RAM, which is adequate for the chosen network sizes.

A. Sioux Falls Network

This is a fully reported medium network that consists of 76 links, 182 O–D pairs, and 24 nodes (refer to Fig. 5), where the shaded nodes represent both trip origins/destinations, and the link coding numbers are provided. This network was first

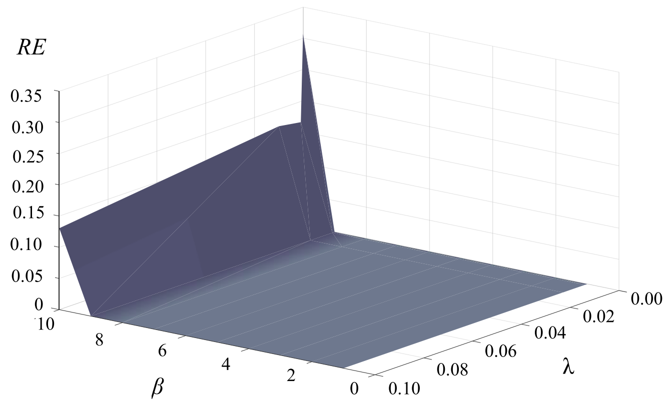


Fig. 6. Adjusting Alpha (λ) and Beta (β) according to the model's *RE*.

introduced for the TSLP problem in [16]. It is considered a benchmark problem in most TSLP studies [1], [16], [29], [31], [40], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79]. The reference demand for the Sioux Falls network supplied by [25] is utilized in conjunction with zero γ_w and σ_w equal to $(0.25t_w^0)$.

Various values of λ and β were employed, and the relative inaccuracy of each model was reported. The model achieved almost the same relative error for all values of λ and β, with the exception of $\beta = 10$. Therefore, the default settings of MATLAB were used. The relationship between the demand sample size ($|S_T| = n$) and the demand flow relative error (RE) is depicted in Fig. 6, which compares the model performance when applying shallow learning (no SAEs) against applying a stack of varying numbers of AEs (one, two, three, and four) with 10 units in each. As shown in the graph, there is a significant difference between the use of shallow learning (no SAEs) and the use of SAEs. The model achieved a lower RE percentage for all sample sizes when the SAEs were used than when shallow learning was used. While increasing the number of AEs in the stack decreased the RE percentage, the performance of the model hardly improved when four AEs were used. Consequently, the SAE model was implemented with three hidden layers (three AEs), with 10 units in each hidden layer; λ and β were, respectively, set to 0.04 and 4. The learning rate was set to 0.01. In Fig. 7, the effect of the sample size on the accuracy is shown because any ML technique requires a sufficient number of points to achieve the required learning. The RE value converged at a sample size of 1000 demand vectors.

To identify the most critical link information in the DL prediction steps, (28) was calculated numerically following the steps described in [44]. Fig. 8 shows the GSI of each link, where the significance is sorted in descending order. The stepwise selection of links is shown in Fig. 9, where the RE is estimated for each selection set. This draw is equivalent to Pareto-optimal solutions, because the analyst can select a suitable solution according to the required accuracy and available budget to purchase the sensors.

B. Large-Scale Network

This subsection demonstrates the application of the proposed technique to larger networks. A 900-node grid network

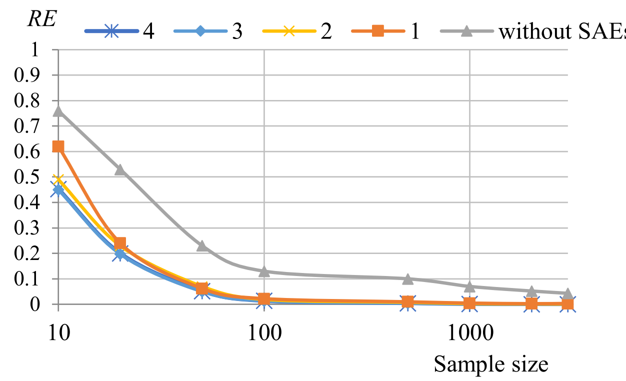


Fig. 7. Sample size effect on model prediction accuracy.

