DeepRisk: A Deep Transfer Learning Approach to Migratable Traffic Risk Estimation in Intelligent Transportation using Social Sensing

Yang Zhang, Hongxiao Wang, Daniel Zhang, Dong Wang
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
{yzhang42, hwang21, yzhang40, dwang5}@nd.edu

***Abstract*—This paper focuses on the *migratable traffic risk estimation* problem in intelligent transportation systems using the social (human-centric) sensing. The goal is to accurately estimate the traffic risk of a *target area* where the ground truth traffic accident reports are not available by leveraging an estimation model from a *source area* where such data is available. Two important challenges exist. The first challenge lies in the discrepancy between source and target areas (e.g., layouts, road conditions, and local regulations) and such discrepancy would prevent a direct application of a model from the source area to the target area. The second challenge lies in the difficulty of identifying all potential features in the migratable traffic risk estimation problem and decide the importance of identified features due to the lack of ground truth labels in the target area. To address these challenges, we develop DeepRisk, a social sensing based migratable traffic risk estimation scheme using deep transfer learning techniques. The evaluation results on a real world dataset in New York City show the DeepRisk significantly outperforms the state-of-the-art baselines in accurately estimating the traffic risk of locations in a city.**

***Keywords*-Social Sensing, Intelligent Transportation, Migratable Traffic Risk Estimation, Deep Transfer Learning**

I. Introduction

Social sensing has emerged as a new networked sensing paradigm that uses humans as sensors to report the states of the physical world [1]. Examples of social sensing applications include tracking the real-time traffic condition using mobile crowdsensing [2], obtaining the real-time disaster and emergency awareness using online social media [3], and monitoring the urban air quality using reports from citizens [4]. Compared to traditional sensing paradigms that use physical sensors, social sensing has been shown to be more pervasive, scalable and economic in its applications [5]. In this paper, we focus on a *migratable traffic risk estimation* problem in intelligent transportation systems using social sensing. Our goal is to accurately estimate the traffic accident rate of locations in a *target area* where the ground truth traffic accident reports are *not available* by leveraging an estimation model from a *source area* where such data is available.

Previous efforts have been made towards addressing the traffic risk estimation problem in data mining, networked sensing, and intelligent transportation systems [6]–[10]. Those solutions primarily rely on historic ground truth labels of traffic accidents in the studied area to build reliable estimation models. However, such ground truth data is not always available in many places due to various resource constraints and privacy/legal concerns. For example, less than 1% of cities in United States have open web portals to access their traffic accident data [1]. The traffic monitoring devices are prohibited by law in 10 states in United States [2]. On the other hand, the rich traffic information collected by some widely deployed mobile crowdsensing applications (e.g., Waze) is privately owned by the companies and not available to the public access [11]. The lack of publicly available ground truth data on traffic accidents presents a fundamental challenge to the traffic risk estimation problem.

To address the above challenge, this paper develops a social sensing based deep transfer learning solution to estimate the traffic risk of locations in an area where the ground truth traffic accident data is not available. For example, consider the traffic risk estimation problem in two cities near our campus: South Bend and Mishawaka. The two cities are both located in the northern Indiana region with similar population densities (i.e., 2,457/sq mi vs. 2,765/sq mi) and weather conditions (i.e., long snow season). However, the city of South Bend has an open data portal for accessing traffic accident data of the city [3] motivated by its smart city initiative but Mishawaka does not. In this example, our goal is to estimate the traffic risk of locations in Mishawaka (target area) by "migrating" the traffic risk estimation model learned in South Bend (source area). Such a migratable traffic estimation problem is not trivial to solve due to several technical challenges elaborated below.

Discrepancy Between Source and Target Areas. A simple solution to address the migratable traffic risk estimation problem is to directly apply the estimation model learned from the source area to estimate the traffic risks in the target area. However, a major issue of this solution is that the target and source areas may be different in many aspects (e.g., layouts, road conditions, traffic volumes, and local regulations) that

[1] https://www.forbes.com/sites/metabrown/2017/06/30/quick-links-to-municipal-open-data-portals-for-85-us-cities/#274a68962290

[2] https://www.iihs.org/iihs/topics/laws/automated_enforcement/enforcementtable?topicName=speed

[3] https://data-southbend.opendata.arcgis.com/

would prevent a direct application of a model learned from the source area to the target area [12]. Such discrepancy between the source and target areas can potentially lead to the undesirable overfitting problems in the estimations (i.e., the model learned from the source area might be an overfitted model to estimate traffic risk in the target area) [13]. Therefore, the migrated estimation model needs to explicitly accommodate the discrepancy between different areas.

Complex and Latent Risk Features. The second challenge refers to the fact that it is extremely difficult (if possible) to identify all relevant features in the migratable traffic risk estimation problem due to the arbitrarily large feature space and the latent nature of certain features (e.g., cognitive conditions of drivers, the driving habits of an area) [8]. Furthermore, it is also challenging to decide the importance (e.g., weights) of the identified features in an estimation model when the training data set is insufficient or not available at all (e.g., target areas in our problem). Therefore, it is a not trivial task to automatically identify a critical set of complex and latent traffic risk features and encode them into the migratable estimation model.