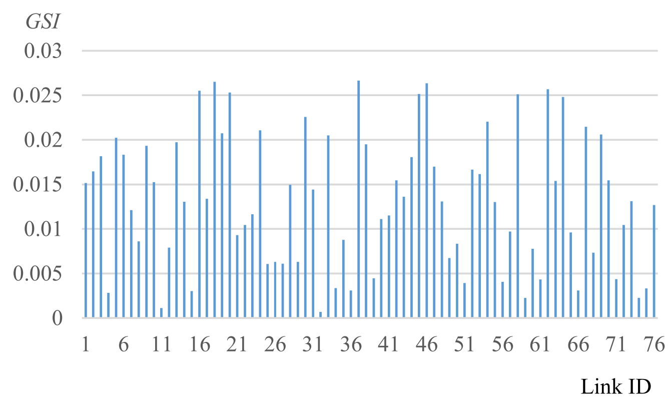


Fig. 8. Global sensitivity analysis for link flow information importance in predicting O-D values for Sioux Falls network.

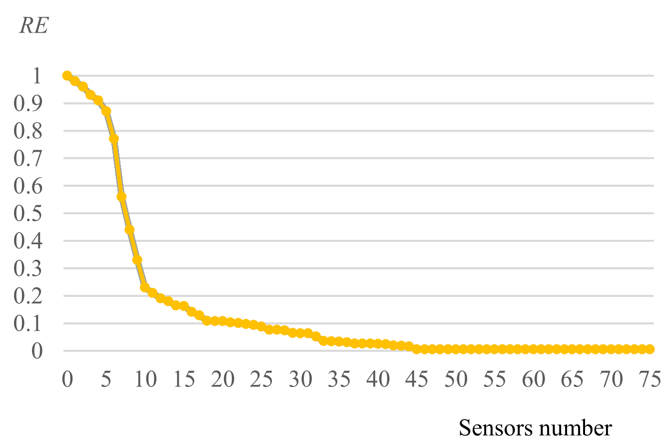


Fig. 9. Sensors number vs. attained accuracy for Sioux Falls network.

was constructed to evaluate the effectiveness of the proposed method. The network comprised 900 nodes and 3,480 links. The connection trip time was set to one minute.

The network had a reasonable arc-to-node ratio of 3.87, although most synthetically built large networks in the literature have values as large as 10 [80]. The vertices of the network consisted of approximately 810,000 node pairs. It is impractical to consider all these node pairs as O–D pairs. Therefore, the demand density was assumed to be 5%, resulting in 40,500 demand pairs. The budget was limited to 10% of the number of links to be equipped. The O–D prior matrix was generated randomly with values ranging from 15 to 100 units/h. The remaining parameter values were set to those

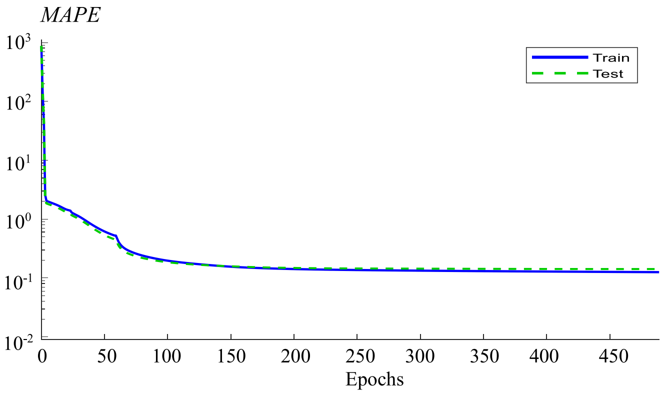


Fig. 10. SAEs curves of learning for testing and training data.