To address the above challenges, we develop DeepRisk, a social sensing based migratable traffic risk estimation scheme using deep transfer learning techniques. To address the challenge of discrepancy between source and target areas, we develop a principled deep transfer learning framework to effectively migrate the risk estimation model from the source area to the target area through a novel transformation neural network design. To address the complex and latent risk feature challenge, DeepRisk judiciously learns the traffic risk related features through a novel adversarial learning algorithm and explicitly encodes learned features into deep transfer learning network for the model migration. To the best of our knowledge, the DeepRisk is the first deep transfer learning based approach to address the migratable traffic risk estimation problem in intelligent transportation systems. We evaluate the DeepRisk scheme on a real-world dataset from New York City. The results show that our scheme significantly outperforms the state-of-the-art baselines in terms of accurately estimating the traffic risk of locations in a city.

II. Related Work

A. Social Sensing

Social sensing has emerged as a new sensing paradigm in networked sensing that uses humans as sensors to report the states of the physical world [3], [14], [15]. This new sensing paradigm is a motivated by the proliferation of potable devices for individuals (e.g., smartphone), the ubiquitous wireless communication technology (e.g, 4/5G), and the mass information dissemination media (e.g., Twitter, Facebook) [16], [17]. Examples of social sensing applications include smart urban environment and facility monitoring [4], disaster and emergency response systems [18], edge computing systems [19], and intelligent transportation systems [20]. This paper focuses on the migratable traffic risk estimation problem by leveraging the massive publicly available social sensing data on traffic. Compare to traditional sensing paradigms that collect traffic data from infrastructure-based sensors (e.g., speed sensors, traffic cameras, loop detectors), social sensing provides a more pervasive, scalable, and economic approach to the problem studied in this paper.

B. Traffic Risk Estimation

A significant amount of efforts have been made towards addressing the traffic risk estimation problem in data mining, networked sensing, and intelligent transportation systems [6]–[9]. For example, Yu *et al.* developed a support vector machine based approach to analyze accident injury severity using the crash reports and real-time traffic speed data [6]. Sun developed a dynamic Bayesian network based model to predict the traffic accident risk using the data collected from speed sensors deployed on urban expressways [7]. Yuan *et al.* proposed a deep learning based model to predict the probability of traffic accidents using long-term motor vehicle crash data and traffic camera data [8]. Qin *et al.* developed a large-scale data-driven framework for urban traffic sensing using vehicle GPS and cellular signaling data [9]. However, those approaches cannot be applied to our migratable traffic risk estimation problem because they primarily rely on a rich set of historic ground truth data on traffic accidents or accurate traffic information collected from infrastructure-based physical sensors in the studied area to build reliable estimation/prediction models [21]. In contrast, we develop a novel deep transfer learning approach to estimate the traffic risk of locations in the areas where the ground truth traffic accident data is not available.

C. Deep and Transfer Learning

Our work is also related to deep and transfer learning techniques, which have been applied in areas such as recommendation systems, remote sensing, computer vision, and Internet of Things (IoT) [22]–[25]. Glorot *et al.* proposed a domain adaptation framework for large-scale sentiment classification using the stacked denoising auto-encoder [22]. Xie *et al.* designed a transfer learning solution to identify complex earth surface features from high-resolution satellite imagery for poverty estimation [23]. Lu *et al.* developed a behavioural biometrics based authentication framework for smartwatches using deep recurrent neural network [24]. Bhattacharya *et al.* proposed an off-line deep learning optimizing approach to learn compact model representation for wearable computing [25]. To the best of our knowledge, the DeepRisk is the first social sensing based deep transfer learning approach to solving the migratable traffic risk estimation problem in intelligent transportation systems.

III. Problem Definition

In this section, we formulate the migratable traffic risk estimation problem using social sensing data. We first define the terms that will be used in the problem statement and then formally present our problem.

Definition 1: **Source Area** (S): We define a source area to be an area (e.g., borough, district, city) where the ground truth traffic accident reports are available for traffic risk estimation.

Definition 2: **Target Area** (T): We define a target area to be the studied area where the ground truth traffic accident records are not available.

Definition 3: **Sensing Cell** (C): We divide both the source and target areas into disjoint sensing cells (e.g., $60\text{m} \times 60\text{m}$ squares) where each cell represents a subarea of interest. In particular, we define A and B to be the number of cells in the source and target area, respectively. In particular, we denote c_a^S as the a^{th} sensing cell in the source area ($a = 1, 2, \cdots, A$), and c_b^T as the b^{th} sensing cell in the target area ($b = 1, 2, \cdots, B$).

Definition 4: **Sensing Cycle**: A sensing cycle is a period of time (e.g., a week) where the estimation model identifies the traffic risk in the target area. In particular, we define w to be the w^{th} sensing cycle and W to be the total number of sensing cycles in the studied duration of the application.

Definition 5: **Ground Truth Traffic Accident Records** (GT): We define GT to be the ground truth traffic accident record data, which is often published by the government authorities (e.g., police department). In our problem, the ground truth traffic accident records are only available in the source area, which we refer to as GT^S.

Definition 6: **Social Sensing Data** (SD): We define SD to be the social sensing data (e.g., social media posts) on traffic accidents. In particular, we denote social sensing data collected from the source and target areas as SD^S and SD^T, respectively.

Definition 7: **Traffic Risk Index** (Y): We define the traffic risk index of a location as the traffic accident rate (i.e., number of accidents per sensing cycle) in that location. In particular, we define $Y_w^S = \{y_{1,w}^S, y_{2,w}^S, ..., y_{A,w}^S\}$ and $Y_w^T = \{y_{1,w}^T, y_{2,w}^T, ..., y_{B,w}^T\}$ as the traffic risk indexes for *source* and *target* areas at sensing cycle w, respectively. $y_{a,w}^S$ and $y_{b,w}^T$ represent the traffic risk indexes of cell c_a^S and c_b^T, respectively. Finally, we define the $\widehat{y_{b,w}^T}$ to be the *estimated* traffic risk index for c_b^T in the target area.