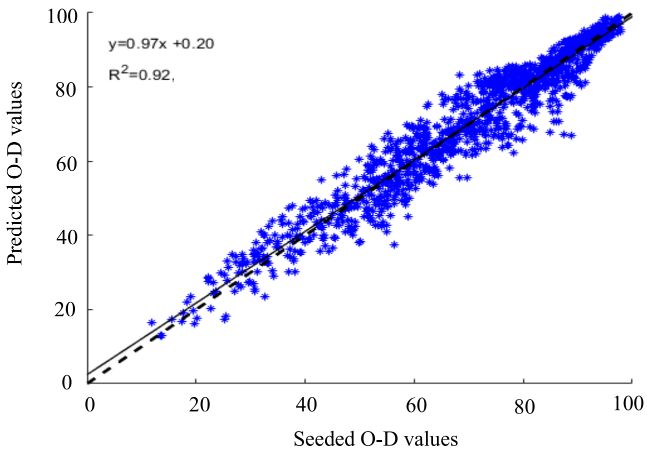


Fig. 11. Seeded-predicted O-D pair plot with LOE for the large-scale network.

of the Sioux Falls network. Fig. 10 shows the performance of the proposed DL architecture during the training and testing stages for a large-scale network. Fig. 11 depicts the seeded-predicted plot for each O–D pair, with the best-fitting line compared with the line of equality (LOE) to provide an overview of how closely the forecasts matched the seeded data. Furthermore, other indicators of global bias were investigated, such as the average error, slope, and intercept. These indicators were evaluated by comparing the seeded and predicted O–D values along the LOE, and then fitting an unconstrained linear trend line to the predicted seeded plot [81]. The fitted line allows the assessment of the reliability of the model through a comparison of the points clustered along it (i.e., the best model has the average error, slope, and intercept values close to 0, 1, and 0, respectively). For instance, a model's slope deviation from unity indicates a substantial dependence of prediction errors on the seeded values [82]. In contrast, the nonzero average error and intercept indicate under- or over-prediction [81], [83]. The overall bias indicators, average error, slope, and intercept are -0.08, 0.95, and 0.17, respectively. In addition to the obtained R^2 of 92%, these values reflect the high efficacy of the model for large-scale networks.

VI. Conclusion

This article presented a novel method for the ODE-TSLP based on a comprehensive literature review. The difficulty of the stated combined problem originates from two major points: the impracticality of placing sensors on all network links, and the problematic conditions for converting passive link counts into O–D flow information. The condition examined was that traffic-counting sensors should be installed on network links to achieve the most accurate O–D prediction using the DL architecture. Moreover, a simulation environment for DL training could be obtained by making appropriate traffic assignment model assumptions. The proposed DL architecture comprised a fully interconnected SAE model layer. The SAE model aims to determine hidden O–D flow characteristics. This may facilitate the formation of significant patterns between the O–D and link flows. Furthermore, SAEs can initiate solutions that are near the ideal solution or achieve superior local minima, thereby reducing the convergence time. The FCL was then utilized in the O–D estimation stage. Finally, the GSA explored the significance of the links in the prediction stage to solve the TSLP. The numerical results demonstrated the capability of the method to produce a set of Pareto-optimal solutions. Furthermore, the proposed method handles large-scale networks with an accuracy exceeding 90%. Errors in observations should be addressed in future studies, because they occur in virtually all real-world applications. As flaws may result in a range of problems, several corrective measures are required. Few studies have explored the possibility of information noise using two key themes: measurement error [32] and total sensor failure [8], [33], [34]. The former focuses on minimizing the propagation of measurement errors while accounting for error fluctuations throughout the solution process. Considering the possibility of sensor failure, the latter method seeks to ensure minimal loss of acquired flow information. Although observational errors were not the major focus of this study, the topic merits a separate study. It would be a new challenge for DL to handle the two types of uncertainty simultaneously (i.e., sensor errors and seeded demand fluctuations). Furthermore, the provided DL training scheme can be extended to a time-dependent network representation, where all the remaining network link flows and travel times can be identified in different time steps from a subset of sensor traffic data. This will help develop a robust vehicle-routing algorithm for time-dependent networks.

References

[1] M. Owais, G. S. Moussa, and K. F. Hussain, "Sensor location model for O/D estimation: Multi-criteria meta-heuristics approach," *Oper. Res. Perspect.*, vol. 6, Jan. 2019, Art. no. 100100.