The goal of the migratable traffic risk estimation problem is to correctly estimate the real traffic risk index of the sensing cells in the target area by "migrating" the traffic risk estimation model learned from the source area. Using the definitions above, our problem is formally defined as:

$$\arg\min_{\widehat{y_{b,w}^T}}\left(\frac{1}{W} \cdot \sum_{w=1}^{W} \frac{1}{B} \cdot \sum_{b=1}^{B} abs(\widehat{y_{b,w}^T} - y_{b,w}^T) \mid SD^T, SD^S, Y^S\right) \tag{1}$$

where $abs()$ is function to generate the absolute value of a given number. Y^S is the set of traffic risk indexes for all sensing cells in the source area.

IV. Solution

In this section, we present the DeepRisk scheme to address the migratable traffic risk estimation problem formulated in the previous section. We first present an overview of the DeepRisk scheme and then discuss its components in detail. Finally, we summarize the DeepRisk scheme with pseudocode.

A. Overview of the DeepRisk scheme

The overview of the DeepRisk scheme is shown in Figure 1. It consists of three major components: i) Traffic Risk Feature Extraction (TRFE); ii) Deep Transfer Network Construction (DTNC); iii) Adversarial Deep Network Learning (ADNL). First, the TRFE component extracts the traffic risk features from the unstructured social sensing data in both source and target areas. Second, the DTNC component constructs a principled deep transfer learning network to project the extracted traffic risk features from the target area to the source area. Finally, the ADNL component learns the optimal instance of the deep transfer learning network to make accurate traffic risk estimation in the target area using a novel adversarial learning algorithm.

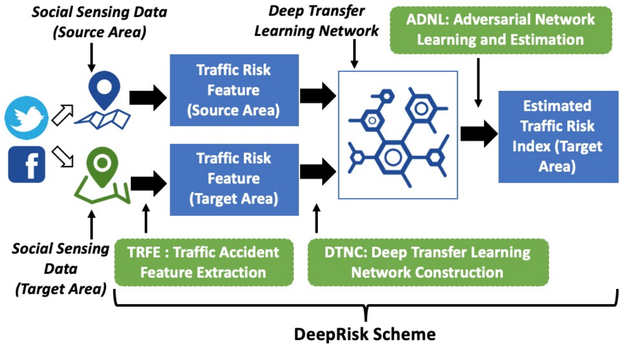

Social Sensing Data (Source Area)
Deep Transfer Learning Network
ADNL: Adversarial Network Learning and Estimation
Traffic Risk Feature (Source Area)
Traffic Risk Feature (Target Area)
Estimated Traffic Risk Index (Target Area)
Social Sensing Data (Target Area)
TRFE : Traffic Accident Feature Extraction
DTNC: Deep Transfer Learning Network Construction
DeepRisk Scheme

Figure 1. Overview of DeepRisk Scheme

B. Traffic Risk Feature Extraction (TRFE)

In this subsection, we describe the TRFE component that extracts the traffic accident related features (i.e., locations and time of accidents) from the unstructured social sensing data collected in both source and target areas for the migratable traffic risk estimation.

The raw social sensing data (e.g., tweets) are usually unstructured (e.g., text) and can not be directly used for traffic risk estimation. For instance, human sensors often provide a high-level description of the accident location instead of the accurate GPS location (e.g., a tweet saying "Incident on #I278 EB at Hunts Point Avenue"). Therefore, the objective of TRFE is to extract the traffic risk features (i.e., locations and time of accidents) and represent the extracted features as feature vectors. In particular, we characterize the traffic risk feature extraction process as follows:

$$\begin{gathered} X_c^{S/T} = \{(\alpha_p, \beta_p) | \alpha_p \in c, \forall p \in SD^{S/T}\} \\ \text{where } (\alpha_p, \beta_p) = \mathcal{F}(p), \forall c \in C^{S/T} \end{gathered} \tag{2}$$

where $X_c^{S/T}$ is the traffic risk feature vector to represent the extracted traffic risk features in sensing cell c from either source area or target area $C^{S/T}$. In addition, we define X^S and X^T to be the sets of traffic risk feature vectors in source area and target area, respectively. p is a piece of social sensing data (e.g., a tweet) collected from source area SD^S or target area SD^T. α_p and β_p are location and time of the accident reported in p. $\mathcal{F}$ is a feature extraction function that extracts the traffic risk features from the raw social sensing data. In particular, traffic accident locations α_p can be extracted from the social sensing data by examining the associated geo-tags (e.g., "coordinates" field of a tweet [4]) or analyzing the content of social sensing data [20]. The accident time β_p can be extracted by checking the timestamp of the data sample (e.g., "created_at" field of a tweet).

C. Deep Transfer Network Construction (DTNC)

In this subsection, we describe the DTNC component (Figure 2) that migrates the risk estimation model from the source to the target area through a principled deep transfer learning network. The DTNC component takes the traffic risk feature vectors generated by the TRFE component as inputs and constructs a set of neural networks for the traffic risk estimation in the target area.

We first define three types of neural networks that will be used in our solution.

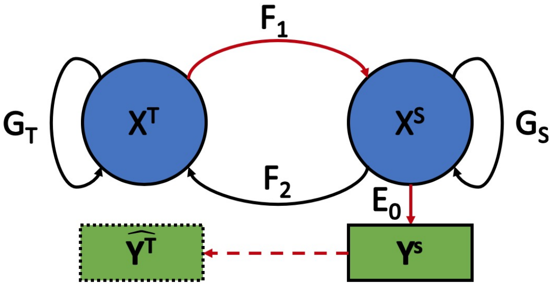


Figure 2. Deep Transfer Learning Network

Definition 8: **Estimation Neural Network E:** We define E as the estimation neural network:

$$E : \mathcal{M} \rightarrow \mathcal{N} \tag{3}$$

where $\mathcal{M}$ is the set of input features (e.g., traffic risk features) and $\mathcal{N}$ is the estimation results (e.g., traffic risk index).

Definition 9: **Transformation Neural Network F:** We define F as the transformation neural network:

$$F : m \in \mathcal{P}(\mathcal{M}) \rightarrow n \in \mathcal{P}(\mathcal{N}) \tag{4}$$

where $\mathcal{P}(\mathcal{M})$ and $\mathcal{P}(\mathcal{N})$ are the distributions of $\mathcal{M}$ and $\mathcal{N}$ defined above, respectively. In particular, the transformation neural network F is able to map a data point m from the distribution $\mathcal{P}(\mathcal{M})$ to a unique data point n from the distribution $\mathcal{P}(\mathcal{N})$.

Definition 10: **Examination Neural Network G:** We define G as the examination neural network:

$$G : \begin{cases} \mathbf{1} : m \in \mathcal{P}(\mathcal{M}) \\ \mathbf{0} : m \notin \mathcal{P}(\mathcal{M}) \end{cases} \tag{5}$$

The network returns "1" (i.e., true) if m belongs to the distribution $\mathcal{P}(\mathcal{M})$ and "0" (i.e., false) otherwise.

In short, the estimation neural network is used to learn an effective traffic risk estimation model in the source area. The transformation and examination neural networks work together to learn the migratable traffic risk estimation model for the target area. We elaborate the above process below.

The first part of our deep transfer learning network is to instantiate an estimation neural network for the traffic risk estimation in the source area (i.e., E_0) as follows:

$$E_0 : X^S \rightarrow Y^S \tag{6}$$

where X^S is the set of traffic risk feature vectors generated by the TRFE component in the source area. Y^S is the set of traffic risk indexes (i.e., traffic accident rate) in the source area, which can be obtained from the ground truth traffic accident records GT^S. The objective of E_0 is to minimize the difference between the estimated and real traffic risk indexes in the source area as follows:

$$\mathcal{L}_E : \arg\min_{E_0} ||E_0(X^S) - Y^S||_2 \tag{7}$$

where $\mathcal{L}_E$ is the objective for estimation network E_0. $E_0(X^S)$ and Y^S are the estimated and real traffic risk indexes of the source area, respectively. $||\cdot||_2$ donates the L2-norm of a given matrix. We develop an efficient adversarial deep network learning scheme to learn the optimal instance (i.e., optimal weight of each node in the network) of E_0, which will be discussed in the next subsection.

The input data to the learned estimation network E_0 has to share the same underlying distribution as the training data (i.e., X^S) to avoid the undesirable overfitting problem [26]. However, the discrepancy between the source and target area naturally lead to different traffic risk distributions between the two areas [12]. Please note that it is not feasible to directly modify E_0 (e.g., re-tuning the weights of the network, etc.) to fit the traffic risk distributions of the target area due to the complex and latent nature of the risk features as well as the insufficient ground truth traffic accident records in the target area [27]. Therefore, we transform the traffic risk features from the target area (i.e., (X^T)) to fit in with the traffic risk feature distribution of the source area (i.e., $\mathcal{P}(X^S)$) using the transformation networks:

$$\begin{aligned} F_1 &: x^T \in \mathcal{P}(X^T) \rightarrow x^S \in \mathcal{P}(X^S) \\ F_2 &: x^S \in \mathcal{P}(X^S) \rightarrow x^T \in \mathcal{P}(X^T) \end{aligned} \tag{8}$$

where $\mathcal{P}(X^T)$ and $\mathcal{P}(X^S)$ are the traffic risk feature distributions in target area and source area, respectively. In particular, we define two transformation networks F_1 and F_2 to ensure the consistency of the transformation process (e.g., $x^T = F_2(F_1(x^T))$, $x^S = F_1(F_2(x^S))$) as follows:

$$\begin{aligned} \mathcal{L}_F : \arg\min_{F1,F2} & \sum_{x^T \in \mathcal{P}(X^T)} (||F_2(F_1(x^T)) - x^T||_1) \\ & + \sum_{x^S \in \mathcal{P}(X^S)} (||F_1(F_2(x^S)) - x^S||_1) \end{aligned} \tag{9}$$

[4] https://developer.twitter.com/en/docs/tutorials/filtering-tweets-by-location.html

where $\mathcal{L}_F$ is the objective for transformation networks F_1 and F_2. $||\cdot||_1$ donates the L1-norm of a given matrix.