[2] M. Owais and A. E. Matouk, "A factorization scheme for observability analysis in transportation networks," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114727.

[3] Y. Nie, H. M. Zhang, and D.-H. Lee, "Models and algorithms for the traffic assignment problem with link capacity constraints," *Transp. Res. B, Methodol.*, vol. 38, no. 4, pp. 285–312, May 2004.

[4] A. Chen, P. Chootinan, S. Ryu, and S. C. Wong, "Quality measures of origin-destination trip table estimated from traffic counts: Review and new generalized demand scale measure," *J. Transp. Eng.*, vol. 138, no. 11, pp. 1340–1349, Nov. 2012.

[5] M. Owais, M. E. Deeb, and Y. A. Abbas, "Distributing portable excess speed detectors in AL Riyadh City," *Int. J. Civil Eng.*, vol. 18, no. 11, pp. 1301–1314, Nov. 2020.

[6] M. Owais, "Traffic sensor location problem: Three decades of research," *Expert Syst. Appl.*, vol. 208, Dec. 2022, Art. no. 118134.

[7] M. Owais and A. I. Shahin, "Exact and heuristics algorithms for screen line problem in large size networks: Shortest path-based column generation approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24829–24840, Dec. 2022.

[8] M. Salari, L. Kattan, W. H. K. Lam, M. A. Esfeh, and H. Fu, "Modeling the effect of sensor failure on the location of counting sensors for origin-destination (OD) estimation," *Transp. Res. C, Emerg. Technol.*, vol. 132, Nov. 2021, Art. no. 103367.

[9] M. Gaudry and L. Lamarre, "Estimating origin-destination matrices from traffic counts: A simple linear intercity model for Quebec," *Logistics Transp. Rev.*, vol. 15, no. 5, pp. 1–12, 1979.

[10] W. H. K. Lam and H. P. Lo, "Estimation of origin-destination matrix from traffic counts: A comparison of entropy maximizing and information minimizing models," *Transp. Planning Technol.*, vol. 16, no. 2, pp. 85–104, Nov. 1991.

[11] E. Cascetta, "Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator," *Transp. Res. B, Methodol.*, vol. 18, nos. 4–5, pp. 289–299, Aug. 1984.

[12] H. Spiess, "A maximum likelihood model for estimating origin-destination matrices," *Transp. Res. B, Methodol.*, vol. 21, no. 5, pp. 395–412, Oct. 1987.

[13] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using Bayesian networks," *Transp. Res. B, Methodol.*, vol. 42, no. 5, pp. 482–509, Jun. 2008.

[14] W. Lam and H. Lo, "Accuracy of OD estimates from traffic counts," *Traffic Eng. Control*, vol. 31, no. 6, pp. 358–367, Jun. 1990.

[15] H. Yang, Y. Iida, and T. Sasaki, "An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts," *Transp. Res. B, Methodol.*, vol. 25, no. 5, pp. 351–363, Oct. 1991.

[16] H. Yang and J. Zhou, "Optimal traffic counting locations for origin-destination matrix estimation," *Transp. Res. B, Methodol.*, vol. 32, no. 2, pp. 109–126, Feb. 1998.

[17] L. Bianco, G. Confessore, and P. Reverberi, "A network based model for traffic sensor location with implications on O/D matrix estimates," *Transp. Sci.*, vol. 35, no. 1, pp. 50–60, Feb. 2001.

[18] A. Ehlert, M. G. H. Bell, and S. Grosso, "The optimisation of traffic count locations in road networks," *Transp. Res. B, Methodol.*, vol. 40, no. 6, pp. 460–479, Jul. 2006.

[19] M. Bierlaire, "The total demand scale: A new measure of quality for static and dynamic origin–destination trip tables," *Transp. Res. B, Methodol.*, vol. 36, no. 9, pp. 837–850, Nov. 2002.

[20] E. Castillo, A. J. Conejo, J. M. Menéndez, and P. Jiménez, "The observability problem in traffic network models," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 23, no. 3, pp. 208–222, Apr. 2008.