The above objective function ensures the one-to-one transformation between the original and transformed feature vectors (e.g., x^T and $F_1(x^T)$). However, the transformed feature vector (e.g., $F_1(x^T)$) has not been regularized to the desired distribution (e.g., $\mathcal{P}(X^S)$). To regularize the transformed feature vectors, we introduce two examination neural networks G_S and G_T for the traffic risk feature distributions in source and target areas (i.e., $\mathcal{P}(X^S)$ and $\mathcal{P}(X^T)$) as:

$$G_S : \begin{cases} \mathbf{1} : x \in \mathcal{P}(X^S) \\ \mathbf{0} : x \notin \mathcal{P}(X^S) \end{cases} \qquad G_T : \begin{cases} \mathbf{1} : x \in \mathcal{P}(X^T) \\ \mathbf{0} : x \notin \mathcal{P}(X^T) \end{cases} \tag{10}$$

Consider a traffic risk feature vector x^T and its transformed feature vector $F_1(x^T)$ generated by the transformation network F_1. The transformation network F_1 is effective if $F_1(x^T)$ belongs to $\mathcal{P}(X^S)$. In particular, the transformed feature vector $F_1(x^T)$ should be verified by the examination network G_S (i.e., returning 1). We update the objective for F_1 and F_2 by leveraging the examination networks as follows:

$$\begin{aligned} \mathcal{L}_F : \underset{F1,F2}{\arg\min} & \sum_{x^T \in \mathcal{P}(X^T)} (||F_2(F_1(x^T)) - x^T||_1) \\ & + \sum_{x^S \in \mathcal{P}(X^S)} (||F_1(F_2(x^S)) - x^S||_1) \\ & + \sum_{x^T \in \mathcal{P}(X^T)} ||\mathbf{1} - G_S(F_1(x^T))||_E \\ & + \sum_{x^S \in \mathcal{P}(X^S)} ||\mathbf{1} - G_T(F_2(x^S))||_E \end{aligned} \tag{11}$$

where $||\cdot||_E$ donates the cross-entropy loss of a given matrix [28]. $\mathcal{L}_F$ is the final objective for transformation network F_1 and F_2. The above process requires examination networks to clearly i) identify imperfect transformed feature vectors (e.g., $F_1(x^T)$) that do not fit in with the desired distributions (e.g., $\mathcal{P}(X^S)$); and ii) differentiate identified feature vectors from the original ones (e.g., x^S). Hence, we define the objective for examination networks G_S and G_T as follows:

$$\begin{aligned} \mathcal{L}_G : \underset{G_S,G_T}{\arg\min} & \sum_{x^T \in \mathcal{P}(X^T)} ||\mathbf{0} - G_S(F_1(x^T))||_E \\ & + \sum_{x^S \in \mathcal{P}(X^S)} ||\mathbf{1} - G_S(x^S)||_E \\ & + \sum_{x^S \in \mathcal{P}(X^S)} ||\mathbf{0} - G_T(F_2(x^S))||_E \\ & + \sum_{x^T \in \mathcal{P}(X^T)} ||\mathbf{1} - G_T(x^T)||_E \end{aligned} \tag{12}$$

where $||\cdot||_E$ donates the cross-entropy loss of a given matrix.

D. Adversarial Deep Network Learning (ADNL))

In this subsection, we describe the Adversarial Deep Network Learning (ADNL) component. The goal of ADNL is to obtain the optimal instances (e.g., optimal weights of nodes in the network) of the neural networks (i.e., E_0, F_1, F_2, G_S, G_T) in DTNC component to make accurate traffic risk estimation for the target area.

We formulate the problem of learning the optimal deep transfer learning network as an adversarial learning problem. In particular, we observe that the transformation neural networks (i.e, F_1, F_2) and examination neural networks (i.e, G_S, G_T) formulated in the DTNC component have opposite objectives (i.e., $\mathcal{L}_F$ in Equation 11 and $\mathcal{L}_G$ in Equation 12). On one hand, the transformation networks target at effective transformations so the examination networks believe all transformed feature vectors belong to the desired distributions. On the other hand, the examination networks target at identification of imperfect feature vectors generated by transformation networks. To capture such conflicting objectives, we divide the neural networks into two adversarial groups as follows:

$$\begin{aligned} & \mathcal{L}_A : \underset{E_0,F_1,F_2}{\arg\min} (\gamma_E \cdot \mathcal{L}_E + \gamma_F \cdot \mathcal{L}_F) \\ & \mathcal{L}_B : \underset{G_S,G_T}{\arg\min} (\gamma_G \cdot \mathcal{L}_G) \end{aligned} \tag{13}$$

where $\mathcal{L}_A$ and $\mathcal{L}_B$ represent the objectives for the two adversarial groups, respectively. In particular, $\mathcal{L}_A$ includes the networks (E_0, F_1, F_2) that aim to make effective feature transformations. $\mathcal{L}_B$ includes the networks (G_S, G_T) that aim to identify any imperfect transformed feature vectors (e.g., $F_1(x^T)$). $\gamma_E, \gamma_F, \gamma_G$ are the weights for different neural networks, which are usually set to be a small positive number with the order of $\gamma_F > \gamma_G > \gamma_E$ to ensure the consistency and effectiveness of the feature transformation process [12]. Given the above adversarial objective functions, we can alternatively minimize $\mathcal{L}_A$ and $\mathcal{L}_B$ using gradient descent techniques (e.g., Adam optimization [29]) to obtain the optimal instances of all neural networks, which ensure the effective feature transformation through F_1 and accurate traffic risk estimation through E_0. Finally, we apply the optimal neural networks to estimate the traffic risk indexes in the target area as follows:

$$x^T \rightarrow F_1^*(x^T) \rightarrow E_0^*(F_1^*(x^T)) \rightarrow \widehat{y^T}, \forall x^T \in X^T \tag{14}$$

where E_0^* and F_1^* are the optimal instances of E_0 and F_1, respectively. In the above estimation, we first transform the feature vector x^T from the target area to fit in with the distribution of the source area as the transformed feature vector $F_1^*(x^T)$. We then apply the estimation neural network E_0^* to obtain the traffic risk estimation result $E_0^*(F_1^*(x^T))$ as the estimated traffic risk index $\widehat{y^T}$ for the feature vector x^T.

We summarize the DeepRisk scheme in Algorithm 1. The inputs to the algorithm are the social sensing data collected from both source and target areas (SD^S, SD^T) and traffic risk indexes in source area Y^S. The output is the estimated traffic accident rate for each sensing cell in target area $\widehat{y^T}$.

[5] threshold is usually set to be a small value (i.e., less than 0.1) to ensure the accuracy of the learned neural networks

Algorithm 1 Summary of the DeepRisk Scheme

```
initialize $F_1$,$F_2$,$G_S$,$G_T$,$E_0$ (e.g., MLP, CNN)
set $\gamma_E$, $\gamma_F$, $\gamma_G$
extract $X^S$ and $X^T$ from $SD^S$ and $SD^T$ using TRFE component and obtain $Y^S$ from $GT^S$
calculate initial $\mathcal{L}_E$, $\mathcal{L}_F$, $\mathcal{L}_G$ using Equation 7, Equation 11, Equation 12, respectively
calculate initial $\mathcal{L}_A$ and $\mathcal{L}_B$ using Equation 13
set $\Delta_{\mathcal{L}_A} \leftarrow +\infty$ and $\Delta_{\mathcal{L}_B} \leftarrow +\infty$
while $\Delta_{\mathcal{L}_A}$ or $\Delta_{\mathcal{L}_B} \geq$ threshold[5] do
    optimize $\mathcal{L}_A$ using Adam optimizer
    calculate updated $\mathcal{L}_E$, $\mathcal{L}_F$ using Equation 7, Equation 11 as $\mathcal{L}'_E$, $\mathcal{L}'_F$, respectively
    calculate updated $\mathcal{L}_A$ using Equation 13 as $\mathcal{L}'_A$
    $\Delta_{\mathcal{L}_A} \leftarrow |\mathcal{L}'_A - \mathcal{L}_A|$
    $\mathcal{L}_E \leftarrow \mathcal{L}'_E$, $\mathcal{L}_F \leftarrow \mathcal{L}'_F$,$\mathcal{L}_A \leftarrow \mathcal{L}'_A$
    optimize $\mathcal{L}_B$ using Adam optimizer
    calculate updated $\mathcal{L}_G$ using Equation 12 as $\mathcal{L}'_G$
    calculate updated $\mathcal{L}_B$ using Equation 13 as $\mathcal{L}'_B$
    $\Delta_{\mathcal{L}_B} \leftarrow |\mathcal{L}'_B - \mathcal{L}_B|$
    $\mathcal{L}_G \leftarrow \mathcal{L}'_G$,$\mathcal{L}_B \leftarrow \mathcal{L}'_B$
end while
for each sensing cell in target area do
    output $\widehat{y^T}$ with optimized neural network $F_1^*$, $E_0^*$ using Equation 14
end for
```

V. Evaluation

In this section, we evaluate the performance of the DeepRisk scheme using the real world traffic datasets collected from New York City. We compare the performance of DeepRisk with state-of-the-art traffic risk estimation baselines in the literature. The evaluation results show that DeepRisk significantly outperforms the baselines in terms of the estimation accuracy.

A. Dataset

We study the migratable traffic risk estimation problem using the real world traffic datasets collected from four different boroughs in New York City (i.e., Manhattan (Mn), Bronx (Bx), Queens (Qs), and Brooklyn (Bn)). These boroughs are observed to have different layouts, road conditions, traffic volumes, and population density, which create a challenge scenario for the DeepRisk scheme. We choose the four boroughs in our evaluation because we can obtain the detailed ground truth labels on traffic accidents in these boroughs as discussed below. We use the ground truth labels in the target area for the evaluation purpose only.

Twitter Traffic Report Dataset: We collected a dataset by using the Twitter API [6] on traffic accidents as our *social sensing data*. This dataset consists of 239,734 traffic-related tweets from the four different boroughs (i.e., Manhattan, Bronx, Queens, and Brooklyn) in New York City over the time period from Jan. 1st, 2016 to Jun. 30th, 2018.

Motor Vehicle Accident Report Dataset: We use a dataset published by New York City Police Department (NYPD) [7] to obtain the ground truth labels of traffic risk index in the studied areas. This dataset consists of 568,051 reports of traffic accidents that happened in the four major boroughs in New York City between Jan. 1st, 2016 and June 30th, 2018. Each report contains the accurate time and location of an accident.

B. Baselines

We choose several representative traffic risk estimation schemes as the baselines in our evaluations. To ensure fairness, the inputs to the baselines and DeepRisk are the same (i.e., social sensing data from both source and target areas and the accident reports from the source area).

- **Random Based Estimation (RAND)**: it estimates the traffic risk index of a sensing cell by randomly selecting a number between 0 and the highest traffic accident rate in our dataset.
- **Linear Regression Based Estimation (LR)**: it trains a linear regression model to minimize the difference between the *true* and *estimated* traffic risk indexes using ordinary least squares (OLS) and makes estimation accordingly [30].
- **Bayesian Based Estimation (ARD)**: it uses the Bayesian automatic relevance determination (ARD) to fit the weights of the estimation model to minimize the difference between the *true* and *estimated* traffic risk indexes and makes the estimation based on the learned weights [31].
- **Lasso Based Estimation (LA)**: it learns the weights of the LA model and adds a Lasso regularizer to enforce the robustness of the learned model and makes the estimation based on the learned weights [32].
- **Multi-layer perceptron Based Estimation (MLP)**: it trains a multi-layer neural network model to learn a non-linear function approximator to minimize the difference between the *true* and *estimated* traffic risk indexes and makes the estimation based on the learned non-linear function approximator [33].