[21] E. Castillo, P. Jimenez, J. M. Menendez, and A. J. Conejo, "The observability problem in traffic models: Algebraic and topological methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 275–287, Jun. 2008.

[22] E. Castillo, I. Gallego, S. Sánchez-Cambronero, and A. Rivas, "Matrix tools for general observability analysis in traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 799–813, Dec. 2010.

[23] A. Chen, S. Pravinvongvuth, P. Chootinan, M. Lee, and W. Recker, "Strategies for selecting additional traffic counts for improving O-D trip table estimation," *Transportmetrica*, vol. 3, no. 3, pp. 191–211, Jan. 2007.

[24] X. Fei, H. S. Mahmassani, and S. M. Eisenman, "Sensor coverage and location for real-time traffic prediction in large-scale networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2039, no. 1, pp. 1–15, Jan. 2007.

[25] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Traffic estimation and optimal counting location without path enumeration using Bayesian networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 23, no. 3, pp. 189–207, Apr. 2008.

[26] X. Fei and H. S. Mahmassani, "Structural analysis of near-optimal sensor locations for a stochastic large-scale network," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 3, pp. 440–453, Jun. 2011.

[27] C. Antoniou et al., "Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation," *Transp. Res. C, Emerg. Technol.*, vol. 66, pp. 79–98, May 2016.

[28] M. G. H. Bell, C. M. Shield, F. Busch, and G. Kruse, "A stochastic user equilibrium path flow estimator," *Transp. Res. C, Emerg. Technol.*, vol. 5, nos. 3–4, pp. 197–210, Aug. 1997.

[29] A. Chen, P. Chootinan, and W. W. Recker, "Examining the quality of synthetic origin–destination trip table estimated by path flow estimator," *J. Transp. Eng.*, vol. 131, no. 7, pp. 506–513, Jul. 2005.

[30] M. Gentili and P. B. Mirchandani, "Locating sensors on traffic networks: Models, challenges and research opportunities," *Transp. Res. Part C: Emerg. Technol.*, vol. 24, pp. 227–255, Oct. 2012.

[31] C. Fu, N. Zhu, S. Ling, S. Ma, and Y. Huang, "Heterogeneous sensor location model for path reconstruction," *Transp. Res. B, Methodol.*, vol. 91, pp. 77–97, Sep. 2016.

[32] M. Shao, C. Xie, and L. Sun, "Optimization of network sensor location for full link flow observability considering sensor measurement error," *Transp. Res. C, Emerg. Technol.*, vol. 133, Dec. 2021, Art. no. 103460.

[33] W. Sun, H. Shao, T. Wu, F. Shao, and E. Z. Fainman, "Reliable location of automatic vehicle identification sensors to recognize origin-destination demands considering sensor failure," *Transp. Res. C, Emerg. Technol.*, vol. 136, Mar. 2022, Art. no. 103551.

[34] M. Salari, L. Kattan, W. H. K. Lam, H. P. Lo, and M. A. Esfeh, "Optimization of traffic sensor location for complete link flow observability in traffic network considering sensor failure," *Transp. Res. B, Methodol.*, vol. 121, pp. 216–251, Mar. 2019.

[35] S.-R. Hu, S. Peeta, and H.-T. Liou, "Integrated determination of network origin–destination trip matrix and heterogeneous sensor selection and location strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 195–205, Jan. 2016.

[36] S. Goswami and A. Kumar, "Traffic flow prediction using deep learning techniques," in *Proc. Int. Conf. Comput. Sci., Commun. Secur.* Cham, Switzerland: Springer, 2022, pp. 198–213.

[37] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[38] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022.

[39] K. Guo et al., "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, Feb. 2021.

[40] M. Owais, G. S. Moussa, and K. F. Hussain, "Robust deep learning architecture for traffic flow estimation from a subset of link sensors," *J. Transp. Eng., A, Syst.*, vol. 146, no. 1, Jan. 2020, Art. no. 04019055.

[41] A. Alshehri, M. Owais, J. Gyani, M. H. Aljarbou, and S. Alsulamy, "Residual neural networks for origin–destination trip matrix estimation from traffic sensor information," *Sustainability*, vol. 15, no. 13, p. 9881, Jun. 2023.

[42] G. S. Moussa and M. Owais, "Pre-trained deep learning for hot-mix asphalt dynamic modulus prediction with laboratory effort reduction," *Construct. Building Mater.*, vol. 265, Dec. 2020, Art. no. 120239.

[43] G. S. Moussa and M. Owais, "Modeling hot-mix asphalt dynamic modulus using deep residual neural networks: Parametric and sensitivity analysis study," *Construct. Building Mater.*, vol. 294, Aug. 2021, Art. no. 123589.

[44] G. S. Moussa, M. Owais, and E. Dabbour, "Variance-based global sensitivity analysis for rear-end crash investigation using deep learning," *Accident Anal. Prevention*, vol. 165, Feb. 2022, Art. no. 106514.

[45] A. Saltelli et al., *Global Sensitivity Analysis: The Primer*. Hoboken, NJ, USA: Wiley, 2008.

[46] F. Pianosi et al., "Sensitivity analysis of environmental models: A systematic review with practical workflow," *Environ. Model. Softw.*, vol. 79, pp. 214–232, May 2016.

[47] A. Sieber and S. Uhlenbrook, "Sensitivity analyses of a distributed catchment model to verify the model structure," *J. Hydrol.*, vol. 310, nos. 1–4, pp. 216–235, Aug. 2005.

[48] M. Owais, "Analysing Witczak 1-37A, Witczak 1-40D and modified Hirsch models for asphalt dynamic modulus prediction using global sensitivity analysis," *Int. J. Pavement Eng.*, vol. 24, no. 1, pp. 1–37, Dec. 2023, Art. no. 2268808.

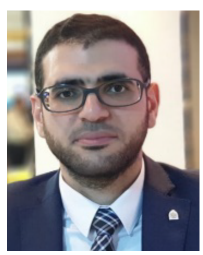
[49] J. Nossent, P. Elsen, and W. Bauwens, "Sobol' sensitivity analysis of a complex environmental model," *Environ. Model. Softw.*, vol. 26, no. 12, pp. 1515–1525, Dec. 2011.

[50] L. K. Idriss and M. Owais, "Global sensitivity analysis for seismic performance of shear wall with high-strength steel bars and recycled aggregate concrete," *Construct. Building Mater.*, vol. 411, Jan. 2024, Art. no. 134498.

[51] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, May 1991.

[52] M. J. Maher, "Inferences on trip matrices from observations on link volumes: A Bayesian statistical approach," *Transp. Res. B, Methodol.*, vol. 17, no. 6, pp. 435–447, Dec. 1983.