C. Evaluation Metric

In our evaluation, we define the following metric to evaluate the performance of all compared schemes.

- **Mean Absolute Error (MAE)**: the average estimation error for all sensing cells across all sensing cycles. Specifically. We define:

$$MAE = \frac{1}{W} \cdot \sum_{w=1}^{W} \frac{1}{B} \cdot \sum_{b=1}^{B} abs(\widehat{y_{b,w}^T} - y_{b,w}^T) \qquad (15)$$

where B is the number of the sensing cells in the target area and W is number of the sensing cycles.

D. Evaluation Results

1) Estimation Accuracy: In this subsection, we present the results of our DeepRisk scheme and all compared baselines on the real world datasets discussed above. We study the performance of all schemes by varying the combinations of *source* and *target* areas. Specifically, in each set of the experiment, we set the source area to be one of the four boroughs in NYC and set each of the remaining three boroughs as the target area. In our experiment, we set the length of the

[6] https://developer.twitter.com

[7] https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

sensing cycle to be one week by considering the frequency of the accidents in the studied area. In our experiment, we focus on the sensing cells with more than 100 accidents over the studied time period, which translates to more than one accident per sensing cycle. In addition, we set the structures of the neural networks in our evaluation for network F_1, F_2, E_0 to be 10 layers with 64 nodes in each layer for complex multi-variable mapping tasks. We set G_S, G_T to be 2 layers with 3 nodes in each layer for the simple binary classification tasks. We also set the activation function in each network to be ReLU function for effective transformation between layers.

The results are shown in Table I to Table IV. In Table I, we observe that the performance gains achieved by DeepRisk scheme compared to the best performed baseline are 27.6%, 10.7%, 12.4% respectively on three different target areas. Such performance gains of DeepRisk are achieved by judiciously constructing a principled deep transfer network to effectively migrate the risk estimation model from the source area to the target area. In particular, the DeepRisk scheme explicitly transforms the traffic risk features from the target area to fit in with the feature distribution of source area and obtains the optimal risk estimation function through an adversarial deep network learning process. In addition, we continue to observe that our DeepRisk scheme to consistently outperform the compared baselines in Table II to Table IV when we shuffle the source and target area combinations. The consistent performance improvements of DeepRisk demonstrate its robustness and effectiveness across different source and target area settings.

Table I
PERFORMANCE COMPARISONS (MAE) ON DIFFERENT SOURCE AND TARGET AREA SETTINGS (SOURCE = BRONX)

	Source and Target Area Settings		
Algorithm	Bx →Qs	Bx → Mn	Bx → Bn
RAND	8.108	8.162	7.963
LR	7.430	3.166	4.667
ARD	1.467	1.161	1.133
LA	2.039	1.478	1.287
MLP	1.845	1.526	1.253
DeepRisk	**1.062**	**1.036**	**0.992**

Table II
PERFORMANCE COMPARISONS (MAE) ON DIFFERENT SOURCE AND TARGET AREA SETTINGS (SOURCE = MANHATTAN)

	Source and Target Area Settings		
Algorithm	Mn → Bx	Mn → Qs	Mn → Bn
RAND	8.191	8.120	8.494
LR	7.664	10.096	1.196
ARD	1.249	1.857	1.064
LA	2.097	2.901	1.076
MLP	1.438	2.284	1.074
DeepRisk	**1.118**	**1.019**	**0.966**

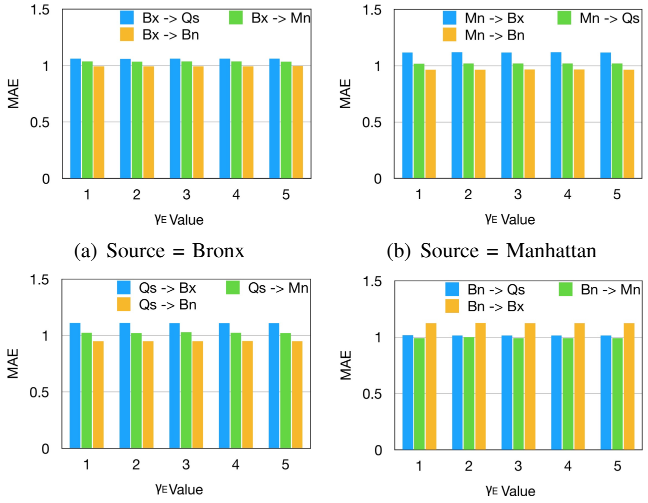
Table III
PERFORMANCE COMPARISONS (MAE) ON DIFFERENT SOURCE AND TARGET AREA SETTINGS (SOURCE = QUEENS)

	Source and Target Area Settings		
Algorithm	Qs → Bx	Qs → Mn	Qs → Bn
RAND	8.633	8.350	7.689
LR	9.495	10.022	2.751
ARD	1.189	1.048	1.040
LA	2.945	1.215	1.263
MLP	1.215	1.113	1.155
DeepRisk	**1.110**	**1.029**	**0.949**

Table IV
PERFORMANCE COMPARISONS (MAE) ON DIFFERENT SOURCE AND TARGET AREA SETTINGS (SOURCE = BROOKLYN)

	Source and Target Area Settings		
Algorithm	Bn → Qs	Bn → Mn	Bn → Bx
RAND	8.687	7.349	8.185
LR	6.434	3.664	4.805
ARD	2.197	1.140	1.842
LA	2.537	1.343	2.065
MLP	2.956	1.100	1.465
DeepRisk	**1.016**	**0.991**	**1.124**

2) Model Robustness: In this subsection, we study the robustness of the DeepRisk scheme by varying the values of the parameters in our model. One key parameter in our DeepRisk scheme is the weighting parameter γ_E (defined in Definition 13). This parameter controls the trade-off between the convergence *rate* and *degree* of the estimation neural network E_0 (defined in Equation 6) [34]. The results are shown in Figure 3. We observe that performance (estimation accuracy) of the DeepRisk is relatively stable on different source and target area settings when the value of γ_E changes.



(a) Source = Bronx

(b) Source = Manhattan

(c) Source = Queens

(d) Source = Brooklyn

Figure 3. Robustness of DeepRisk Scheme

VI. Conclusion

In this paper, we develop a DeepRisk scheme to solve the migratable traffic risk estimation problem in intelligent transportation system using social sensing. In particular, we develop a principled deep transfer learning network to effectively migrate the risk estimation model from the source area to the target area. We also design a novel adversarial learning algorithm to learn the traffic risk related features for effective risk estimation model migration. The evaluation results on the real world case study demonstrate that the DeepRisk achieves significant performance gains compared to the state-of-the-art baselines in accurately estimating the traffic risk of locations in a city.

Acknowledgement

This research is supported in part by the National Science Foundation under Grant No. CNS-1831669, CBET-1637251, CNS-1566465 and IIS-1447795, Army Research Office under Grant W911NF-17-1-0409, Google 2017 Faculty Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] D. Wang, T. Abdelzaher, and L. Kaplan, *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann, 2015.

[2] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, p. 88, 2016.

[3] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on*. IEEE, 2012, pp. 233–244.

[4] Y. Zhang, D. Zhang, N. Vance, and D. Wang, "Optimizing online task allocation for multi-attribute social sensing," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018, pp. 1–9.

[5] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *Computer*, vol. 52, no. 1, pp. 36–45, 2019.

[6] R. Yu and M. Abdel-Aty, "Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data," *Safety science*, vol. 63, pp. 50–56, 2014.

[7] J. Sun and J. Sun, "A dynamic bayesian network model for real-time crash prediction using traffic speed conditions data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 176–186, 2015.

[8] Z. Yuan, X. Zhou, and T. Yang, "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 984–992.

[9] Z. Qin, Z. Fang, Y. Liu, C. Tan, W. Chang, and D. Zhang, "Eximius: A measurement framework for explicit and implicit urban traffic sensing," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2018, pp. 1–14.

[10] D. Wang, T. Abdelzaher, L. Kaplan, and C. C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 2013, pp. 530–539.

[11] Y. Gu, Z. S. Qian, and F. Chen, "From twitter to detector: Real-time traffic incident detection using social media data," *Transportation research part C: emerging technologies*, vol. 67, pp. 321–342, 2016.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks."

[13] C. Schaffer, "Overfitting avoidance as bias," *Machine learning*, vol. 10, no. 2, pp. 153–178, 1993.

[14] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti *et al.*, "Using humans as sensors: an estimation-theoretic perspective," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 35–46.

[15] D. Zhang, Y. Ma, C. Zheng, Y. Zhang, X. S. Hu, and D. Wang, "Cooperative-competitive task allocation in edge computing for delay-sensitive social sensing," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 243–259.

[16] N. Vance, D. Y. Zhang, Y. Zhang, and D. Wang, "Privacy-aware edge computing in social sensing applications using ring signatures," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 755–762.

[17] D. Y. Zhang, L. Shang, B. Geng, S. Lai, K. Li, H. Zhu, M. T. Amin, and D. Wang, "Fauxbuster: A content-free fauxtography detector using social media comments," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 891–900.

[18] D. Zhang, D. Wang, N. Vance, Y. Zhang, and S. Mike, "On scalable and robust truth discovery in big data social media sensing applications," *IEEE Transactions on Big Data*, 2018.

[19] D. Zhang, Y. Ma, Y. Zhang, S. Lin, X. S. Hu, and D. Wang, "A real-time and non-cooperative task allocation framework for social sensing applications in edge computing systems," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2018, pp. 316–326.

[20] Y. Zhang, Y. Lu, D. Zhang, L. Shang, and D. Wang, "Risksens: A multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1544–1553.

[21] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference." in *AAAI*, 2016.

[22] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 513–520.

[23] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon, "Transfer learning from deep features for remote sensing and poverty mapping," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] C. Xiaoxuan Lu, B. Du, P. Zhao, H. Wen, Y. Shen, A. Markham, and N. Trigoni, "Deepauth: in-situ authentication for smartwatches via deeply learned behavioural biometrics," 10 2018, pp. 204–207.

[25] S. Bhattacharya and N. D. Lane, "Sparsification and separation of deep learning layers for constrained resource inference on wearables," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 176–189.

[26] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2868–2876.

[27] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[28] G. E. Nasr, E. Badr, and C. Joun, "Cross entropy error function in neural networks: Forecasting gasoline demand." in *FLAIRS Conference*, 2002, pp. 381–384.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012, vol. 329.

[31] J. Drugowitsch, "Variational bayesian inference for linear and logistic regression," *arXiv preprint arXiv:1310.5438*, 2013.

[32] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[33] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[34] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.