[53] M. G. H. Bell, "The estimation of an origin-destination matrix from traffic counts," *Transp. Sci.*, vol. 17, no. 2, pp. 198–217, May 1983.
[54] Y. Nie, H. M. Zhang, and W. W. Recker, "Inferring origin–destination trip matrices with a decoupled GLS path flow estimator," *Transp. Res. B, Methodol.*, vol. 39, no. 6, pp. 497–518, Jul. 2005.
[55] H. Spiess, "Technical note—Conical volume-delay functions," *Transp. Sci.*, vol. 24, no. 2, pp. 153–158, May 1990.
[56] C. Fisk, "Some developments in equilibrium traffic assignment," *Transp. Res. B, Methodol.*, vol. 14, no. 3, pp. 243–255, Sep. 1980.
[57] W. B. Powell and Y. Sheffi, "The convergence of equilibrium algorithms with predetermined step sizes," *Transp. Sci.*, vol. 16, no. 1, pp. 45–55, Feb. 1982.
[58] M. Maher, "Algorithms for logit-based stochastic user equilibrium assignment," *Transp. Res. B, Methodol.*, vol. 32, no. 8, pp. 539–549, Nov. 1998.
[59] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2007, p. 153.
[60] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[61] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Mar. 2010.
[62] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[63] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[64] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: A comparison between traditional and deep learning neural networks," *Int. J. Big Data Intell.*, vol. 3, no. 1, p. 28, 2016.
[65] E. Hosseini-Asl, J. M. Zurada, and O. Nasraoui, "Deep learning of part-based representation of data using sparse autoencoders with non-negativity constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2486–2498, Dec. 2016.
[66] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 873–880.
[67] F. Perez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2008, pp. 1666–1670.
[68] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[69] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," M.S. thesis, DTU-Inform., Tech. Univ. Denmark, Lyngby, Denmark, 2012.
[70] H. Yang, L. Gan, and W. Tang, "Determing cordons and screen lines for origin-destination trip studies," *Proc. Eastern Asia Soc. Transp. Stud.*, vol. 3, no. 2, pp. 1–12, 2001.
[71] C. Yang, P. Chootinan, and A. Chen, "Traffic counting location planning using genetic algorithm," *J. Eastern Asia Soc. Transp. Stud.*, vol. 5, pp. 898–913, Oct. 2003.
[72] S.-R. Hu, S. Peeta, and C.-H. Chu, "Identification of vehicle sensor locations for link-based network traffic applications," *Transp. Res. B, Methodol.*, vol. 43, nos. 8–9, pp. 873–894, Sep. 2009.
[73] S.-X. He, "A graphical approach to identify sensor locations for link flow inference," *Transp. Res. B, Methodol.*, vol. 51, pp. 65–76, May 2013.
[74] Y. Liu, N. Zhu, S. Ma, and N. Jia, "Traffic sensor location approach for flow inference," *IET Intell. Transp. Syst.*, vol. 9, no. 2, pp. 184–192, Mar. 2015.
[75] M. Zangui, Y. Yin, and S. Lawphongpanich, "Sensor location problems in path-differentiated congestion pricing," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 217–230, Jun. 2015.
[76] M. Hadavi and Y. Shafahi, "Vehicle identification sensor models for origin–destination estimation," *Transp. Res. B, Methodol.*, vol. 89, pp. 82–106, Jul. 2016.
[77] X. Xu, H. K. Lo, A. Chen, and E. Castillo, "Robust network sensor location for complete link flow observability under uncertainty," *Transp. Res. B, Methodol.*, vol. 88, pp. 1–20, Jun. 2016.
[78] M. Owais, "Location strategy for traffic emission remote sensing monitors to capture the violated emissions," *J. Adv. Transp.*, vol. 2019, pp. 1–9, Mar. 2019.
[79] M. Owais and A. Alshehri, "Pareto optimal path generation algorithm in stochastic transportation networks," *IEEE Access*, vol. 8, pp. 58970–58981, 2020.
[80] F. B. Zhan and C. E. Noon, "Shortest path algorithms: An evaluation using real road networks," *Transp. Sci.*, vol. 32, no. 1, pp. 65–73, Feb. 1998.
[81] G. Al-Khateeb, A. Shenoy, N. Gibson, and T. Harman, "A new simplistic model for dynamic modulus predictions of asphalt paving mixtures," *J. Assoc. Asphalt Paving Technologists*, vol. 75, pp. 1–40, Jan. 2006.
[82] J. Li, A. Zofka, and I. Yut, "Evaluation of dynamic modulus of typical asphalt mixtures in northeast U.S. region," *Road Mater. Pavement Des.*, vol. 13, no. 2, pp. 249–265, Jun. 2012.
[83] H. Azari, G. Al-Khateeb, A. Shenoy, and N. Gibson, "Comparison of simple performance test |E*| of accelerated loading facility mixtures and prediction |E*|: Use of NCHRP 1-37A and Witczak's new equations," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1998, no. 1, pp. 1–9, Jan. 2007.



Mahmoud Owais received the M.S. degree in transportation planning and the Ph.D. degree in transit planning from the University of Assiut in 2011 and 2014, respectively. He is currently an Associate Professor in transportation planning and traffic engineering with the Civil Engineering Department, Faculty of Engineering, Assiut University. He has published about 30 studies and conducted 220 reviews in well-reputed journals, such as IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Expert Systems with Applications*, and *Accident Analysis & Prevention*. For more information, visit the link: https://scholar.google.com.eg/citations?user=UNwlx2MAAAAJ&hl=en.